

opentextTM

OpenTextTM InfoArchive

Version 16 EP4

Configuration and Administration User Guide

Legal Notice

This documentation has been created for software version 16 EP4.

It is also valid for subsequent software versions as long as no new document version is shipped with the product or is published at <https://knowledge.opentext.com>.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111

Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440

Fax: +1-519-888-0677

Support: <https://support.opentext.com>

For more information, visit <https://www.opentext.com>

Copyright © 2018 Open Text. All Rights Reserved.

Trademarks owned by Open Text.

Adobe and Adobe PDF Library are trademarks or registered trademarks of Adobe Systems Inc. in the U.S. and other countries.

Disclaimer

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the accuracy of this publication.

Table of Contents

Revision History	17
Chapter 1 Product Overview	19
What is InfoArchive?	19
Decommissioning Legacy Applications	19
Live Archiving	19
What Can Be Archived?	20
Key Features	20
High-Level Overview: Using InfoArchive	21
Archiving Overview	24
Backing Up vs. Archiving Data	24
Standards	24
Table Archiving and SIP Archiving	25
File Archiving	25
Data Record Archiving	26
Compound-Record Archiving	26
The Archiving Process	27
What Type of Archiving Should I Use?	28
Setting Archiving Goals	28
Understanding the Source Application	29
Selecting the Appropriate Archiving Method	29
Use Cases	30
Applications and Data	32
Architecture	32
How Data is Ingested	32
How Data is Stored	32
How Data is Searched	33
How InfoArchive is Configured for Data	34
Data Organization for Table Archiving	35
Extracted and Converted Data	35
Metadata File	35
Table Data Files	36
Build Properties File	36
Build File	37
The Configuration YAML File	37
Authentication for Declarative Configuration Applications	37
Searches	37
The Indexing Process for Ant Script Applications	38
The Indexing Process for Declarative Configuration Applications	39
Configuring InfoArchive Server Table Indexing	39
System-Wide Configuration Settings	39
Table Indexing Batch-Specific Configuration Settings	40
Table Indexing Job Configuration Settings	40
Tweaking and Fine-Tuning	41
Full-Text Indexing vs. Value Indexing	43
Installation and Ingestion	44

Data Organization for SIP Archiving.....	45
Extracted and Converted Data.....	45
SIPs	45
Best Practices for SIPs	46
AIPs	47
AIUs.....	48
Holdings.....	49
Data Submission Sessions	50
SIP Descriptor	50
SIPs and Data Submission Sessions.....	51
PDI File and Schema File.....	52
Best Practices for the Schema File.....	53
Additional SIP Archiving Files for Ant Script Applications.....	54
Base Files	54
Ingestion File.....	55
pdi-crypto.xml File	55
pdi.xml File.....	55
Additional SIP Archiving Files for Declarative Configuration Applications.....	56
The Configuration YAML File.....	57
Authentication	57
PDI Crypto File	57
PDI Configuration File	57
Installation and Ingestion.....	58
Cache-In and Cache-Out	59
SIP Ingestion Modes	61
Unitary Archiving and Aggregation	62
Operational Concepts	64
Authentication Mechanisms	64
Authorization	65
User Roles	65
Actions	66
Auditing	66
Chapter 2 Configuration	69
Creating an Application Using the Web Application	69
Editing an Application.....	72
Deleting an Application	72
Deleting Data from an Application	73
Creating Federations, Databases and Storage Systems Using the Web Application	73
Registering a Federation	74
Linking a Database	75
Configuring xDB Federations and xDB Databases.....	75
Adding a Storage System Using the Web Application	76
Configuring Storage Systems	79
Configuring ECS Storage	80
ECS Data Model for InfoArchive	80
Configuring Centera Storage	81
Configuring AWS S3 with Amazon Glacier	81
Using Multi-Part Capabilities to Improve SIP Upload Performance.....	82
Configuring Custom Storage	82
Creating a Space Using the Web Application	84
Adding a Store Using the Web Application	86
Editing and Deleting Administration Configuration Objects	88

Configuring a Holding	99
Configuring a Holding Using the Holding Configuration Wizard	100
Limitations of the Holding Wizard	111
Holding Store Configuration	111
Staging Store and Centera	111
Glacier	112
Pooled Libraries	112
Configuring the Ingestion Process to Store Multiple AIPs in the Same xDB Library.....	112
Executing the Close Job to Close the xDB Libraries and Perform a Back Up	113
Partition Keys	114
Types of Partition Keys	114
Indexes	115
Types of Indexes.....	115
Configuring Indexes for a SIP Archive	115
Configuring Indexing	115
Using a Path Value Index	116
Configuring Indexes for a Table Archive	117
Creating Path Value Indexing	117
Confirmation.....	119
The Confirmation Job and the Aggregates	120
Cache-In/Cache-Out Feature	120
Cache Size of the Application	120
Cache-Out Job	120
Troubleshooting	121
Manual Cache-In/Cache-Out Requests	121
Reset Library Access Statistics	122
Frequently Asked Questions – Caching	123
Configuring Back Up and Restore	124
Backing Up and Restoring the Managed Item Database.....	124
Configuring the Managed Items Database	124
Backing Up the Managed Items Database	125
Setting the Managed Item Store	125
XdbLibrary	126
Restoring the Managed Database	126
Using the XdbLibrary	126
Selected Managed Items.....	128
Using the Managed Items.....	129
Backing Up and Restoring Table Databases	130
Configuring an Audit	131
Configuring Audit Events Using the Audit Tab	132
Configuring System-Level Audits for Provisioning Events	135
Configuring System-Level Audits for Other Events.....	136
Configuring Tenant-and Application-Level Audits for Provisioning Events	136
Configuring Tenant-and Application-Level Audits for Compliance Events	139
Configuring Tenant-and Application-Level Audits for Ingestion Events	140
What are Ingestion Events.....	140
Configuring Tenant-and Application-Level Audits for Other Events	142
What are the Other Events	142
Supplemental Data	144
Testing that Audit Changes Work	144
Audit Troubleshooting.....	144

Language Support	145
Adding New Language Support.....	145
Customizing Branding.....	146
Configuration to Add Sample Branding Customization.....	146
Configuration to Add New Branding Customization	146
Setting the Customization Location when Deploying to Tomcat or Other Containers	147
Verifying and Viewing the Branding Customization	148
Troubleshooting	148
InfoArchive JDBC Driver	148
SQL Support	148
Supported SQL Functions	148
Supported SQL Features	149
Unsupported SQL Features	151
SQL Type and XQuery Translation	151
Connection Setup	152
Configuring OpenText Directory Services.....	154
Downloading and Installing OTDS	156
Using OTDS in Authentication Provider Mode.....	156
Configuring OTDS for Authentication Provider Mode	157
Using OTDS in SSO Mode.....	158
Configuring OTSD for SSO Mode	159
Troubleshooting OTDS Issues.....	164
Setting the Login Format.....	165
Validating the Integration of SAML 2.0 with OTDS and InfoArchive	166
Configuring Okta as a SAML 2.0 IDP Inside OTDS	168
Creating a Non-synchronized Partition <code>infoarchive-okta</code>	168
Creating the SAMPL 2.0 Authentication Handler.....	169
Working with the Deployment Configuration Files	170
Changing the Default Ports for InfoArchive Components.....	170
Working with Gateway and InfoArchive Web Application	
Configuration Files	171
InfoArchive Server and InfoArchive Web Application	
Communication Setup	180
Self-Signed Certificates-Based Setup	182
Creating Certificates and Trust Establishment	187
One-Way TLS	187
Two-Way TLS.....	187
Working with the Server's <code>application.yml</code> File	188
Security Profile	188
<code>logging.level</code>	189
<code>infoarchive</code> Section	189
<code>managedItemData</code> Section.....	189
<code>defaultNames</code> Section	190
Storage Paths and OAIS Sections	190
<code>job</code> Section.....	190
<code>roles</code> Section	191
Updating the Working Directory	191
Default File System Root.....	191
System and Audit Database	191
Configuring the Number of Items Listed in the Search Results.....	193
Configuring the Time Limit for a Background Search.....	193
Configuring the xDB Segment Size for a Table Archive	194
Limiting the Size of Files Transferred Through REST	194
Working with the <code>xdb.properties</code> File	195
Running Chain-of-Custody Tests for Table Archives	195
Using InfoArchive's Batch Processing Functionality	196

Configuring the Batch Size	197
Viewing Batch and Log Information	198
Chapter 3 Declarative Configuration	201
Configuring a SIP Archive	201
Configuring PDI.....	202
Configuring Indexes	206
Types of Indexes.....	206
Configuring Indexing	207
Using a Path Value Index	207
Creating Path Value Indexing for a Table Archive	208
PDI.INDEX.CREATOR.....	210
PDI.AIU.CNT.....	211
PDI.AIU.ID	212
Using Partition Keys	212
Types of Partition Keys	213
Configuring Partition Keys.....	213
Three Common Partition Key Queries.....	214
RI.INIT	215
XDB.PDI.CI.ID	215
Configuring a Holding	216
Holding Configuration – Stores	217
Holding Configuration: Hashing	218
Holding Configuration: Retention	218
Configuring Ingestion.....	219
Configuring xDB Ingestion Mode	219
Unitary Archiving and Aggregation	219
Receiving and Ingesting a SIP in One Request	219
Configuring the Ingestion Process to Store Multiple AIPs in the Same xDB Library	220
Executing the Close Job to Close the xDB Library and Perform a Back Up.....	221
Configuring the Ingestion Process to Allow Aggregation	221
Reception.....	222
Using a Staging Store (Optional)	223
Executing the Confirmation Job to Identify Open and Closed Aggregates.....	223
Retrieving a CI from an Aggregate	223
Configuring a Search	224
Configuring the AIC	224
Using an AIC Predicate	225
Configuring the Query	226
Query Template	227
Configuring the Query Quota.....	227
Configuring the ResultHelper.....	228
Configuring Repeating Elements	232
Using the Confirmation Mechanism	233
Working with the Confirmation Object.....	234
Working with the DeliveryChannel Object	236
Creating an Application Using Declarative Configuration	239
Detailed Information About the Declarative Configuration Format	239
Using the Correct Syntax.....	240
Including Other Configuration Files.....	242
Using Default Values	243
Using Properties Files	245
Working with Namespaces and Queries	246

Preventing Overwrites	246
Configuring Import/Export Functionality	247
Chapter 4 Searches in InfoArchive	249
What is Search?	249
How Searches Work in a SIP Archive	249
Who Uses the Search Functionality?.....	250
Composition and Configuration	250
Use of Search.....	250
Searches and Amazon Glacier	250
Preparation for Searches	252
How Search Differs for Table and SIP Archives.....	252
Custom Exports	252
Export Pipeline.....	253
Transformation.....	253
Export Configuration	253
Custom Presentation.....	254
Writing XProc Pipelines to Export Search Results	254
Referencing External Resources	257
Restrictions to the InfoArchive Search Result Export XProc Pipelines	257
Extending InfoArchive XProc Export Functionality	257
Search Composition for the Developer	260
Overview	260
Search.....	284
Search Listing.....	284
Exporting a Search.....	284
Importing a Search	285
Creating a Duplicate Search	285
Editing a Search.....	285
Creating a Search.....	286
Search Sets	286
Viewing Search Sets	286
Creating a Search Set	286
Composing the Search Form Using the XForms Editor	289
Composing the Search Form Using the User Interface	290
Adding Fields	290
Moving and Resizing Fields	291
Adding Containers	292
Customizing a Container	295
Configuring Multiple Default Values for Checkboxes	295
Adding a Default Value to a Text Field	296
Adding a Default Value to a Number Field.....	297
Adding Default Values to a Number Range Field.....	297
Adding Default Date Range Values to a Search Form	297
Adding Default Number Values to a Search Form.....	298
Customizing Number Ranges.....	298
Searching for a Value in Multiple Fields	298
Configuring a Table-Based Search to Allow Wildcard Searches.....	299
Customizing a Radio Group	301
Customizing the Select Element	302
Example of Data Binding for a SIP Archive	305
Creating a Value List	305
Using Value Lists	306
Deployment	307

Configuration.....	308
XQueries for Value Lists.....	309
Samples	309
Troubleshooting Value Lists	309
Configuring Data Resolution for Supporting Search Value Expansion	310
Configuring Data Resolution to Use XQuery	310
Configuring Data Resolution for Form Control	310
Executing Search During Runtime	311
Configuring Data Resolution to Use an External Service.....	311
Configuring Data Resolution for Form Control	312
Executing the Search at Run Time.....	312
Main Search XQuery Adjustment for a Table Application.....	312
Unsupported Form Controls	313
Samples	313
XQuery Modules	313
Composing the Result List	315
Adding Columns	315
Customizing the Column Type	315
Configuring a Column of Search Results	316
Adding Filters to a Column of Search Results	317
Adding an External Link to Search Results	318
Creating a Nested Search	320
Configuring a Result Column to Allow Downloadable Content.....	321
View	322
OpenText Brava! Viewer	324
Native Browser	326
Configuring Exports	326
Adding the Export Action to the Result List	327
Including Repeating Elements in Exported Data	329
Enabling Collection Functionality	329
Custom View	329
Composing Result Details	330
Adding Details to a Side Panel.....	330
Adding an In-line Panel	331
Adding Tabs.....	331
Customizing Panel Fields for a Table Archive	332
Customizing Panel Fields for a SIP Archive	334
Reordering Fields	336
Managing Permissions	336
Saving a Search Set	336
Renaming a Search Set	336
Deleting a Search	337
Deleting a Search Set.....	337
Duplicating a Search Set.....	337
Search Form Composition Tips	337
Updating a Search Form Status to Ready.....	339
Nested Searches	340
Configuring a Dependent Select Control Powered by XQuery	340
Adding Date-Time Fields During Search Composition	343
Searches and the Cache-In/Cache-Out Feature	344
Synchronous Search.....	344
Background Search.....	346
Limiting Access to an Application or Search.....	346
Searching Content in AWS Glacier	347
Search Troubleshooting.....	349

Chapter 5	Administration	351
	Generating SIPs.....	351
	Connectors.....	351
	Application and Platform Examples.....	352
	Managing Packages	352
	Using the Packages Tab	352
	How to Read the Type Column in the Packages Tab	355
	Retention and ECS and Centera Storages.....	355
	Applying Actions to a Package	355
	Rejecting or Invalidating an AIP	356
	Applying a Retention Policy to an AIP	358
	Applying a Hold to an AIP.....	358
	Dashboard	358
	Storage Metrics.....	359
	Calculation for Pricing	359
	Storage Footprint Calculation.....	359
	Licensed Volume	360
	Questions & Answers – Pricing	360
	Logging	361
	Using a Preexisting Open Source Logging Solution.....	361
	Generating Job Instance and Order Item Logs.....	363
	Downloading Job Instance and Order Item Logs.....	364
	Configuring Jobs	364
	Job Scoping	364
	Application Scoping	364
	System Scoping	365
	Working with Jobs: List of Available Jobs	365
	Archive Audits Job	369
	Clean Job	370
	Clean Up Purge Candidate List and Applications Job	372
	Close Job	372
	Generate Purge Candidate List	373
	Post Ingest Processing Job	374
	Refresh Metrics Job	377
	Remove Policy Job	377
	Requalification Job.....	378
	Trigger Event Policy Job	378
	Populating Event Dates for the Trigger Event Policy Job	379
	Using the Jobs Tab	380
	Viewing a Job's Run History	381
	Creating a Job	381
	Editing a Job.....	384
	Running a Job	386
	Suspending a Job Schedule	386
	Deleting a Job	386
	Reviewing the Logging Information for Jobs	387
	Managing the Log History	387
	Troubleshooting Issues with Jobs	388
	Managing User Accounts and Permissions	389
	Managing Groups.....	389
	Managing Permissions	389
	Restricting Access to an Application or Search.....	390
	Performing a Byte Count on Application Data	391
	Usage Examples	391
	Conducting Periodic Heartbeat Checks on InfoArchive	392

Chapter 6	Compliance – General Concepts	395
	Compliance-Related Roles.....	396
	The Retention Lifecycle	396
	What is a Retention Policy?	396
	What is a Retention Set?	399
	What is a Hold?	399
	What is a Hold Set?.....	399
	Granularity	400
	What is Disposition?	400
	What is a Purge Candidate List?	401
	What is Granular Disposition?.....	401
	Timing of Applying Holds when Using Granular Retention	402
	The Retention Lifecycle	403
Chapter 7	Compliance Related Tasks	405
	Using the Retention Policies Tab	405
	Creating a Retention Policy	406
	Applying Retention	407
	Choosing Where to Apply Retention and the Consequences	407
	Applying Multiple Retention Policies.....	408
	Verifying that an Item is Protected	409
	Verifying Using the Application Info Tab	409
	Verifying Using the Application's Tables Tab	409
	Verifying Using the Application's Packages Tab.....	410
	Verifying Retention in Search Results.....	410
	Table Archiving – Application	411
	SIP Archiving.....	413
	Record-Based Retention	415
	Mechanisms for Applying Retention.....	415
	Using the Applications Tab	415
	Applying a Retention Policy to an Application	415
	Applying a Hold to Search Results	416
	Applying Retention to Records.....	418
	Applying a Retention Policy to an AIP	419
	Using Jobs to Apply Retention.....	420
	Using the Apply Retention Policy to Records Job	420
	Using the Apply Retention Rule Job	422
	Using Rules to Apply Retention.....	423
	Running Rules to Apply Retention for a Table Application	425
	Using the Rule Engine to Apply a Retention Policy or Hold to Records	426
	Defining How a Retention Policy is Applied to Records in an AIP or Table	427
	Rules for Retention Application.....	427
	Ingestion.....	427
	Job.....	427
	Rule Files	428
	Loading Rules	428
	Access to Metadata	429
	Record Bean.....	431
	AipRecordBean	431
	AiuiRecordBean	431
	TableRecordBean	432
	TableRowRecordBean	432
	Execution Rules.....	432

Rules Object	433
REST API.....	434
IAShell.....	435
Applying Retention to Records for SIP	436
Applying Granular Retention Using the Holding Wizard.....	436
Checking that Data is Protected	437
Using the Retention Sets Tab	437
Viewing Items in a Retained Set.....	438
Example of Granular Retention	438
Event-Based Retention	445
What are Compliance Events?	445
Basic Flow.....	446
Define Retention Policy	446
Apply Retention to Records.....	446
Fulfill Events	446
How Does Aging Work?	447
Applying an Event-Based Retention Policy.....	447
Fulfilling Events with the Trigger Event Policy Job	448
Populating Event Dates for the Trigger Event Policy Job	448
Choosing Between Event-Based and Mixed Retention Policies	449
Fulfilling Events with Rules.....	449
Process Retention Events Job	450
SIP Retention.....	450
When is Retention Effective?	450
Confirming a Purge	451
How Does Refactoring Work	451
Hardware Retention	451
When are Dates Pushed to the Hardware?.....	451
Hardware Retention Support Considerations for ECS, Centera and Isilon.....	453
Determining if a Retention Policy is in Use	453
Changing a Retention Policy.....	453
Editing a Retention Policy	454
Impact on Existing Items that have Retention Applied	454
Running the Requalification Job	455
Limitation of Changing Retention Policies based on the Content Store	455
Disposition Flow	455
How are Items Added to a Purge Candidate List?.....	455
Using Rules to Automatically Approve Purge Candidate Lists	457
How Does Disposition Work	458
Application Disposition	458
AIP Disposition.....	459
AIU Disposition	459
Disposition of AIPs and AIUs.....	460
Table Disposition	460
Table Record Disposition.....	460
Disposition of Tables and Table Rows	460
Disposition in SIP-Based Applications	461
When is the Confirmation Job Required?.....	461
Disposition of a Tables	461
Disposing a Table with Records Under Hold	462
Running the Dispose Purge Candidate List Job.....	462
Using the Purge Lists Tab	463
Performing Actions to a Purge List	464

Running the Clean Up Purge Candidate Lists and Applications Job	465
How Long Does Exported Data Stay in the Archive?.....	465
Retention Applied Directly to Table	465
Retention Applied Directly to Package	466
Exporting Purge Lists	467
Running Reports to Check for Overdue and Upcoming Dispositions	467
Holds	468
Using the Holds Tab	469
Creating a Hold.....	469
Editing a Hold.....	470
Deleting a Hold	470
Using the Hold Sets Tab	470
Creating Collections	471
Using the Collections Tab.....	472
Creating a Collection and Applying a Legal Matter.....	473
Using the Legal Matters Tab	474
Applying a Hold	475
Applying a Hold to an Application.....	475
Creating a Legal Matter.....	476
Applying a Hold to an AIP	476
Using the REST Client to Apply a Hold.....	476
Viewing Records Under Hold.....	477
Using Rules to Apply Holds	477
Removing Retention	479
Removing Retention from an Application	479
Deleting a Retention Policy	479
Running the Remove Policy Job.....	480
Removing Holds	480
Viewing Items in a Hold Set	480
Removing an Item from a Hold Set.....	481
Removing a Hold from an Application.....	481
Removing a Collection from a Legal Matter.....	481
Deleting a Collection.....	482
Metrics	482
Performing a Byte Count on Application Data	482
Usage Examples	483
Understanding the Compliance Dashboard	484
Running the Refresh Metrics Job.....	484
Audits	485
Using Audits for Compliance	485
Metadata Fields.....	485
Audit Event Type	486
Application-Specific Audits.....	487
Metadata Fields	487
Searching for Audits	487
Audit Entries.....	487
Using the Archive Audit Job	489
Compliance Troubleshooting.....	491
Chapter 8 InfoArchive for the End User	493
Overview	493
The Applications Landing Page	494
Finding an Application	495
Switching the Language of the User Interface	495

Searching	495
Overview	495
Finding a Specific Search Form	496
Entering Information in a Search Form.....	496
Entering Dates in a Search Field	497
Entering Times in a Search Field	497
Using Multiple Values in a Search Field	498
Using Operators in a Search Field	498
Executing a Search.....	499
Filling a Search Form	499
Running a Search in the Background	499
Viewing Search Results	499
Using a Filter to View Search Results	500
In-Line Panels	502
Viewing Side Panel Search Results.....	502
Downloading Search Results	503
Previewing Search Results	504
Exporting Search Results.....	507
Background Requests Tab	508
Viewing Search Results from a Background Search	509
Deleting a Background Task	510
Chapter 9	
Appendix A – XQuery Best Practices	511
XQuery Structure	511
Prolog.....	512
FLWOR Expression.....	513
XQuery Modules	514
XQuery Performance Considerations	514
XQuery Optimization	515
Index Types.....	515
Multipath index	515
Path Value Index	516
Composite key index	516
Full-Text Search Versus Value Comparison	517
Full-Text Analyzer.....	517
Search for Typed Data.....	518
Table Joins	519
Range Queries	520
XQuery Context and Index Location	520
The XQuery Context Contains the Index	520
The Children of the XQuery Context Contain the Indexes	521
The Parent of the XQuery Context Contains the Indexes.....	521
XQuery Optimizer	522
Optimizer Strategy	522
The Selection Done by the Optimizer is not Always the Best.....	522
Influencing Index Selection	523
Hidden Optimization for Table Archiving	523
Index Statistics	524
The Optimizer Only Optimizes a Subset of XQuery Expressions	524
Debugging an XQuery	525
Making a Search Produce XQuery Debug Output	525
How to Use the xDB Admin Tool for Debugging	527
Finding the XQuery that is Run for a Search	529
Selecting the Context of an XQuery.....	529
How Can I See the Indexes of My XQuery Context?	531
Testing Form Input in the xDB Admin Tool	532

Interpreting XQuery Debug Output.....	532
How can I see if my XQuery is using an index?.....	532
How can I see if an XQuery is using an index in the most optimal way?.....	533
How Can I Learn More About the Decisions of the XQuery Optimizer?.....	534
Building Up an XQuery Step-By-Step.....	534
What to do with a Slow XQuery	536
Limitations of xDB Admin Tool	536
Troubleshooting	536
Further Reading	538
Chapter 10 Appendix B – Using Metrics to Improve Performance	539
Improving Performance for a SIP Archive	540
Improving Ingestion Speed	542
Improving Search Speed for a SIP Archive	542
Improving Performance for a Table Archive	543
Chapter 11 Appendix C – Mapping XSD Data Types	545
Chapter 12 Appendix D – Custom SIP Format Support	549
Goals	549
Reception.....	549
Create a new implementation based on the interface	
SipReceptionHandler.....	549
Configure the reception to handle the new SIP format.....	549
How to receive/ingest a SIP with a specific format with the CLI.....	550
Ingestion.....	550
Create a new implementation based on the interface	
SipIngestionHandler.....	550
Configure the ingestion to handle the new SIP format.....	551
Content Retrieval.....	551
Create a new implementation based on the interface	
ContentDownloadHandler.....	551
Configure the server to perform the transformation	552
Chapter 13 Appendix E – Glossary and Acronyms	553

Table of Contents

Revision History

The following changes have been made to this document:

Revision Date	Description
April 2018	Initial 16 EP4 release.

Revision History

Chapter 1

Product Overview

What is InfoArchive?

InfoArchive is a powerful, secure, and scalable archiving solution for the enterprise. It preserves, maintains, and controls continuing access to valuable enterprise information assets. It is also application agnostic, providing one unified archive for all of your application data.

As organizations generate more data, and retain data longer and longer, IT costs continue to rise. With InfoArchive, you can decommission legacy applications to reduce costs, including maintenance, special hardware, and consultants. You can also archive data from live systems so that the load on the systems is lessened, reducing hardware requirements and improving performance.

In addition to saving costs and improving performance, InfoArchive can help you make better use of your structured and unstructured data by opening up availability and putting the data to work in new ways for the business. InfoArchive is standards compliant and presents data easily in any format. And it features robust compliance functionality so that you can meet all retention requirements.

Decommissioning Legacy Applications

Decommissioning legacy applications is one of the main use cases of InfoArchive. Often, legacy applications are no longer officially supported, run on older hardware that is nearing the end of its lifecycle, and contain data that is no longer being updated. But this data can be valuable to the business and important to preserve for compliance reasons.

With InfoArchive, you can archive the data in a standard format on lower-cost storage, create searches to make the data accessible to your users, and then safely switch off the legacy application. With decommissioning, data is typically extracted all at once from the source application and then archived in InfoArchive.

Live Archiving

Live archiving is the other main use case of InfoArchive. You can decrease the load on a live application, increasing performance and reducing hardware costs, by offloading some of the application's data to InfoArchive.

The data that you choose to archive, and the frequency of archiving, is based on your business rules. For example, on a weekly basis, a bank can archive all bank statements that are a year old or more.

Once the data has been archived, InfoArchive can send a confirmation to the source application to purge the data. With live archiving, data is continually added to the source application and then archived in InfoArchive at defined points.

What Can Be Archived?

InfoArchive can store both structured and unstructured data. Structured data is information that is separated into independent, predictable parts, defined by metadata. InfoArchive stores this content as XML. Examples of structured data include transactions from ERP systems, legal records, and economic data.

Unstructured data is information that does not follow a specified format or predefined model. InfoArchive stores this content on a file system, and InfoArchive stores the content's metadata as XML. Examples of unstructured content include Microsoft Office documents, print streams, image files, and videos.

Depending on the data source, the data might be a combination of structured and unstructured data. InfoArchive supports the following types of data sources:

Data Source	Type of Data	Example
Tables	Mostly structured, might also have some unstructured data	Tables from an RDBMS, such as an Oracle database
Files	Often unstructured, might also have some structured data	Media archives
Data records	Often structured, might also have some unstructured data	Transaction histories
Compound records	Unstructured and structured data in a single record	Laboratory reports

Key Features

The following list summarizes some of the key features of InfoArchive.

- Archive all information types, and structured and unstructured data, in a single archive
- Flexible, controlled access to archived data
 - Allow authorized users to search, view, and export content
 - Role-based permissions and data encryption
 - Customize searches and the interface's look and feel
 - Leverage APIs to integrate query capabilities into your own business applications
- Compliance with open industry standards for platform-independent retention
 - Reference Model for an Open Archival Information System (OAIS)
 - Extensible Markup Language (XML)
 - XQuery
- No dependencies on the originating application

- Synchronous (transactional) and asynchronous (batch) ingestion
 - Archive large and steady input streams of data
 - Schedule ingestion in batches for optimal performance when information comes in intermittently
- Powerful tools for archiving operations and management, including compliance tasks, audits, and metrics
 - Create and apply retention policies and legal holds
 - Apply policies at different levels, including application, package, and record
 - Apply retention during or after ingestion
 - Granular disposition of records
- Extensible and customizable
 - Amazon S3 support for storing unstructured data
 - Integrate any storage type using the storage API interface
 - Configure the presentation of search results using standard HTML templates
- Highly scalable, can meet increasing demands and complexity
 - Supports large volumes of data
 - Supports large numbers of source applications
 - Supports increasing complexity of regulations
- Minimizes IT costs
 - Move data to lower-cost storage
 - Save maintenance and licensing costs for legacy applications
 - Reduce the load on live applications

High-Level Overview: Using InfoArchive

InfoArchive is a powerful and versatile tool. Having a sense of how its functionality fits together, and when to use which features, can speed up your success.

The following table presents the major goals and tasks when using InfoArchive, from initially learning about the product through maintaining and monitoring a fully configured and live production deployment. Most of the tasks are covered in this guide, except for installing and upgrading, which are covered in the *InfoArchive Installation Guide*.

Goal	Task	Where to Find More Information
Learning about InfoArchive and archiving	<ul style="list-style-type: none"> Understand what InfoArchive does Understand how InfoArchive archives data Understand how to satisfy your archiving goals Determine which types of archiving to use Understand the areas of functionality and technical components 	<ul style="list-style-type: none"> What is InfoArchive? Archiving Overview What Type of Archiving Should I Use? <i>InfoArchive Installation Guide</i>
Installing a demo configuration	<ul style="list-style-type: none"> Test the software and familiarize yourself with it 	<ul style="list-style-type: none"> <i>InfoArchive Installation Guide</i>
Preparing the source application for archiving	<ul style="list-style-type: none"> Model the information that you want to archive Prepare data for ingestion (configure ETL tools or connectors, and generate SIPs) 	<ul style="list-style-type: none"> Applications and Data Data Organization for Table Archiving Data Organization for SIP Archiving Operational Concepts
Installing and configuring InfoArchive	<ul style="list-style-type: none"> Install a secure production configuration Upgrade an older version of InfoArchive Connect to user directory services Configure back up and restore Configure search Configure language support and customize branding 	<ul style="list-style-type: none"> <i>InfoArchive Installation Guide</i> Configuring Back Up and Restore Backing Up and Restoring Table Databases Language Support Customizing Branding
Creating and configuring InfoArchive applications	<ul style="list-style-type: none"> Create an InfoArchive application Configure archives, searches, and audits 	<ul style="list-style-type: none"> Creating an Application Using the Web Application Configuring an Audit
Archiving data	<ul style="list-style-type: none"> Start an InfoArchive application 	<ul style="list-style-type: none"> Configuring a Holding

Goal	Task	Where to Find More Information
Searching	<ul style="list-style-type: none"> Create searches Perform searches 	<ul style="list-style-type: none"> Searches in InfoArchive Search Composition for the Developer Managing Permissions
Administering	<ul style="list-style-type: none"> Manage user accounts and permissions Create storage Manage packages Manage jobs 	<ul style="list-style-type: none"> Managing User Accounts and Permissions Creating Federations, Databases and Storage Systems Using the Web Application Managing Packages Configuring Jobs
Auditing and compliance	<ul style="list-style-type: none"> Understand compliance and the retention lifecycle Create and apply retention policies Set holds Dispose of data View metrics and audit 	<ul style="list-style-type: none"> Compliance – General Concepts Compliance Related Tasks Metrics Audits
Maintaining and monitoring	<ul style="list-style-type: none"> Monitor the deployment Optimize an InfoArchive application's performance Troubleshoot and fix issues 	<ul style="list-style-type: none"> Troubleshooting topics throughout this guide

Archiving Overview

Backing Up vs. Archiving Data

Archiving data is sometimes confused with backing up data. When you back up data, you create a copy of the data in a recovery mechanism, to protect against the event that the data is accidentally destroyed or corrupted. When you archive data, you store and protect information that is not needed for everyday operations, but still must be available. You can archive the data on less expensive storage systems and retain it for the long term, to meet operational or regulatory requirements.

Backup	Archive
Offers protection: frequent snapshots of data to protect against data loss	Offers preservation: movement of data to a low-cost platform to ensure compliance and reduce costs
Copies data	Moves data off production disk
Supports operations and recovery	Supports business and compliance
Supports availability	Supports operational efficiencies
Point-in-time only	Comprehensive in nature
Poor solution for regulatory compliance	Ideal solution for regulatory compliance
Not easily searched	Easily searched
Often, old backups cannot be restored	Provides historic reference

Standards

InfoArchive uses the following standards to ingest, store, query, and retrieve data during the archiving process:

Standard	Definition	Use in InfoArchive
OAIS (ISO 14721)	A reference model that a wide variety of organizations use for archiving digital information for long-term preservation.	Specifies the format that data is ingested into, stored in, and retrieved from InfoArchive throughout the information's lifecycle
XML	A markup language for encoding information in a format that is readable by both humans-and machines. Designed for long-term, platform-independent retention.	Format for archiving structured data and metadata
XQuery	A query language for searching XML data and metadata.	Used to execute queries when users search for information

Table Archiving and SIP Archiving

An archive can be either table-based or SIP-based:

- A *table-based archive* uses a schema to ingest structured data and linked files from a table (for example, an RDBMS).

You should use table archiving to migrate structured data in application tables and linked files from transaction systems to InfoArchive with few, if any, transformations. Table archiving can reduce the up-front analysis involved with decommissioning an application and virtually eliminate the data-integrity risks associated with other archiving methods.

However, because data is stored in an aggregated manner, access is less flexible than the SIP-based archiving described below. Access is traditionally limited to query-based reports.

- A *SIP-based archive* uses submission information packages (SIPs) to ingest data from files, data records, or compound records.

For example, a customer record could include contact information (structured content), a picture of the customer (unstructured content), transactions (structured content), and a contract (unstructured content). A package bundles these different types of content together into XML data and metadata, and content files.

There are three types of SIP-based archives: file archives, data-record archives, and compound-records archives.

File Archiving

A file archive stores mostly unstructured data and its associated metadata in a single record. The data can be preserved in its original format or transformed into a more future-proof format, such as PDF-A. One record can contain several files to create sets of related information. Metadata attributes can be derived from the content itself, or associated with other systems.

File archiving is especially useful when you want to reuse data in a context that is different from the source application. For example, you can transform large print streams (such as customer statements) from print-oriented formats (such as AFP or metacode) into a PDF available on a web-based platform. Or you can reprocess image archives (typically multi-page TIFF files with limited attributes) to add document types and full-text optical character recognition (OCR). Users can find this information more easily by searching for the associated application metadata rather than directly accessing the infrastructure.

Data Record Archiving

A data-record archive stores mostly structured data in a single record. One record can contain multiple XML packets, and information from multiple systems can be drawn together according to the requirements of the project – all while preserving a complex multi-system chain-of-custody.

Data record archiving is especially useful when you want to reuse data, while reducing costs, in a context that is different from the source application. Also, data archiving is well suited for the active archiving of live systems. It typically requires additional and more business-oriented analysis of application data than table archiving does. Examples include SWIFT transactions, sales histories, and patient histories.

There are two advantages to data record archiving:

- The complex data model of the source application is transformed into a simple data model in the archive. This can reduce costs and simplify future access.
- Because there is no direct link between the source application and the data in the archive, any change in the source application does not force a change in the archive. When a change in the source application results in an update to the archive data model, InfoArchive ensures that results for searches of data sets include all records across all of the changed data.

By extending or recording metadata, customers can harmonize records and support searching and filtering across data sets.

Data archiving is used with transaction systems for active archiving of individual structured data records (for example, transaction history tables). It is also used with interaction systems (for decommissioning data and queries, optimizing searches, and advanced analytics), and with content systems. Because it presents data as single records, it is ideal for archiving information according to government requirements and legal mandates.

Compound-Record Archiving

A compound-record archive stores structured and unstructured data in a single record. The structured elements are modelled as XML, and the unstructured elements can be preserved in the original format or transformed into a more future-proof format, such as PDF-A. One record contains multiple files of related information.

Compound-record archiving is especially useful when you want to archive systems with a blend of structured information (such as wikis or blogs) with unstructured information (such as attachments), while meeting compliance rules.

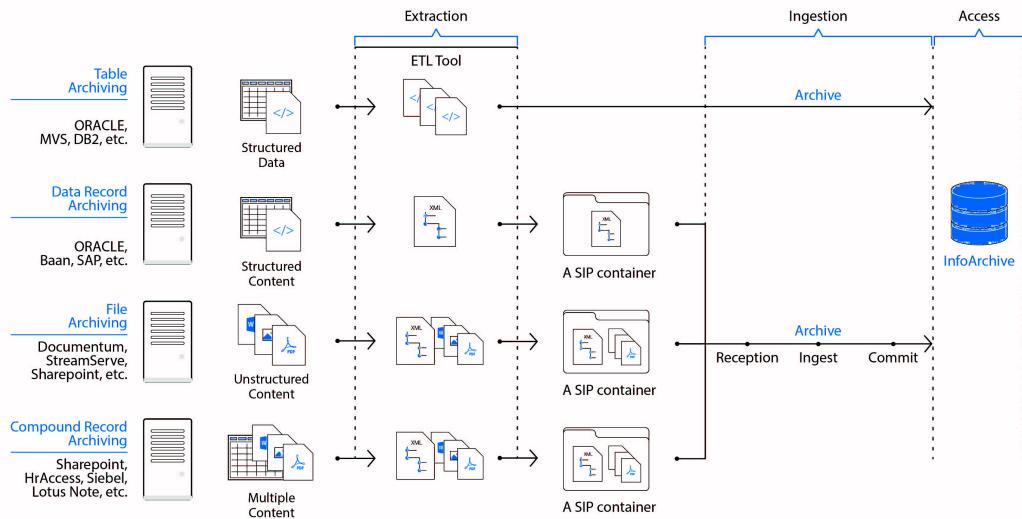
As business processes become more complex and regulations more demanding, there is a growing need to archive complex business records that contain both structured and unstructured data, which must be brought together to create a final business record. Examples include financial trades, cases and laboratory reports. Customers can retain business events as single records that they can reuse for analytics or regulatory audits. Users can search the records using pure business logic without switching from one application to another.

The Archiving Process

The following steps describe the archiving process at a high level, for both table archiving and SIP archiving. The SIP archiving steps and data formats for ingestion, storage, and retrieval comply with the OAIS data model, including the naming conventions for information packages (SIP, AIP, and DIP).

Note: While InfoArchive stores data as XML, it does not transform the data from its source format into XML. This transformation must take place before InfoArchive ingests the data. You can use whichever Extract, Transform, Load (ETL) tools that you prefer to perform this transformation, or you can use connectors that are available for Documentum, SharePoint, EMC Kazeon, and other sources, including partner tools.

1. An ETL tool or connector extracts data and metadata from the producer (the source application). For table archiving, the tool or connector converts structured data and metadata to XML, and stores unstructured data as content files. For SIP archiving, the tool or connector packages files, data records, and compound records into Submission Information Packages (SIPs). The SIPs store structured data and metadata in XML and unstructured data as content files.
2. An InfoArchive application ingests the table data and SIPs. It verifies the SIPs, and creates an Archival Information Package (AIP) from each SIP. The frequency of ingestion is based on the organization's business rules.
3. InfoArchive transfers the table data and AIPs to archived storage. It stores structured data in an xDB database and unstructured data as files in a configurable storage system (for example, a file system, EMC ECS, Amazon S3, or CAS).
4. InfoArchive sends a confirmation back to the source application, to indicate that the data has been successfully ingested.
5. Consumers (users or applications) search for information from the archive by performing a synchronous search (for immediate results) or creating an asynchronous order (a search done in the background) on the table data and AIPs.
6. InfoArchive uses XQuery to execute the query. The query examines the table data and the Archival Information Units (AIUs) that are contained in the AIPs. Any table data or AIUs that match the specified criteria are returned to the consumer in the form of a Dissemination Information Package (DIP).
7. Retention managers apply retention policies to the data and dispose of data that no longer needs to be retained.



What Type of Archiving Should I Use?

It is important to determine which of the four archive types is the best method for archiving your data:

- Table archiving
- File archiving
- Data record archiving
- Compound record archiving

There are three basic steps to making this decision:

1. Set your archiving goals
2. Understand your source application
3. Select the appropriate archiving method

Setting Archiving Goals

You should consider the following questions when setting your archiving goals:

- Do you want to decommission legacy applications or set up live archiving for active applications?
- Do you want to apply analytics to archived data, or expose the data for other uses?
- Do you need to accommodate many different applications and data formats?

Live archiving can use any of the four archive types, except for table archiving. If information aggregation and reuse is important to you, then you should consider file, data record, and compound record archiving.

Understanding the Source Application

When choosing the best archiving method for a source application, you should consider the following questions:

- What type of data do you want to archive?
- Going forward, how will users access the data?

Most organizational data is managed in one of the following system types:

System Type	Description
Transaction systems	Transaction systems have databases that hold details of past business events, such as those related to accounting processes, enterprise resource planning (ERP), enterprise asset management, and supply chain management. Transaction systems are used to maintain reference data in master files, record activities in transaction files, and store old records in transaction history tables. They can include cloud-based systems and allow many people to add small bits of detail over time.
Print stream systems	Traditionally referred to as COLD (computer output to laser disk) systems, these systems store print-stream data for long-term preservation. Most of this data consists of customer communications, but another example is green-bar reporting systems.
Content and image repositories	Content and image repositories store unstructured information and metadata, typically in their native formats. Examples include traditional enterprise content management (ECM) systems, as well as storage-based systems.
Interaction systems	Interaction systems connect users with an organization for quick access to complete information. Examples include systems that support customer relationship management (CRM) and collaborative tasks. These systems include data as well as transaction, grouping, and unstructured files.
Collaborative systems	Collaborative systems address the needs of groups of individuals to share information and communicate with each other around specific topics. These systems have all the characteristics of interaction systems, but generally cater to a less-structured approach. Notable examples include eRoom, Microsoft SharePoint, and Lotus Notes.

Selecting the Appropriate Archiving Method

The four archiving methods offered by InfoArchive are optimized based on the format of the data that is being archived, the ease of extraction and up-front analysis, and how the data is to be used after it is ingested into InfoArchive. This choice is a critical success factor for large-scale information

management programs that involve a wide range of applications. For more information about these archiving methods, see the following sections:

- [Table and SIP Archiving](#)
- [File Archiving](#)
- [Data Record Archiving](#)
- [Compound Record Archiving](#)

Use Cases

The following are examples of how you might use InfoArchive:

Archiving Goals	Source Application Analysis	Consider These Archiving Methods
Reduce IT costs for legacy applications	<p>Legacy and redundant applications have been superseded by an enterprise resource planning (ERP) system, replicated during an acquisition, or must be decommissioned as part of a business sale, closure or industry mandate. Application data must remain accessible for business reporting, audits, and compliance with data-retention regulations. Once the organization shuts down the applications, it can save on maintenance and support costs.</p>	<p>Application decommissioning</p> <p>Table archiving</p>
<p>Make production applications and infrastructure more efficient</p> <p>Compliant data retention</p>	<p>Production costs for live business applications have been escalating while performance has been degrading. Data should be periodically archived to reduce the demand on the applications, reduce storage and backup costs, and reduce licensing and administration costs. Large volumes of inactive transaction records (such as checks and statements) must be retained to meet industry regulations. Information from some completed projects (such as pharmaceutical studies, cases, and construction projects) must be archived together.</p>	<p>Live archiving</p> <p>Data record archiving</p> <p>Compound record archiving</p>
<p>Put data to work in new ways</p> <p>Remove information silos</p>	<p>New and innovative uses for data are becoming business requirements, such as advanced and predictive analysis and application modernization programs. A platform is needed for data aggregation and management, offering access to business records individually through web services, or in bulk through the Hadoop Distributed File System (HDFS).</p>	<p>Live archiving</p> <p>File archiving</p> <p>Compound record archiving</p>

Applications and Data

Architecture

For information about the following, see the *InfoArchive Installation Guide*:

- InfoArchive components
- InfoArchive directories
- Demo configurations and production configurations
- InfoArchive applications and example applications
- ETL connectors

An understanding of these concepts is necessary for much of the following material.

How Data is Ingested

Before you ingest data into InfoArchive from a source application, you must extract the data from the source application using an ETL tool or connector. The three ETL functions are as follows:

Extract	Extract data in its natural form from a source application, whether it is in a table or file format.
Transform	Apply rules or functions to convert source data into XML. Some systems can natively export in XML.
Load	Ingest the transformed XML data into the xDB. Digital data storage (DDS) provides a toolset for loading data with assigned retention and metadata.

You use one of the InfoArchive connectors created by OpenText, a partner or third-party tool, or a tool of your own to perform the first two functions, extract and transform. InfoArchive is open to any ETL tool that can transform data into XML or a SIP. Some applications have a built in facility to export to XML.

The Load (ingestion) function is always performed by an InfoArchive application, and you must configure the InfoArchive application before ingestion.

How Data is Stored

How data is stored on ingestion depends on whether the data is structured or unstructured:

- *Structured data*, such as an AIU or record, is stored in an xDB database. A minimum of one xDB federation and xDB database must be configured. The xDB database must not, however, be the same database that is configured for Spring Data.
- *Unstructured data*, such as an image file, is stored in a configurable storage system (for example, a file system, ECS, S3, or CAS).

When an InfoArchive application ingests data, it ingests the data into the InfoArchive application's space, which is a collection of libraries in xDB databases (called SpaceRootXdbLibraires) and folders in FileSystemRoots (called SpaceRootFolders).

The following list illustrates the hierarchy for storage configuration, with the most important attributes per item:

- XdbFederation (name, bootstrapUrl, superUserPassword)
 - XdbDatabase (name, adminPassword)
- FileSystemRoot (name, path)
- Other storage systems (for example, ECS, S3, or CAS)
- Tenant (name)
 - Application (name, ...)
 - Space (name)
 - SpaceRootXdbLibrary* (name, XdbDatabase)
 - SpaceRootFolder* (path, FileSystemRoot)
 - Bucket
 - ...

How Data is Searched

Searching is the main way that a user accesses data that InfoArchive has ingested. Some user roles can execute searches for applications that they have permissions to see.

Searches are slightly different for table-based archives and SIP-based archives:

- For a table-based archive, a search can be associated with a schema or table. Table-based searches require a query, built on an XQuery template, as well as the schema or table that the search will be executed against.
- For a SIP-based archive, a search is associated with an archive information collection (AIC) and a query configuration. SIP-based searches require the name of the AIC and the query configuration, which determines the criteria and results for the search.

The Developer is responsible for defining a search which includes which criteria to use, which fields to show, and in the case of tables, how to query the data from the database. For example, when creating a table-based search form, the Developer specifies a schema and a table within the schema.

For table-based searches, the search form designer must understand XQuery and how it is related to search design. The search form designer needs to ensure the following:

- The query is valid
- If an XQuery is defined, all of the mandatory parameters that do not have defaults must be supplied by the form
- The correct binding for elements is used in the result set

A SIP-based search uses two queries:

- The first query returns the appropriate AIP based on the partition key. This information appears in the wizard when selecting the field for the search criteria.
- The second query filters the suitable AIU in the selected AIP list. Search performance can be improved by using and filling at least one partition key in the search form.

In the IA Web App, search forms are customizable. A Developer can add or remove fields as required.

Searches provide a rich graphical user interface for both criteria and results. In the following image, the first screen captures a typical search form, which is designed by the Developer. The End User enters relevant information required to execute a search. The second screen captures the search results, which are also designed by the Developer:

The screenshot shows a 'Trade Search' interface. At the top, there is a search bar labeled 'Search'. Below it, there are three input fields: 'Trader Name:' (dropdown), 'Customer ID' (text input with placeholder 'Enter number between 400'), and 'Customer Last Name:' (dropdown). Underneath these, there are four sections: 'Trade Date' (date range inputs 'From: yyyy-mm-dd' and 'To: yyyy-mm-dd'), 'TradeID' (text input with placeholder 'Enter number between 100'), 'Exchange' (radio buttons for 'NASDAQ' and 'NYSE'), and 'Ticker' (dropdown menu listing 'APPL', 'BAC', 'CSCO', 'EMC', and 'F'). Below these sections is a table with trade data:

	Trade...	Trade ID	Custo...	Custo...	Custo...	Trade...	Excha...	Ticker	Price
<input type="checkbox"/>	2014-03-27...	10007	4007	Bridges	9993	Chloe Padi...	NYSE	EMC	95.83
<input type="checkbox"/>	2014-03-27...	10007	4007	Bridges	9993	Chloe Padi...	NYSE	EMC	95.83
<input type="checkbox"/>	2013-09-24...	10028	4028	Suarez	9972	Aspen Mo...	NYSE	EMC	59.79
<input type="checkbox"/>	2014-08-19...	10049	4049	Mack	9951	Bryant	NYSE	EMC	85.66

To the right of the table, there is a 'Details' panel with the following information:

Projected Disposition Date	Feb 27, 2002 11:00:00 PM
Retention Policy	Yes
On Hold	No
Longest Retention	Trades-policy

Result data might be simple text, a link to a nested search, downloadable content, or viewable content. The search results pages are customizable. A Developer can add or remove fields as required.

Searches can sometimes run in the background for a variety of reasons. For more information about running a search in the background, see [Background Search](#).

How InfoArchive is Configured for Data

Users with the Administrator and Developer roles perform the configuration as follows:

- The Administrator configures one or more xDB federations and xDB databases for structured data.
- The Administrator configures one or more storage systems for unstructured data.
- The Developer extracts data from a source application using an ETL tool or connector.
- The Developer configures an InfoArchive application.

5. The Developer configures a space for the InfoArchive application with a SpaceRootXdbLibrary in the XdbDatabase and slices (folder, buckets, etc.) of storage in one or more configured storage systems.
6. The Developer starts ingestion.
7. The Developer creates searches that End Users can use to access ingested data.

Data Organization for Table Archiving

Extracted and Converted Data

An ETL tool or connector performs the following tasks:

1. Extracts data and metadata from the producer (the source application).
2. Converts structured data and metadata to XML.
3. Stores unstructured data in the corresponding InfoArchive application's blobs directory.

For example, for an Ant script application, see `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/Tickets/blobs`. For a declarative configuration application, see `<INFOARCHIVE_ROOT>/examples/applications/Tickets/data/blobs`.

Ant script applications and declarative configuration applications use the following files to organize data:

File Type	Used by Ant Script Applications?	Used by Declarative Configuration Applications?
Metadata file	✓	✓
Table data files	✓	✓
Build properties file	✓	
Build file	✓	
Configuration YAML file		✓

Metadata File

The metadata file describes the data to be ingested. It defines the following:

- The schema name
- The table count

- The table metadata
- Whether to use indexing and encryption (the type of encryption to use is defined in the `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/build.properties` file)

The file name for the metadata file is as follows:

- For Ant script applications, the file name is `metadata.xml`
- For declarative configuration applications, the file name is `database-<APPLICATION_NAME>SqlDb.xml` (for example, `database-BaseballSqlDb.xml`)

There is one metadata file for each InfoArchive application. The following are examples of the metadata file:

- For an Ant script application: `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/Patent/tables/metadata.xml`.
- For a declarative configuration application: `<INFOARCHIVE_ROOT>/examples/applications/Patent/config/data-model-config/database-PatentSqlDb.xml`.

You can see the schema name in the `<name>` element, and the number of tables in the `<tableCount>` element. For every table, the file contains one `<tableMetadata>` element. Each table has a `<recordCount>` element to specify the number of records, which InfoArchive uses for validation. The `<index>` element specifies whether to index the data.

Table Data Files

For ingestion, an application requires table data from source applications to be transformed to XML format in files that do not exceed a few hundred megabytes in size. Larger files can cause indexing to take a long time, or in some cases, stop unexpectedly. The tables must have at least one row. Individual tables can span multiple table data files, and the names for the additional files are appended with a dash and sequence number. The schema and table names in the table data files must match the schema and table names in the metadata file.

You can find examples of table data files in the following locations:

- For an Ant script application, `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/Tickets/tables`
- For a declarative configuration application, `<INFOARCHIVE_ROOT>/examples/applications/Tickets/data/TICKETS`

The table data files are all of the files stored there, except for the `metadata.xml` file in the case of an Ant script application. The files `TICKETS-TICKET_ATTACHMENT.xml` and `TICKETS-TICKET_ATTACHMENT-2.xml` are an example of table data spanning multiple files.

Build Properties File

Each Ant script application has its own build properties file (for example, `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/Tickets/build.properties`). This should not be confused with the build properties file that is used by all Ant script applications and declarative

configuration applications (`<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/build.properties`).

The application-specific build properties file defines variable information used by Ant scripts on ingestion, including the following:

- Application name, category and description
- Authentication information
- Encryption information

Authentication is completed using the gateway, username, and password properties. The gateway is the URL of IAWA, and the user must be an Administrator or Developer.

Build File

Each Ant script application has its own build file (for example, `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/Tickets/build.xml`). The build file defines the Ant project and targets. For table-based archives, the build file usually references the `build-table.xml` file that is located in the `<INFOARCHIVE_ROOT>/tools` directory. This file controls the indexing process.

The Configuration YAML File

Each declarative configuration application has its own configuration YAML file (for example, `<INFOARCHIVE_ROOT>/examples/applications/Tickets/config/configuration.yml`). The configuration YAML file describes the application, including the following:

- Application name, category and description
- Encryption information

You can find an example of the configuration YAML file in the `<INFOARCHIVE_ROOT>/examples/applications/Tickets/config` directory. The `install.iashell` file defines the IAShell commands that are run when you install the application.

Authentication for Declarative Configuration Applications

Declarative configuration applications complete authentication using the settings in the `<INFOARCHIVE_ROOT>/config/iashell/application.properties` file.

Searches

You can create and add predefined searches during ingestion, including XForm, XQuery and supporting files. Searches require XForm definitions, XQuery definitions, and other files.

To create a search quickly, you can modify and add a search created for a different InfoArchive application, such as a example application.

You can also use XQuery modules to build utility functions once and reuse them in multiple searches. This avoids redundant code and enables the writing of modular XQuery that can be maintained efficiently. XQuery modules are predefined modules that contain XQuery code, which table searches can use to retrieve data. You can add predefined XQuery functions to XQuery modules and import them during ingestion.

When you create a search for a table-based archive, you use XQuery code to retrieve data. You might want to have helper functions defined. Instead of manually typing or pasting the command every time you run the XQuery for a search, you can put it into an XQuery module, which is then stored in the xDB database. After the module is ingested, you will not have to define the command again because it is known to xDB.

InfoArchive provides the ability to export data from search results. You can add exports during ingestion by creating a pipeline, configuration, and transformation.

The Indexing Process for Ant Script Applications

The indexing process for Ant script applications for table archiving is usually controlled by the `build-table.xml` file, which is located in the `<INFOARCHIVE_ROOT>/examples` directory. The following target element specifies the indexing process:

```
<target name="ingest" depends="ingest-tables, run-chain-of-custody-if-enabled-for-ingest, index-build, create-searches, create-xquery-modules" />
```

After the ingestion process finishes, the Ant script starts indexing the ingested data.

If multiple ingestion actions are required, to ingest data in the same tables of a new application in relatively short time, you should disable the automatic indexing and perform the indexing manually once all of the table data has been ingested. You can do this as follows:

1. In a text editor, open `build-table.xml`, find the `target` element, and remove the `index-build` step.
2. Perform the ingest actions with the regular Ant commands.
3. Once the table data has been ingested, start the background indexing by running the following command:

```
ant index-build
```

An alternative is to create and update the indexes as the data is being ingested. However, if the data set being ingested is large, this is not advised as it has performance implications. To create and update indexes during ingestion for your application, edit the `metadata.xml` file in the `tables` directory of the application, and add the following line below the `<defaultSchema>...</defaultSchema>` line:

```
<indexingOnIngest>true</indexingOnIngest>
```

If you do not use one of the methods described above, and you perform multiple ingestions after each other, the ingestion process might fail with the following error: "Cannot ingest new table data while index creation jobs are underway". This means that the system has determined that an indexing job is indexing data while the same tables are being updated as a result of an ingestion process.

The Indexing Process for Declarative Configuration Applications

After the ingestion process, the IA Shell script starts indexing the ingested data. If multiple ingestion actions are required, to ingest data in the same tables of a new application in a relatively short time, disable the automatic indexing and start the indexing manually once all of the table data has been ingested:

1. In the <INFOARCHIVE_ROOT>/config/iashell/application.properties file, set the default.enableIndexBuildInBatch to false.
Alternately, override this property by adding it to the local_iashell.properties file of the application directory.
2. Run the ingest commands manually until all data has been loaded.
3. Manually start indexing by running the following command from a connected IA Shell session:

```
index-build--d path/to/application/database
```

For the example applications, the actual command can be viewed in the install.iashell script (for example, index-build --d applications/Tickets/databases/Tickets -sql-db).

An alternative is to create and update the indexes as the data is being ingested. However, if the data set being ingested is large, this is not advised as it has performance implications. To create and update indexes during ingestion for your application, edit the metadata file and add the following line below the <defaultSchema>...</defaultSchema> line:

```
<indexingOnIngest>true</indexingOnIngest>
```

If you do not use one of the methods described above, and you perform multiple ingestions after each other, the ingestion process might fail with the following error: "Cannot ingest new table data while index creation jobs are underway". This means that the system has determined that an indexing job is indexing data while the same tables are being updated as a result of an ingestion process.

Configuring InfoArchive Server Table Indexing

This section describes the available configuration settings for the asynchronous (post-ingest) table indexing functionality, as well as how to tweak and fine-tune this feature with regards to performance. Because this functionality builds on top of both the Job, Order Item and Batch framework of InfoArchive, corresponding configuration includes both system-wide and table indexing-specific settings.

System-Wide Configuration Settings

These settings apply to all (batch) order item operations per individual IA Server node, in which the corresponding functionality is enabled. As such, although changes to these settings may affect the performance and/or scalability of the table indexing logic, it also affects all other order item and batch item executions related to other type of operations. Furthermore, in a deployment with multiple

InfoArchive nodes that can run order items and/or batch items, these numbers stack, which impacts the end result of any changes made to these configuration settings.

background.orderItems.numberOfThreads: The number of order items that are allowed to run in parallel on this InfoArchive node.

background.orderItems.pollingDelay: Frequency (interval) at which the order item framework on this InfoArchive node checks for ready-to-execute order items (in milliseconds).

background.batchItems.numberOfThreads: The number of batch items that are allowed to run in parallel on this InfoArchive node.

background.batchItems.pollingDelay: Frequency (interval) at which the batch item framework on this InfoArchive node checks for ready-to-execute batch items (in milliseconds).

Table Indexing Batch-Specific Configuration Settings

These settings apply to the batch framework in general, and allow operation-specific overrides for three generic criteria. For convenience purposes, each criterion must specify an all instance of which the value is used as a fallback value in case no operation-specific value is configured. To explicitly configure these settings for the table indexing logic, a value for the tableIndexing instance should be specified.

batch.interval.[all | tableIndexing]: The polling interval at which a batch order item checks for status of its underlying batch items (this value * # of unfinished batches):

- Increase this value to reduce the overhead of performing such checks at the expense of making the batch operation feel more sluggish/less responsive.
- Decrease this value to have a more "responsive" batch operation at the expense of more frequent status checks.

batch.size.[all | tableIndexing]: The number of indexes to process per batch. Increasing this value reduces the number of batches created and, therefore, results in less batch management overhead at the expense of potential scalability. For table indexing, the default value is set to 1, which is likely appropriate for most deployments but, under certain circumstances, it may need to be set slightly higher to improve scalability. Refer to [Tweaking and Fine-Tuning](#) for further information.

batch.chunk.[all | tableIndexing]: In general, this controls the amount of items to process within a single transaction, therefore, resulting in multiple transactions per batch. However, the actual meaning of this setting can differ per operation. For table indexing, this property controls the minimum amount of time (in seconds) to spend on indexing per transaction. Primarily optimizes performance for non-applicable (empty) indexes and does not have much effect otherwise. The default value for table indexing is set to 3 and there should usually be no reason to change this at all.

Table Indexing Job Configuration Settings

The Table Indexing Job Definition itself also has one configuration setting that controls how many batches should be created per index. This property is called `batchesPerIndex`.

This property is by default set to 1, and it effectively controls how many 'duplicates' of each batch item are created. Each batch item that processes an index does so one document at a time. The

internal implementation allows this to be done in parallel, so that we can have multiple batches run in parallel, each indexing a different document for the same index. This way, the construction of a particular index can be significantly sped up, as long as all related batch items do in fact run in parallel. Once all documents have been processed for a particular index, any batch item processing that index will no longer have anything to do for that index, even if it has not even started yet.

For example, should one such batch item become queued up and not start until after the other 'duplicate' batch items have already finished processing the index, the queued up batch item will finish processing that index immediately, as soon as it starts running. It will then not have contributed to constructing the index. If it was only assigned this one index, it essentially becomes a no-op batch item, which did not do anything, but does need to be managed by the framework.

This setting also applies to batch items that process multiple indexes, in which case, it is possible certain batch items may have nothing to do for a few indexes, but can still participate in the construction of other indexes.

Tweaking and Fine-Tuning

Scalability is accomplished by being able to run as many activities as possible in parallel in as many parallel threads as possible. With regards to (any operation of) the InfoArchive Batch framework, this essentially means having as many batch items as possible run in parallel, spread across as many InfoArchive nodes as possible. Having more batch items than can be run in parallel can still be beneficial, though having more than twice as many batch items than threads to run them in parallel will face quickly diminishing returns.

The work that table indexing can do in parallel is processing a single document to construct an index. Unfortunately, the underlying API does not expose access to individual documents (remaining) to be indexed, so the unit we can batch on is individual indexes instead. Luckily we can create multiple 'duplicate' batches that all process the same index, by each processing different documents (which is taken care of by the xDB implementation automatically).

As such, the factors that define scalability for table indexing are as follows:

- The number of InfoArchive nodes participating in the execution of batch items.
- The number of threads per InfoArchive node available to execute batch items.
- The number of indexes to process.
- The number of indexes per batch item to process.
- The number of 'duplicate' batch items, each processing the same index(es).

To calculate the number of indexes that will be created, the following information is important:

- For each schema, two internal (row@id and *@ref) indexes will be created.
- For each table, a single multipath index will be created.
- Each table may or may not define additional path value indexes.

Be aware of the following caveats:

- Indexes are created for each individual schema library. As such, the number of indexes calculated using the information above needs to be multiplied by the number of schema libraries. Currently, each schema always has only a single library, but this may change in future releases.
- Although perhaps uncommon, it is possible for a table application to define more than one schema. To optimize cross-schema searches in such applications, it is necessary to apply all table indexes of all schemas to all schemas of a table application. However, because such cross-schema indexes are generally not applicable, creating them will usually be fairly cheap. Because of this, such cross schema indexes are always created in bulk using a dedicated single non-duplicated batch order item. This way, there is no need to take this behavior into account when determining the settings for optimal scalability.

Keep the following points in mind:

- The default configuration settings will create one batch per index.
- Batches that create a multipath index may take longer than those that create a path value index, because these are generally more complex and, therefore, expensive to construct.
- Applications that contain a small number of tables/indexes are less scalable when using default settings, because only a few batches will be created. There will, therefore, be relatively few threads that can do actual work in parallel for such applications.
 - It may be beneficial to have the data being processed in parallel by creating multiple 'duplicate' batches, resulting in each index being processed by more than one batch in parallel. This is especially true for large tables (with a lot of data)
 - Duplicate batches each processing the same index can only actually work in parallel when there are enough individual documents to be indexed. This will require the table to be ingested using multiple documents. Ingesting a large table using a single document does not allow for indexing it in parallel.
- Creating duplicate batches only helps when those duplicate batches actually run in parallel.
 - When there are (far) less threads available than the number of created batches, efficiency will be reduced, possibly up to the point where some of the duplicate batches no longer contribute to scalability at all.
 - Even if a duplicate batch becomes functionally pointless because of a lack of threads to run it in parallel, it will not break the indexing process. It will introduce a minor management overhead per batch though.
- it is perfectly fine to combine creating batches that process multiple indexes with create 'duplicate' batches by means of the `batchesPerIndex` job property. Although in this case the name of this setting may not really convey its effect, it may still be beneficial to do this under certain circumstances.

Doing this could be a means to even more effectively balance the load in case of a lot of data and unbalanced indexes. In case the creation of some indexes is very cheap, while the creation of others is very expensive, creating a single batch per index may not be very efficient. The majority of batches could finish quickly, while a few batches could take a lot longer, but not being able to run in parallel, wasting a lot of potential scalability. Having each batch process multiple indexes and duplicate those batches by an equal amount, the process may take less time overall because it can be run more in parallel and thus optimizes scalability.

The (significant) downside of doing this is that it makes understanding/interpreting the progress reporting a lot harder if not impossible. Progress can only be reported by the number of batches that have already finished versus how many batches have not. When the majority or all batches are doing the same work in parallel, they either all finish at roughly the same time, or the first half will do all the work and the second half will have practically nothing to do afterwards.

This is, however, a matter of both preference and prioritization. It is a balancing act between the ability to track progress and somewhat predict remaining time versus achieving optimal scalability and have the operation finish as quickly as possible, scaling out horizontally as/if needed.

Full-Text Indexing vs. Value Indexing

You can configure InfoArchive applications to use full-text (multi-path) indexing only, value (path value) indexing only, or both. The following code demonstrates how to edit the metadata file to configure the LAST_NAME column for value indexing only:

```
<column>
  <name>LAST_NAME</name>
  <ordinal>6</ordinal>
  <type>VARCHAR</type>
  <typeLength>32</typeLength>
  <indexing>VALUE</indexing>
  <encrypt>false</encrypt>
</column>
```

The following table lists the elements and values that you can use to configure the column index for full-text and value indexing:

Description	Elements
Only index full-text search	<indexing>FULL_TEXT</indexing>
Only index value search	<indexing>VALUE</indexing>
Index both full-text and value search	<indexing>FULL_TEXT_AND_VALUE</indexing>

Full-text indexing requires much more storage space than value indexing, so you should only use the full-text option when absolutely required.

By default, the following example InfoArchive applications use full-text indexing for the following searches:

- The Patent application's Fulltext Name Search search. The NAME column of the PATENT_ASSIGNEES table is indexed for a full-text search only.
- The Baseball application's Player Search - Last Name Fulltext search. The NAMELAST column of the MASTER table is indexed for both full-text and value search because this column is used in other searches for value comparisons.

By default, all other searches in the example InfoArchive applications for application decommissioning (Baseball, Order_Management, Patent, and Tickets) use only value comparisons.

Installation and Ingestion

After you have set up the configuration files for an Ant script application or declarative configuration application for table archiving, you can install the application. For Ant script applications, you run Ant to install the application. For declarative configuration applications, you run an installation script (`install.bat` in Windows and `install.sh` in Linux), which runs the `install.iashell` script with the `iashell` command.

When you install an application, IA Shell performs the following four steps:

1. Connects to the system
2. Imports the configuration of the application, either to create and initialize the application or to update its existing configuration
3. Imports the tables (if you run the application multiple times, the data is imported multiple times)
4. Starts the background indexing (only when the installation script is run for the first time for an application)

During installation, most applications for table archiving create the following:

- A database
- A space
- Crypto objects
- Stores
- Other objects (for example, XQuery modules)

After an application creates all the required configuration objects, it can perform an initial ingestion. You can ingest more data later.

You should also consider restricting access to the application, in case it contains sensitive data. For more information about permissions, see [Managing User Accounts and Permissions](#).

Data Organization for SIP Archiving

Extracted and Converted Data

An ETL tool or connector performs the following tasks:

1. Extracts data and metadata from the producer (the source application)
2. Packages structured data and metadata, and unstructured content files, in SIPs
3. Stores the SIPs in the corresponding InfoArchive application's `sips` directory (for an Ant script application) or `data` directory (for a declarative configuration application)

For SIP archiving, InfoArchive uses several information formats and standards specified by OAIS:

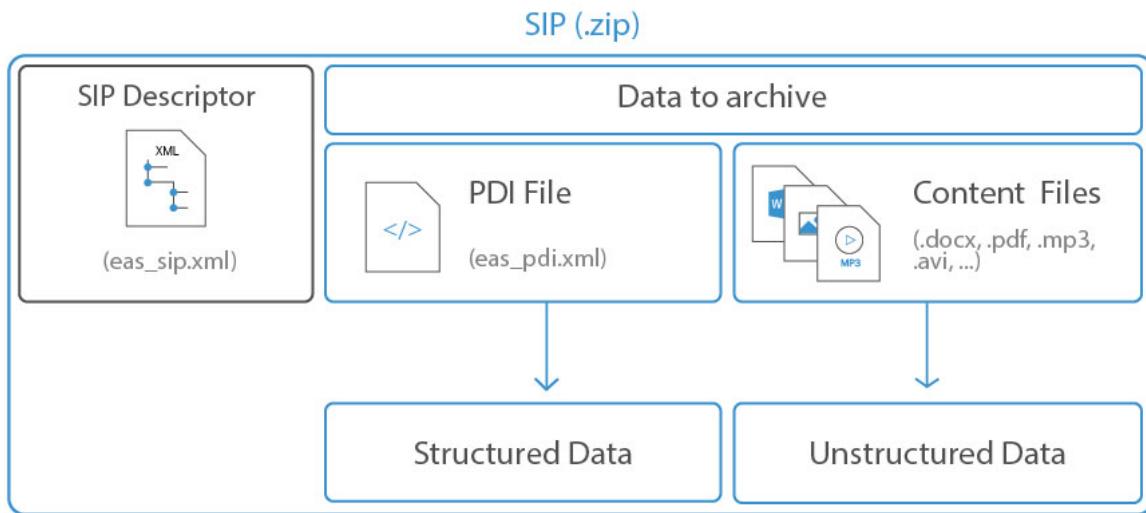
- SIP
- AIP
- AIU
- Holding
- Data submission session
- SIP descriptor
- PDI file
- Schema file

InfoArchive also uses several additional files for SIP archiving. These files are different for Ant script applications and declarative configuration applications.

SIPs

A SIP is a package of data that an InfoArchive application ingests. A SIP must be compressed into ZIP format and have the following files at the root level:

SIP descriptor	The SIP descriptor (<code>eas_sip.xml</code>) contains metadata that describes the data being archived. This file must be in accordance with the SIP schema. There can be only one SIP descriptor in a SIP. The SIP descriptor is part of the OAIS standard. For more information, see SIP Descriptor .
PDI file	The Preservation Description Information (PDI) file (<code>eas_pdi.xml</code>) contains structured data in AIU format for archiving. You must create a schema file that specifies the structure of this file. There can be only one PDI file in a SIP. The PDI file is part of the OAIS standard. For more information, see PDI File .
Content files	Optionally, a SIP can contain any related content files (for example, PDF files).



InfoArchive can archive any type of data produced from any source application as long as the data is packaged into SIPs that meet the file structure and format requirements. However, InfoArchive is not responsible for generating SIPs from the source application's data. The source application or an ETL tool or connector must produce them. You can use a file transfer program of your choice to move the SIPs from where they are generated to where an InfoArchive application can ingest them.

The PDI file contains the structured data, and each AIU (piece of structured data) is essentially a record.

Best Practices for SIPs

You should use the following best practices when creating SIPs:

- If a SIP contains unstructured data, it must be stored in the root level of the SIP. Any unstructured content that is stored in a sub-directory or in another path will not be ingested and an error will occur.

For an example of a SIP that includes unstructured data, see <INFOARCHIVE_ROOT>/examples/legacy-ant-applications/PhoneCalls/sips/PhoneCallsSample-2001.zip.

- When you create a SIP ZIP file, ensure that the archived file names are in UTF-8 encoding. Otherwise, you might experience issues ingesting the data.
- The following is an example of a command that you can use to create a SIP:

```
"C:\Program Files\7-Zip\7z.exe" a -tzip -mchu test.zip  
C:\infoarchive\examples\legacy-ant-applications\Invoices\sips\can_be_ingested\*
```

The correct options to pass to the tool to create a ZIP depend on the archive tool that you are using. The above is an example using the 7-Zip tool.

- Date and time values should explicitly include the time zone information to avoid time conversion problems for searches. For example, in the following text, the bold text specifies one hour offset ahead of UTC time:

2002-11-18T16:22:04.104**+01:00**

- Depending on SIP size and its content, you should configure an optimal ingestion mode, xDB indexes, and partitioning keys before you start ingesting data.

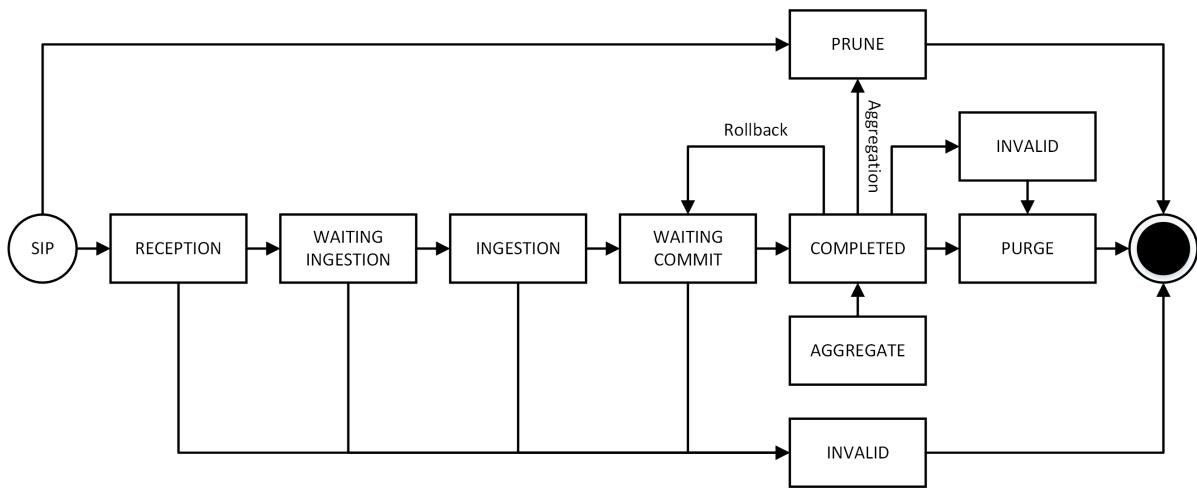
AIPs

When InfoArchive ingests and verifies a SIP, it creates an AIP from the SIP. AIPs are packages of information that can contain structured data, metadata, and unstructured data. The structured data and metadata are represented as XML.

The data and metadata elements can be extracted directly from the source application, derived from other systems, or programmatically constructed. The ability to maintain separate data elements in an AIP allows InfoArchive to balance the requirements to maintain exacting standards around chain-of-custody with the desire to build richer data sets than those that existed in the original applications. For example, raw transaction history data can be extracted, modeled as XML, and verified as 100% accurate and complete for chain-of-custody purposes. At the same time, additional information from extended systems can be made part of the AIU in another data element, making the AIU more usable without compromise.

The following diagram illustrates the process whereby a SIP is received, ingested and transformed into an AIP. The following notes provide further clarification about the diagram:

- An AIP reaches the COMPLETED stage when all AIPs that are part of the same data submission session (DSS) have been ingested. An AIP is only searchable once it has reached this stage. For more information about DSS, see [Data Submission Sessions](#).
- The PRUNE phase is only used when you have configured the xDB ingestion mode to use AGGREGATION mode. When the aggregate is closed, all AIP children are moved to the PRUNE phase and the aggregate moves to the phase COMPLETED.
- Once the retention needs on the data have been met, the AIP can be disposed.



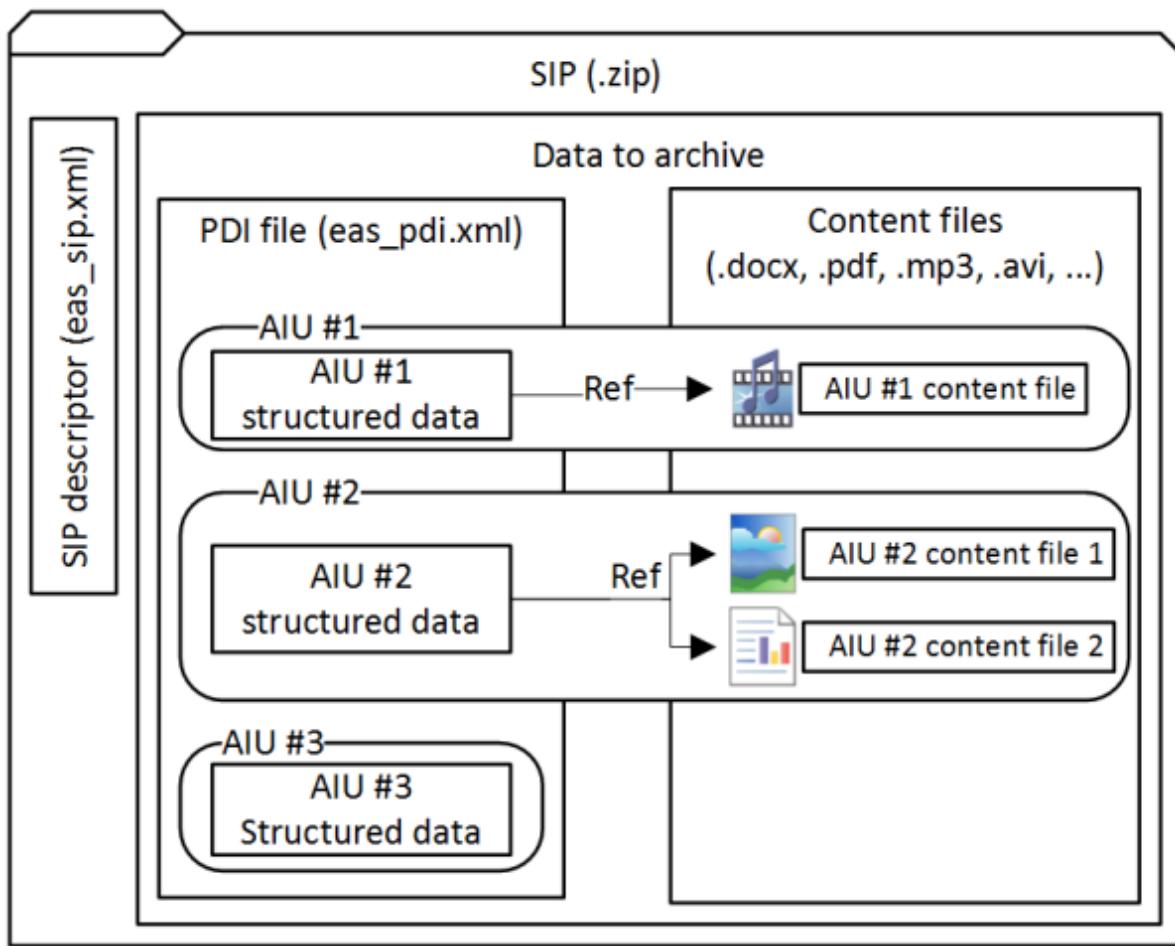
AIUs

An AIU is the smallest archival unit of an information package. Each AIU corresponds to a record or item of the archived data (for example, a single customer order, a patient profile, or a financial transaction record).

The PDI file in a SIP describes all of the AIUs in the package. An AIU in a PDI file consists of an XML block containing structured data and, optionally, references to one or more associated unstructured content files.

For example, in the following diagram:

- AIU #1 is described by structured data in the PDI file, with a reference to one content file.
- AIU #2 is described by structured data in the PDI file, with a reference to two content files.
- AIU #3 only contains structured data stored in the PDI file, with no content files.



In a PDI file, each `<call>` element represents one of the AIUs. For an example, see the `eas_sip.xml` file in the `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/PhoneCalls/sips/PhoneCallsSample-2001.zip` file.

Holdings

A holding is essentially a basic archive application, a logical destination where data that shares common characteristics is archived. Some examples of common characteristics include:

- Data from the same source application
- Data in the same format (for example, audio recordings)
- The same type of data (for example, communication through email, chat, and faxes)
- Data that belongs to the same business entity

Holdings are used exclusively with SIP-based archives. Table-based archives do not use holdings.

The SIP descriptor includes the name of the holding to be used. Typically, for the example InfoArchive applications, the holding name is the same as the application name. However, an InfoArchive application can use multiple holdings. Multiple holdings can also exist for a single data type, which means that you are able to apply different access rights or use a different storage area.

A holding is the central configuration object in SIP archiving. The following is defined in a holding:

- Storage area
- Retention
- Ingestion sequence
- The AIP mode and xDB mode being used

When you create a holding, you should consider the types of data that will be archived as well as the data segregation and isolation restrictions.

If a holding is properly designed, then it should be able to handle a lot of data. The ingestion speed is very sensitive to a holding configuration. If performance is poor, it is most likely because of an improperly designed holding configuration. With the InfoArchive Holding Configuration Wizard, you can configure a holding that is optimized by default and uses indexes efficiently during ingestion and searches.

For more information about holdings, see [Configuring a Holding](#).

Data Submission Sessions

Sometimes information to be archived cannot be packaged in a single SIP because of the following:

- ZIP limitations: a ZIP file has an upward size limit
- File transfer limitations: the FTP that you are using might have a built-in limitation for file size
- Time limitations: a single, large ZIP file might cause a connection to time out

A data submission session (DSS) associates multiple SIPs together. This is useful when you want to ingest SIPs together in a single batch, which is a process called asynchronous ingestion. A business application can produce several SIPs belonging to the same DSS.

The DSS ID is specified in the SIP descriptor. InfoArchive uses the DSS ID to determine that the SIPs belong to the same DSS.

SIP Descriptor

The following table lists the elements that are contained in the SIP descriptor, in the order that they appear. For an example, see the `eas_sip.xml` file in the `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/PhoneCalls/sips/PhoneCallsSample-2001.zip` file.

Element	Description
<code><dss></code>	All of the SIPs that belong to a particular DSS have the same values in the <code><dss></code> block of elements.
<code><holding></code>	Name of the holding where the SIP must be archived.
<code><id></code>	DSS identifier assigned by the business application producing the SIP.

Element	Description
<pdi_schema>	URN of the schema applied by the PDI file. You should put the version of the schema in the URN, as this is common XML practice.
<pdi_schema_version>	Not recommended for use. Can specify a given version of the schema when this information is not included in the URN. Included for alignment with the xsd:schema standard.
<production_date>	The <production_date> element that is contained in the <dss> element is the production date of the DSS.
<base_retention_date>	Base date to be considered for computing the retention date.
<producer>	Code of the business application producing the SIP.
<entity>	Code of the business entity that owns the data contained in the SIP.
<priority>	Ingestion sub-priority of the SIP.
<application>	Code of the business application producing the data contained in the PDI file.
<retention_class>	An alias that can be used to associate a retention policy that will be applied to the SIP on ingestion. The retention class overrides any default that is specified by the holding. The holding must specify the name of the retention class and map to zero or more retention policies that would be applied.
<production_date>	The <production_date> element that is contained outside of the <dss> element is the production date of the SIP.
<seqno>	Sequence number of the SIP within the DSS that it belongs to. It is common to have only one SIP in a DSS.
<is_last>	Boolean indicating whether this SIP is the last SIP of the DSS.
<aiu_count>	Number of AIUs contained in the PDI file. Must match the actual AIU count or an ingestion error occurs.
<page_count>	Reserved for future use. Always set this to 0.
<pdi_hash>	Optional element that specifies the encoded hash value of the PDI file.

SIPs and Data Submission Sessions

InfoArchive uses elements in the SIP descriptor to create the DSS external identifier and SIP external identifier.

For an example, compare the table below to the `eas_sip.xml` file in the `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/PhoneCalls/sips/PhoneCallsSample-2001.zip` file.

Type of Identifier	Elements Used	Example Element Values	Example Identifier
DSS external identifier	<holding> <id> <producer>	<holding>PhoneCalls </holding> <id>1000000</id> <producer>CC </producer>	PhoneCallsCC1000000
SIP external identifier	<holding> <id> <producer> <seqno>	<holding>PhoneCalls </holding> <id>1000000</id> <producer>CC </producer> <seqno>1</seqno>	PhoneCallsCC10000001

InfoArchive is insensitive to the order that it receives SIPs that belong to the same DSS. InfoArchive can receive and ingest the SIPs in any order, and can also do so concurrently.

InfoArchive provides a native commit and rollback at the DSS level. If there is an issue with one of the SIPs in the DSS, then the DSS can be rolled back in a single transaction. If there are no issues with the SIPs in the DSS, and the SIPs have all been ingested, then the DSS can be committed in a single transaction.

For a DSS to be valid, the following must be true:

- There must be no gaps in the <seqno> values. For example, if there are SIPs with the <seqno> values 1, 2, and 4, then there must be also a SIP with the value 3 for InfoArchive to commit the DSS.
- The last SIP in the DSS must have <is_last> set to true, and all other SIPs in the DSS must have <is_last> set to false.
- All of the SIPs that belong to a particular DSS must have the same values in the <dss> block of elements (for example, they must have the same value for <id>).

PDI File and Schema File

You must specify the structure of the PDI file by creating a schema file. The file name for a schema file is as follows:

- For Ant script applications, pdi-schema.xsd
- For declarative configuration applications, pdiSchema-urn<TARGET_NAMESPACE>.1.0.xsd (for example, pdiSchema-urnAcmeCorpXsdCertificates.1.0.xsd)

The schema file should be driven by business requirements. InfoArchive uses the schema file to validate the XML in the PDI file.

You typically store the schema file as follows:

- For Ant script applications, in an InfoArchive application's resources\content directory
- For declarative configuration applications, <INFOARCHIVE_ROOT>\examples\applications\<APPLICATION_NAME>\config directory

You should use an XML editor to define and analyze the schema file.

For examples of the schema file and a corresponding PDI file, see the following:

- For Ant script applications: <INFOARCHIVE_ROOT>/examples/legacy-ant-applications/PhoneCalls/resources/content/pdi-schema.xsd and the eas_pdi.xml file in the <INFOARCHIVE_ROOT>/examples/legacy-ant-applications/PhoneCalls/sips/PhoneCallsSample-2001.zip file
- For declarative configuration applications: <INFOARCHIVE_ROOT>/examples/applications/PhoneCalls/config/data-model-config/pdiSchema-urnEasSamplesEnXsdPhonecalls.1.0.xsd and the eas_pdi.xml file in the <INFOARCHIVE_ROOT>/examples/applications/PhoneCalls/data/PhoneCallsSample-2001.zip file

Each <Call> element represents an AIU.

Best Practices for the Schema File

You should use standard XSD features to control XML content, including the following:

- Where possible, use standard XML data types, especially for date and time information.
- Configure the minimum and maximum length of attribute and element values.
- Configure value uniqueness, if acceptable from a performance point of view. (Uniqueness checking is performed during XML validation, at the beginning of ingestion. If the XML is large, uniqueness checking can slow ingestion.)
- In the PDI file, include date and time values that explicitly include the time zone information. For example, in the following text, the bold text specifies one hour offset ahead of UTC time:

```
<CallStartDate>2002-11-18T16:22:04.104+01:00</CallStartDate>
```

- Include the version number in the schema URN, which is defined as a value of the targetNamespace:

```
<xss:schema xmlns:xss="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:eas-samples:en:xsd:office.1.0"
  xmlns:ns1="urn:eas-samples:en:xsd:office.1.0">
  ...
</xss:schema>
```

In the example above:

- *urn* is the prefix
- *eas-samples* can be the name of your company

- *en* is the language (in this case, English)
- *xsd* specified the category or application as well as the version
- Adopt a consistent naming rule for the schema URNs. This will make it easier to remember the URNs, which are referenced in multiple places during the configuration.

Additional SIP Archiving Files for Ant Script Applications

The following files are used for SIP archiving:

Files	Location
Build properties file (<code>build.properties</code>)	The InfoArchive application's directory
Build file (<code>build.xml</code>)	
Ant task files	The InfoArchive application's resources directory
Ingestion file (<code>ingest.xml</code>)	The InfoArchive application's resources /content directory
<code>pdi-crypto.xml</code>	
<code>pdi.xml</code>	
Other supporting Ant files	The <code><INFOARCHIVE_ROOT>/examples</code> directory and the <code><INFOARCHIVE_ROOT>/examples/legacy-ant-applications</code> directory

Base Files

The base files usually do not need to be changed from one InfoArchive application to another. These files include the following:

- The build properties file, which contains variable declarations
- The build file, which defines the targets to execute
- Ant task files, which are XML files that are referenced in the `build.xml` file

When you create your own ZIP file for ingestion, you can copy the base files from a example InfoArchive application. However, you should inspect each base file to ensure that there are no dependent values. You should also update the build properties file, and you might need to update other Ant task files. It should be obvious which values need to be changed. You can change what happens when you run Ant by changing the build file.

Ingestion File

The ingestion file defines the different processors that InfoArchive needs to perform operations on SIPs. You do not need to change this file, and you should not delete it.

The ingestion file essentially contains Java classes that are defined by InfoArchive and allow Ant to perform some functions. All of the processors are defined by InfoArchive, so it is not possible to add your own processor.

For an example of an ingestion file, see the `ingest.xml` file in the `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/PhoneCallsGranular/resources/content` directory.

pdi-crypto.xml File

The `pdi-crypto.xml` file contains the configuration of cryptography processors. If you use cryptography, you need to change the file because it points to specific elements in the PDI.

pdi.xml File

The `pdi.xml` file contains processor configuration information for the following:

- Indexes
- Partition keys that are used during searches
- Counting AIUs
- Content items associated with AIUs

For an example, see the `pdi.xml` file in the `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/PhoneCallsGranular/resources/content` directory.

The following indexes are available:

- Path value index
 - The most common index
 - Indexes values of elements and attributes
 - You can create composite indexes using multiple elements or attributes
 - The `<name>` element is required, which usually refers to the element or attribute being indexed
 - The `<path>` element is also required, which is the path to the AIU and the element or attribute to be indexed
- Full-text index
 - Indexes values of elements and attributes
 - Tokenizes elements into terms
 - Consumers more storage than path value indexes

- Enables wildcard searches
- The <name> and <element.uri> elements are required

Indexes and partitions are important for efficient searching. Partition keys are used to narrow searches by finding only AIPs that contain the AIUs that match the search. Partition keys allow you to search only the AIPs that contain a specific value, and then search the AIUs (records) within. Each AIP can have multiple partition keys to satisfy different search criteria.

In a pdi.xml file, in the pdi.pkeys section, the partition keys are defined as pkey elements. In the example pdi.xml file, the partition keys are called dateTIme01, dateTIme02, and values01.

Partition keys are computed during ingestion using XQueries and stored as attributes on the AIP object (systemdata). For more information, see [Using Partition Keys](#).

InfoArchive uses an XQuery called pdi.aiu.cnt to count AIUs during ingestion and validate the count against the aiu_count value in a SIP descriptor. If the counts match, then the archive is valid.

InfoArchive assigns an XQuery called pdi.aiu.id to each AIU and uses the query to retrieve all AIU nodes. You can change the query in the pdi.aiu.id section of the pdi.xml file to return nodes appropriate for your own AIUs.

SIPs that contain unstructured content must provide queries that InfoArchive can use to access the content. These SIPs must use the ri.init section in the pdi.xml file. This section should query the unstructured content and return <content> elements, which allow InfoArchive to create a table of contents to display the content elements. The <content> elements must have the type attribute, which is the MIME type of the content. The <content> element value should be the AIU value that refers to the name of the unstructured content.

In the example pdi.xml file, a query searches the unstructured content and returns audio/mpeg content.

InfoArchive also requires an XQuery called xdb.pdi.ci.id for any SIPs that contain unstructured content. This query generates content identifiers for the unstructured content. It returns the value of the content name concatenated with the string :ci:, concatenated with a sequence number.

Additional SIP Archiving Files for Declarative Configuration Applications

The following files are used for SIP archiving:

Files	Location
Configuration YAML file (configuration.yaml)	The InfoArchive application's config directory
PDI crypto file (pdicrypto-<APPLICATION_NAME>.PdiCrypto.xml)	The InfoArchive application's data/data-model-config directory
PDI configuration file (pdi-<APPLICATION_NAME>.Pdi.xml)	

The Configuration YAML File

The configuration YAML file (`configuration.yml`) defines the application, including the following:

- Application name, category and description
- Encryption information

You can find an example of the configuration YAML file in the `<INFOARCHIVE_ROOT>/examples/applications/PhoneCalls/config` directory. The `install.iashell` file defines the IAShell commands that are run when you install the application.

Authentication

Declarative configuration applications complete authentication using the settings in the `<INFOARCHIVE_ROOT>/config/iashell/application.properties` file.

PDI Crypto File

The PDI crypto file contains the configuration of cryptography processors. If you use cryptography, you need to change the file because it points to specific elements in the PDI.

PDI Configuration File

The PDI configuration file contains processor configuration information for the following:

- Indexes
- Partition keys that are used during searches
- Counting AIUs
- Content items associated with AIUs

For an example, see the `pdiCrypto-PhoneCallsGranularPdiCrypto.xml` file in the `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications/PhoneCallsGranular/config/data-model-config` directory.

The following indexes are available:

- Path value index
 - The most common index
 - Indexes values of elements and attributes
 - You can create composite indexes using multiple elements or attributes

- The `<name>` element is required, which usually refers to the element or attribute being indexed
- The `<path>` element is also required, which is the path to the AIU and the element or attribute to be indexed
- Full-text index
 - Indexes values of elements and attributes
 - Tokenizes elements into terms
 - Consumes more storage than path value indexes
 - Enables wildcard searches
 - The `<name>` and `<element.uri>` elements are required

Indexes and partitions are important for efficient searching. Partition keys are used to narrow searches by finding only AIPs that contain the AIUs that match the search. Partition keys allow you to search only the AIPs that contain a specific value, and then search the AIUs (records) within. Each AIP can have multiple partition keys to satisfy different search criteria.

In a PDI configuration file, in the `pdi.pkeys` section, the partition keys are defined as `pkey` elements. In the example PDI configuration file, the partition keys are called `dateTime01`, `dateTime02`, and `values01`.

Partition keys are computed during ingestion using XQueries and stored as attributes on the AIP object (`systemdata`). For more information, see [Using Partition Keys](#).

InfoArchive uses an XQuery called `pdi.aiu.cnt` to count AIUs during ingestion and validate the count against the `aiu_count` value in a SIP descriptor. If the counts match, then the archive is valid.

InfoArchive assigns an XQuery called `pdi.aiu.id` to each AIU and uses the query to retrieve all AIU nodes. You can change the query in the `pdi.aiu.id` section of the PDI configuration file to return nodes appropriate for your own AIUs.

SIPs that contain unstructured content must provide queries that InfoArchive can use to access the content. These SIPs must use the `ri.init` section in the `pdi.xml` file. This section should query the unstructured content and return `<content>` elements, which allow InfoArchive to create a table of contents to display the content elements. The `<content>` elements must have the `type` attribute, which is the MIME type of the content. The `<content>` element value should be the AIU value that refers to the name of the unstructured content.

In the example PDI configuration file, a query searches the unstructured content and returns `audio/mpeg` content.

InfoArchive also requires an XQuery called `xdb.pdi.ci.id` for any SIPs that contain unstructured content. This query generates content identifiers for the unstructured content. It returns the value of the content name concatenated with the string `:ci:`, concatenated with a sequence number.

Installation and Ingestion

After you have set up the configuration files, you can perform an installation. During installation, most applications for SIP archiving create the following:

- A database
- A space

- Crypto objects
- Stores
- Other objects

After an application creates all the required configuration objects, it can perform an initial ingestion. You can ingest more data later.

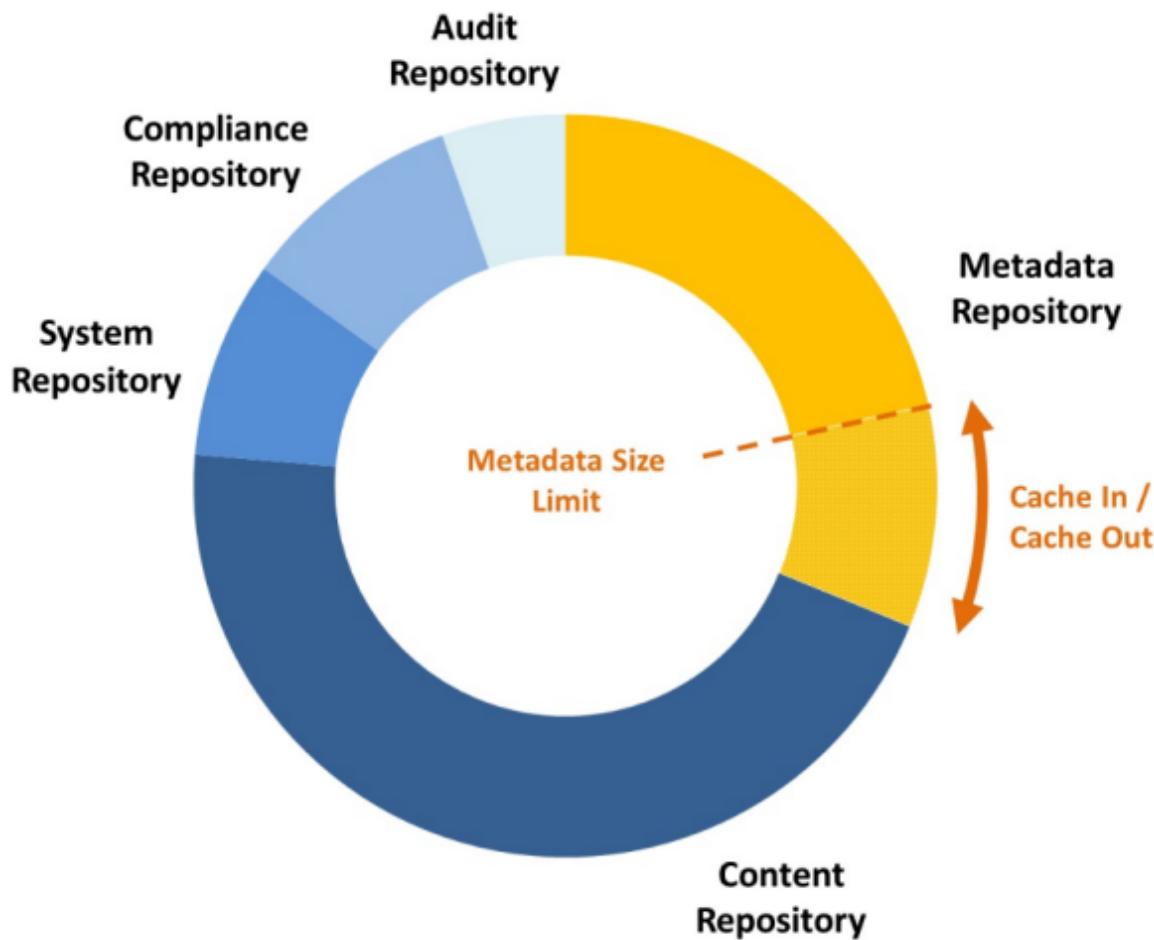
You should also consider restricting access to the application, in case it contains sensitive data. For more information about permissions, see [Managing User Accounts and Permissions](#).

Cache-In and Cache-Out

Cache-in and cache-out provides an ability for SIP archiving applications to improve performance by reducing the number of libraries in xDB.

InfoArchive stores data from SIP-based applications in several xDB repositories:

- The *audit repository* contains audits generated by the system until they are archived
- The *compliance repository* contains retention and hold information
- The *system repository* contains metadata for system objects (for example, configuration objects and AIPs)
- The *content repository* contains the unstructured contents of archived AIPs
- The *metadata repository* contains all of the metadata (in particular, PDI files) of archived AIPs



Cache-in and cache-out functionality allows you to manage disk space for the metadata repository. You can choose to limit the data in the cache (the metadata repository) for a particular application. If this limit is exceeded, the CacheOut job removes the least frequently accessed AIPs from the metadata repository until the size of the cache satisfies the limit that you set.

For a user to perform a synchronous search for an application, the AIPs targeted by the search must be present in the metadata repository. If an AIP targeted by a synchronous search is not present in the metadata repository, InfoArchive asks the user to do a background search instead. During the background search, the system finds the targeted AIP and caches it back in.

From a technical perspective, the cache-in and cache-out functionality automatically reduces the number of xDB segments (internal xDB files that contain data). To perform a synchronous search, all metadata must be present into the metadata repository. The xDB segments must be stored in a file hierarchy on one of the following:

- Storage Area Network (SAN), which is fast but expensive
- Network Attached Storage (NAS), which is slower but less expensive

The cache-in and cache-out mechanisms are based on the xDB library's individual backup. The xDB server performs this backup during the ingestion process and saves the backup file in the content repository (for example, Centera, ECS, or S3).

When InfoArchive needs to cache-in and restore the segment, the xDB server retrieves the backup file from the content store and stores it in xDB.

The CacheOut job determines which xDB libraries should be removed from the metadata repository, based on usage statistics as follows:

- The usage statistics are stored in the system repository.
- For each xDB library in the metadata repository, the last access date is stored. The access date is updated when the library is created or restored, as well as each time a corresponding AIP is the target of a search.
- When an xDB library is removed from the cache, it is also removed from the usage statistics. For the compliance metrics, the records are still part of the calculations even if the library was cached-out.
- When the CacheOut job needs a candidate for removal, it takes the one with the oldest access date.

SIP Ingestion Modes

InfoArchive supports three ingestion modes: private, pooled, and aggregated. Using the proper ingestion mode is key for optimizing search performance. In general, your choice of ingestion mode depends on the SIP package characteristics.

- **Private mode:** If you might want to store a huge number of AIUs in the SIP package (for example, 100,000 AIUs), then you should use private mode. In this mode, indexes are created during ingestion.
- **Aggregated mode:** If the SIPs contain few AIUs (for example, 2 to 100), and if many SIPs must be ingested, then aggregated mode works better. The ingestion will be quick and the AIPs will be searchable quickly. The AIPs could be aggregated by the Close Job to reduce their number. The structured data of the AIPs will be stored in a pooled library. The indexes are created only after the Close job has been executed.
- **Pooled mode:** This mode is similar to aggregated mode, but the AIPs cannot be aggregated by the Close job. This mode is appropriate if SIPs contain a medium number of AIUs.

	Private	Pooled	Aggregated
Description	Each SIP package has its own dedicated xDB library	All SIP packages share the same xDB library	All SIP packages share the same xDB library before being combined into one package at the library's closing
Use case	Standard archiving < 50 packages a day	SAP archiving < 1000 packages a day	Healthcare archiving 1000+ small packages a day
Retention	Applied to each package	Applied to each package	Applied to aggregated package

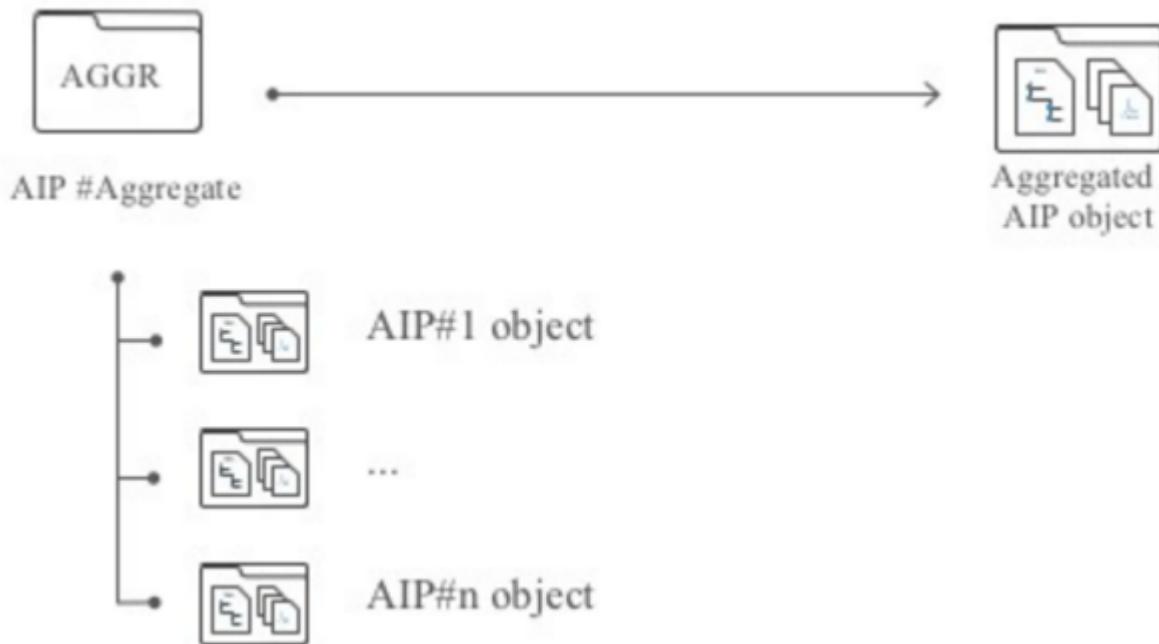
	Private	Pooled	Aggregated
Backup	Immediate	At library close	At library close
Search	Ideal when partition keys are used to reduce the search scope (200 AIPs after filtering)	Ideal when partition keys do not help to reduce the search scope (10,000 AIPs / 200 xDB libraries after filtering)	Ideal when partition keys are used to reduce the search scope (200 AIPs after filtering)

Unitary Archiving and Aggregation

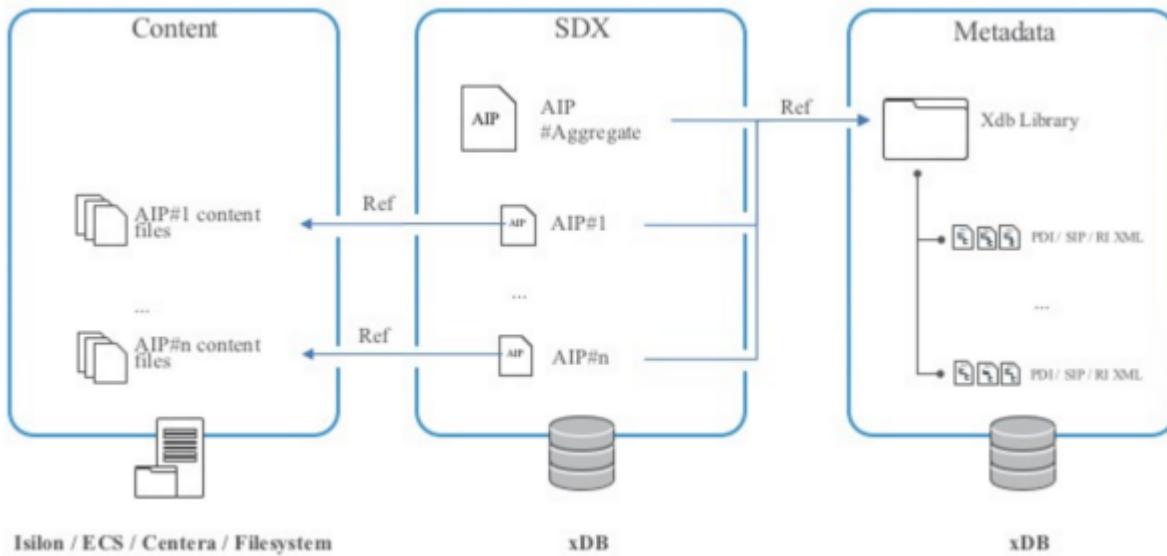
Unitary archiving allows a client application to archive data synchronously in InfoArchive. If the web service call succeeds, archived data can be immediately searched. Unitary archiving executes the same processing as batch ingestion.

Unitary archiving can be completed by an aggregation step to optimize storage and reduce the repository footprint.

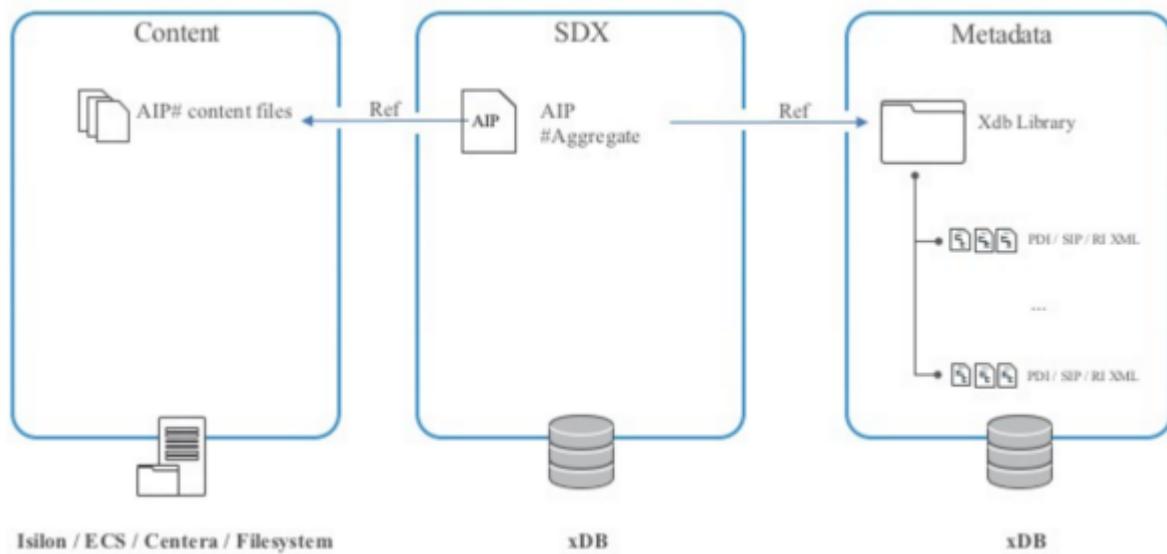
The following figure illustrates unitary archiving before aggregation:



The following figure illustrates unitary archiving during aggregation:



The following diagram illustrates unitary archiving after aggregation:



Operational Concepts

Authentication Mechanisms

InfoArchive supports the following authentication mechanisms:

- OpenText Directory Services (OTDS)
- Active Directory (AD)
- Lightweight Directory Access Protocol (LDAP)
- Example user accounts (also known as in-memory user accounts)

Authentication takes place when a user logs into IAWA. Authentication is based on the OAuth 2.0 framework and uses JSON web tokens.

For LDAP/AD-based authentication, InfoArchive basically configures the Spring Security LDAP implementation. The actual authentication is implemented by the Spring Security LDAP library. Refer to the following for the details of the configuration parameters:

- <https://docs.spring.io/spring-security/site/docs/4.2.3.RELEASE/reference/htmlsingle/#nsa-ldap-authentication-provider-attributes>
- <https://docs.spring.io/spring-security/site/docs/4.2.3.RELEASE/reference/htmlsingle/#ldap-authentication>

The logging for Spring Security LDAP can be configured using the following key in the <INFOARCHIVE_ROOT>/config/iawebapp/application.yml file:

```
logging:  
  level:  
    org.springframework.security.ldap: 'DEBUG'
```

To enable embedded Tomcat access logs when running in standalone mode, ensure the <INFOARCHIVE_ROOT>/config/iawebapp/application.yml file includes the following:

```
server:  
  host: ${infoarchive.gateway.host}  
  port: ${infoarchive.gateway.port}  
  tomcat:  
    accesslog:  
      directory: "c:/tomcat/accesslog"  
      enabled: true
```

In addition, the person configuring LDAP/AD must be well familiar with or have access to the personnel familiar with the specific structure of the directory in LDAP/AD server, and understand how the groups and users are located and queried.

In a production configuration, you should use OTDS, AD, or LDAP to authenticate user accounts. The example user accounts are intended for use with a demo configuration, when you want to quickly set up InfoArchive so that you can test its features, set up a proof of concept, give a short demonstration, or set up a basic development environment. For more information about example user accounts and demo configurations, see the *InfoArchive Installation Guide*.



Caution: Do not use the example user accounts in a production environment. Attackers can use one of the example user accounts to gain unauthorized access to your InfoArchive system and the assets that it contains. These out-of-the-box accounts are meant for demo purposes, and the default password for each account is *password*.

Passwords must be at least five characters in length, and there is no limit on the maximum length.

Authorization

After user accounts are authenticated, InfoArchive authorizes the user accounts to use one or more InfoArchive resources. Users, groups, roles, and actions are mapped as follows:

- *Users* are mapped to *groups* in Active Directory, LDAP, or in the file that manages example user accounts (<INFOARCHIVE_ROOT>/config/iawebapp/application-infoarchive.gateway.profile.AUTHENTICATION_IN_MEMORY.properties)
- *Groups* are mapped to *roles* in the <INFOARCHIVE_ROOT>/config/iaserver/application.yml file
- *Roles* are mapped to *actions* in the <INFOARCHIVE_ROOT>/config/iaserver/application.yml file

You should map your existing groups in Active Directory or LDAP to roles in InfoArchive. When you map a group to a role, the role is assigned across all InfoArchive applications. However, you can control which applications are visible to a group. By default, all applications are visible.

User Roles

InfoArchive defines specific user roles that are usually involved in the application archiving process. The InfoArchive user interface is built and pre-configured for these fixed roles.

The following table illustrates the actions available for each role:

Action	Administrator	Auditor	Business Owner	Developer	End User	Retention Manager	E-Discovery Administrator	IT Owner
Run searches, including background		✓	✓	✓	✓	✓	✓	
Create search				✓				
Compliance operations						✓		
Run jobs	✓			✓				
Ingest				✓				

Action	Administrator	Auditor	Business Owner	Developer	End User	Retention Manager	E-Discovery Administrator	IT Owner
Configure system	✓							
View storage dashboard	✓		✓					✓
Configure audits	✓							
Search audits		✓	✓	✓	✓	✓	✓	
Matters and collections							✓	

Actions

A user's role, which is inherited by membership in a group, determines the actions that the user can perform. Users can access the functionality on various tabs depending on the roles they are a member of, via the group membership. For example, an Administrator can see an **Administration** tab, and a Retention Manager can see a **Compliance** tab.

Administrators can see which actions are available for a group in the **Groups** tab.

Auditing

InfoArchive allows the auditing of many events. There are events related to the following:

- System
- Tenant
- Individual applications

InfoArchive stores audited events in the xDB database. In IAWA, Administrators can access the **Administration > Audit** tab to select which events to audit.

Audits can be searched after you install the Audit application and run the Archive Audits job. The job ingests events from the xDB database into the database associated with the Audit application. The Audit application applies a retention policy to the events that it ingests, which allows audited information to be purged.

For more information about installing the audit application, see the *InfoArchive Installation Guide*. For more information about working with audits in IAWA, see [Configuring Audit Events Using the Audit Tab](#).

Product Overview

Chapter 2

Configuration

Creating an Application Using the Web Application

An application is a logical configuration object in an archive system that presents a customer business item for preserving and storing data. An application can be one of the following types: SIP or table.

Applications in InfoArchive provide access to all archived data. Each application represents a single decommissioned or active archive. Typically, most users will access applications via the InfoArchive web application (IA Web App). It is also possible, however, to access applications via the REST API, which allows customers to create a custom interface to interact with the applications.

There is an **Applications** page in IA Web App that is available right after logging in. It includes a list of the available applications. From here, a user can create new or edit existing applications, or a user can select an application in order to execute a record search. Note, that the create and edit operations are available for the Developer user role only.

A user can quickly scan for a specific application by using the **Find an Application** filter located in the navigation panel of **Applications** page. The filter functionality matches the names and descriptions of each application to input text in the filter field and returns the suitable results.

Prior to creating a search form or any back-end artifacts, the Developer needs to create an application. An unfinished application can be saved and completed at a later time.

To create a new application, complete the following steps:

1. In the **Applications** tab, click + > **Create Application**.
2. Enter the following information:

Field	Description
Application Name	A name that identifies the application.
Description	A description of the application.
Category	A category or select a previously used category.

Field	Description
Default Retention Policy	<p>The default retention policy for the application. The selected retention policy will be applied to ingested SIP if the holding is configured for package based retention and no default retention class was specified on the holding nor the AIP defined its own retention class. See the holding wizard for options on how to configure retention on ingestion.</p> <p>The Default Retention Policy is only used if the default retention class is not specified on the holding and the SIP does not specify a retention class.</p> <p>Do not specify a default retention policy if retention will be directly applied to records. See the holding wizard configuration for options on how to apply retention at ingestion for SIP applications.</p>
Application Type	<p>An option that specifies a type of archive. It can be one of the following:</p> <ul style="list-style-type: none"> • Application Decommissioning • Active Archiving

Field	Description
Archive Type	<p>The available options depends on the selected Application Type.</p> <p>If the Application Type is 'Application Decommission', then one of the following options is available:</p> <ul style="list-style-type: none"> • 'Based on Table Schema' for a table archive. • 'Based on packages' for a SIP archive. <p>If the Application Type is 'Active Archiving', select Based on packages.</p>
Cache size	<p>This field is only available if the Application Type field is set to Active Archiving.</p> <p>The cache-out/cache-in feature allows the management of the metadata repository size. From a technical perspective, the cache-out/cache-in functionality manages the metadata repository disk space by automatically reducing the number of segments attached to xDB. Refer to Cache-Out/Cache-in Concepts for further information.</p> <p>Select one of the following options:</p> <ul style="list-style-type: none"> • All data is in cache: All AIPs archived in the application will not be cached. • Limit data in cache. Search may take longer for data not in cache: If selected, enter a cache size and indicate whether the limit is in megabytes, kilobytes, etc. <p>If this limit is exceeded, the Cache-Out job removes the less-used AIPs from the metadata repository (cache-out), thereby decreasing its size.</p>

3. Click **Create**.

The application is now listed on the **Application** tab.

When an application is created, it has a Status of 'In Test', which allows a customer to test the application, typically, with fake data.

When creating a table application, the user must specify the XML files for uploading, even if they are empty. Refer to [Creating a Space Using the Web Application](#) for further information.

Editing an Application

A user with role of Developer or Administrator have the ability to edit an application:

1. From the **Applications** page, click the context menu for the application being edited and select **Edit Application**.
2. Edit the application information. For more information about the fields available, see [Creating an Application](#).

The following fields will be disabled if the application has ingested data or includes a defined search configuration:

- Application Type
- Archive Type

If required, edit the **Status** of the application. The Developer can only delete all of the data from an application if the application's Status is 'In Test'. If the application's Status is 'Active', data can only be disposed through the disposition process.

3. If required, update the **Cache Size** field. This field is referring to the xDB cache that the xDB libraries are consuming. The main reason for setting this cache is to control the size of the xDB database for the AIPs. Use this mechanism to organize data so that older historical data will likely not be searched.

When the xDB library is cached out, it still uses the disk space, but will be more expensive to search against.

- a. **All data is in the cache**
- b. **Limit data in cache. Search may take longer for data not in cache:**

4. Click Save.

Deleting an Application

Only a user with the Administrator role can delete an archived application in "In Test" status. The delete operation typically occurs in the development environment, especially during the ETL and data validation processes.

The Administrator must delete data first and then delete the application.

1. Remove all retention and holds from the application. If retention has been defined on the holding, the packages will have had retention applied upon ingestion.
2. The Administrator cannot delete an application that has a job running against it. Suspend the schedule of any job scoped to the application.
Any job instances and order items related to the application will be deleted.
3. [Delete data from the application](#).
4. From the **Applications** page, click the context menu for the application being deleted and select **Delete Application**.
5. When prompted to confirm that you want to delete the application, click **Delete**.

The application is no longer listed on the **Application** tab.

Note: If you delete an application, re-ingest the data, but have difficulty running a search, refer to the [Compliance Troubleshooting](#) to learn how to resolve the issue.

Deleting Data from an Application

This operation requires that the Retention Manager remove all holds and retention policies applied to any items in the application (or applied directly on the application). If the user deletes data, searches executed against the application will no longer work, as schemas will be deleted upon completion of this procedure. If you have created any rules, they are also deleted.

To delete an entire archived application, the application:

- Should not have any packages, tables or records under retention or hold..
- Should not contain any hold sets.
- Must be in “In Test” status
- There can be no indexing in the application.

The user with Developer role can only delete all of the data from an application if the application's status is 'In Test'. If the application's status is 'Active', data can only be disposed through the disposition process. Any retention or hold policies need to be removed from the data before deleting it from the application.

Any time the data of a table application is deleted, the application's searches may not work unless the data is re-ingested. Re-ingesting the data recreates the database, schema and tables.

1. On the **Applications** page, click the context menu for the desired applicationand select **Delete Data**.

It is possible that an application may contain significant data and, consequently, the delete data process runs as a background task.

2. When prompted to confirm that you want to delete all of the ingested data for application, click **Delete**.

Creating Federations, Databases and Storage Systems Using the Web Application

The **Administration > Storage** tab allows the user with Administrator role to create, edit and delete:

- [Federations](#), to act as a container for databases;
- [Databases](#), to hold archived records; and
- [Storage systems](#), to hold unstructured content belonging to records.

Registering a Federation

An xDB federation is a storage configuration resource that contains set of properties to establish a connection with physical xDB federation. A federation is a container for xDB databases.

Existing federations are displayed in a table that contains the following information:

Column	Description
Federation Name	Indicates the name of the federation.
Bootstrap	Indicates the Xhive connection string.
In Use	Indicates whether the federation is currently in use by an xDB database. If a federation is not in use, then user is able to edit or delete it. If a federation is in use, then user is able only to edit it.

The number of federations listed in the table depends on what mode InfoArchive is running in. InfoArchive by default registers a federation called mainFederation which stores information about the deployment.

The user can put everything into one federation or can choose to store retention information in a separate federation. One of configuration options is to store the retention in a separate federation. This is configured via the application.yml file. If InfoArchive is configured to use more than one federation, an xDB server must be running for each additional federation.

Register a federation to segregate the xDB data into separate servers. The user is responsible for creating those data nodes and creating the database. The following configuration simply registers it.

1. On the **Storage** tab, click the + button.
2. Enter the following information:

Field	Description
Federation Name	A unique name for the federation.
Superuser Password	A password the user will have to enter to configure the federation.
Connection URL	A URL of the federation, which specifies the federation bootstrap.

3. Click Register.

The federation now appears in the table on the **Storage** tab.

Further configuration is still required, however. Refer to [Configuring xDB Federations and xDB Databases](#) for further information.

Linking a Database

An xDB database is a storage configuration resource that represents a database in xDB. It contains a set of properties to access the physical database.

Existing databases are displayed in a table that contains the following information:

Column	Description
Database Name	Indicates the name of the database.
Federation Name	Indicates the name of the parent's federation.
Bootstrap	Indicates the X-Hive connection string.
Applications	The list of applications that use the database.
In Use	Indicates whether the database is currently being used by a space as a reference object SpaceRootXdbLibrary. If a database is not in use, user is able to edit or delete it. If a database is in use, user is able only to edit it.

The following procedure creates an xDB database, which should not be confused with the database being used for table-based applications.

1. On the **Storage** tab, click + and select **Create Database**.
2. Enter the following information:

Field	Description
Database Name	A unique name for the database.
Admin Password	A password the Administrator will have to enter to configure the database.
Federation	A parent federation.

3. Click **Create**.

Further configuration is still required, however. Refer to [Configuring xDB Federations and xDB Databases](#) for further information.

Configuring xDB Federations and xDB Databases

Every xDB federation and xDB database that is not defined in the server's application.yml file must be registered. InfoArchive can connect to a federation, but it cannot start it up.

If using different databases for applications, the databases must be registered.

This can be done using a REST call, IA Shell or IA Web App.

Always define the datanode (federation) and the name of the repository. To connect to the datanode, always specify a bootstrap.

In the case of retention running in a separate datanode, the port may be different. It depends on where the xDB Server for retention resides.

The user can choose a different data node than the system's value, but it is the customer's responsibility to ensure that the xDB server is running.

There is an example in the PhoneCalls application in which a new federation is registered.

Structured data in InfoArchive, including AIPs/AIUs and table data, is stored in xDB databases.

There are also two important repositories:

- A potentially large repository where PolicyApplications and HoldApplications are stored. It is recommended that you keep this in a federation separate from the one under "system".

```
managedItemData:  
  xdb:  
    dataNode:  
      name: retentionFederation  
      # The following value '2910' for the port number must be the actual port  
      # number where the xDB server is located. Adjust it accordingly.  
      bootstrap: xhive://<IP_OF_xDB>:2911  
      superuser:  
        password: <Password>  
    database:  
      name: managedItemDatabase  
      admin:  
        password: <Password>
```

Adding a Storage System Using the Web Application

Storage refers to a storage configuration object that contains a list of properties for target storage configuration. Storage can be one of the following types: Local File System, Isilon, ECS, S3, Centera and Custom Storage. A storage system holds data, such as unstructured content for records, library backups, raw XML files, ingestion logs, etc.

Existing storage systems are displayed in a table that contains the following information:

Column	Description
Storage Name	Indicates the name of the storage system.
Storage Type	Indicates the type of storage. Accepted storage types include: <ul style="list-style-type: none">• Isilon• Local File System• ECS• Amazon S3• Custom Storage• Centera

Column	Description
Properties/Storage Details	The information displayed in these columns depends on the type of storage system being used.
In Use	<p>Indicates whether the storage system is currently being used.</p> <p>If a storage system is not in use, user is able to edit or delete it.</p> <p>If a storage system is in use, user is able only to edit it.</p> <p>All actions are available in context menu for the storage system.</p>

To add a storage system using IA Web App:

1. In the IA Web App, on the **Storage** tab, click **+**.
2. Select a **Storage Type**:

Item Selected	Actions
File Storage	
Isilon	<ul style="list-style-type: none"> a. Enter the Storage Name. b. Enter a Description. c. Enter the Folder Path.
Local File System	<ul style="list-style-type: none"> a. Enter the Storage Name. b. Enter a Description. c. Enter the Folder Path.
Object Storage	
ECS	<ul style="list-style-type: none"> a. Enter the Storage Name. b. Enter a Description. c. Enter the URL of the object storage being added. d. Enter the Credential Name. e. Enter an Credential Description. f. Enter an Access Key ID. g. The ECS Management REST API provides the ability to allow authenticated domain users to request a secret key to enable them to access the object store. Enter the Secret Key. <p>Click '+' to add another credentials object. Click 'x' to delete an access pair.</p>

Item Selected	Actions
S3	<p>a. Enter the Storage Name.</p> <p>b. Enter a Description.</p> <p>c. Enter the URL of the object storage being added.</p> <p>d. Check whether to Enable Glacier.</p> <p>e. Check whether to enable a proxy</p> <p>f. Enter the Credential Name.</p> <p>g. Enter the Credential Description.</p> <p>h. Enter the Access Key.</p> <p>i. Enter the Secret Key.</p> <p>Click '+' to add additional variables. Click 'x' to delete an access pair.</p>
Custom Storage	
Custom Storage	<p>Enter the following information. The values of these fields can be set according to your preference:</p> <p>a. Enter the Storage Name.</p> <p>b. Enter a Description.</p> <p>c. Enter the Factory Service Name. This value is used in the implementation of your content store API.</p> <p>Provide a bag of properties to use in your implementation:</p> <ul style="list-style-type: none"> • Enter the Name. • Enter the Value. <p>Click '+' to add additional variables. Click 'x' to delete an access pair.</p> <p>Once complete, you are able to create a custom space and store.</p>

Item Selected	Actions
Legacy Object Storage	
Centera	<p>a. Enter the Storage Name.</p> <p>b. Enter a Description.</p> <p>c. Enter the Connection String.</p> <p>d. Enter the following Pool Entry Authorization (PEA) information:</p> <ul style="list-style-type: none"> • Enter the Variable. • Enter the Content. <p>Click '+' to add additional variables. Click 'x' to delete an access pair.</p>

3. Click the **Test Connection** button to ensure the connection works. User can only use the Test Connection button with ECS, Amazon S3 and Centera storage systems. The button does not appear for Isilon, local file or custom storage systems.

Once the **Test Connection** button is pressed, InfoArchive tries to establish a connection with the storage system. User is notified if the procedure is successful or not.

If the connection is not successfully established, the error message indicates the reason why the connection failed. Make the necessary changes to the fields indicated in the error message and click the **Test Connection** button again.

Note: It is still possible to create the storage system even if the Test Connection process fails. This may occur if the user with an Administrator role wants to create the configuration objects first, and then make the storage system available.

4. Click **Create**.

Configuring Storage Systems

Proceed with the steps in one of the following sections, depending on the type of storage being used:

- [Configuring ECS Storage](#)
- [Configuring Centera Storage](#)
- [Configuring Custom Storage](#)

Existing storage systems are displayed in a table that contains the following information:

Storage Systems			
Storage Name ▲	Storage Type ▲	Properties	Storage Details
defaultFileSystemRoot	Local File System (File Storage)	Description: Folder Path:	Default FileSystemRoot data/root
testCustomStorage	Custom Storage (Custom Storage)	Description: Factory Service Name:	testCustomStorageDescr testCustomStorageFactory

Displaying 1 - 2 of 2

Configuring ECS Storage

This section illustrates how to configure system objects required to ingest data into Elastic Cloud Storage (ECS).

To configure an ECS object for ingestion, you must:

1. [Add a storage system](#) with an ECS storage type.
 - a. Specify the **URL**, **Access Key** and **Secret Key** to connect to the ECS instance.
2. [Create an application](#).
3. [Create a space](#) under the newly created application.
 - a. Specify Object Storage in the **Storage System** field.
 - b. Select the URL created in step 1a.
4. [Add a store](#).
 - a. Create a bucket to store the data in.

After following these steps, you will be able to ingest data into ECS.

ECS Data Model for InfoArchive

The section describes the data model of Elastic Cloud Storage (ECS) in InfoArchive.

System data types specific to ingestion of unstructured content in ECS are as follows:

- StorageEndPoint: It contains the attribute URL used by the IA Server to connect to an ECS server node. Another attribute of StorageEndPoint is “type”, and the value must be “ECS” if configuring an ECS storage. Its REST relation <http://identifiers.emc.com/storage-end-points> is available under the Home resource of the IA Server (<http://identifiers.emc.com/services>), which is used to create and retrieve StorageEndPoint objects.
- StorageEndPointCredential: Used to persist an ECS accesskey and a secretkey in SDX. A reference of a StorageEndPointCredential object (once it is created) is specified by the client in the payload for POSTing a new SpaceRootObject. The ECS access key and secret key are generated by a third party and will be preexisting at the time of creating a new StorageEndPointCredential object. Its REST relation <http://identifiers.emc.com/storage-end-point-credentials> is available under StorageEndPoint resource.
- SpaceRootObject: Has a reference to a Space and a StorageEndPointCredential. Its REST relation <http://identifiers.emc.com/space-root-objects> is available under a Space resource, which is used to create and retrieve SpaceRootObject objects.
- Bucket: Contains a reference to SpaceRootObject. Its REST relation <http://identifiers.emc.com/buckets> is available under a SpaceRootObject resource, which is used to create and retrieve Bucket objects.

REST APIs are available to manipulate the system data objects to perform end-to-end operation of content ingestion. Data to be ingested is represented by the Content object. It has a reference to Store, which has one-to-one mapping with Bucket. In other words, Store has a reference to Bucket. The Store has a type ECS, which enables InfoArchive to ingest content into ECS. For successful ingestion,

the following SDX objects are required to be created: StorageEndPoint, StorageEndPointCredential, Tenant, Application, Space, SpaceRootObject, Bucket and Store.

Configuring Centera Storage

This section illustrates how to configure system objects required to ingest data into Centera storage.

To configure a Centera object for ingestion, you must:

1. [Add a storage system](#) with a Centera storage type.
 - a. Specify the **Connection String**, which is the Centera IP.
 - b. Enter the following Pool Entry Authorization (PEA) information:
 - Enter the Variable.
 - Enter the Content.
2. [Create a space](#) under the newly created application.
 - a. Select Legacy Object Storage in the **Additional System Storage** field.
 - b. Select the connection string created in step 1a.
3. [Add a store](#).
 - a. Select Centera as the value for the **Space Root** field.
 - b. Create a bucket to store the data in.
4. Create an application or edit an existing application to use the newly created Centera store. For a SIP application, it is necessary to assign the stores (Centera, ECS, Isilon, File) at the holding level.

Be sure to process the post-installation steps to include Centera SDK libraries.

Configuring AWS S3 with Amazon Glacier

In order to archive data, you must complete the following instructions to configure AWS S3 and AWS Glacier. The following should be configured prior to starting this procedure:

- Federation
- Database
- Application
- Space

The general rule of thumb is that any content that the system needs to access regularly should not be stored in Amazon Glacier.

1. Configure a storage system that can be done in the Storage section of the IA Web App.
 - a. For the **Storage Type** field, select S3.
 - b. Provide values for the remaining fields. Refer to [Adding a Storage System Using the Web Application](#) for more information.

When entering the **URL**, ensure it points to the end point of the desired AWS S3 service.

When entering the **Secret Key**, enter the AWS S3 account

2. If desired, configure the Glacier feature by selecting “Enable Glacier” so that data can be archived in AWS Glacier. The archival process is driven by the rules as specified while configuring a store. The following rules apply to all of the content objects in a bucket:

- **S3 to Glacier Transition Rule:** Signifies the time period (in days) that starts from the creation date of the content objects of a bucket, which is also configured in the Stores section. At the end of this period, those content objects will be archived in AWS Glacier.

Note: It may take a while before the archived process is completed.

- **Glacier to S3 Restoration Rule:** Decides how long the restoration of an archived object may take so that it can be read. Users can specify one of the three options:
 - Expedited (fastest: 1-5 minutes for retrieval),
 - Standard (3-5 hours for retrieval) and
 - Bulk (5-12 hours for retrieval).
- **Rule Name:** Provide a unique name for configuration rule.
- **S3 Duration:** Signifies the number of days until a restored object will be kept in AWS S3.

If AWS S3 (with Glacier feature) is accessed via a proxy server, “Enable Proxy” should also be checked besides “Enable Glacier”. The relevant details of the proxy (for instance, Proxy URL, Proxy User Name and Proxy User Password are subsequently specified). Proxy User Name and Proxy User Password are optional and only specified if required by the proxy server being used.

When a content is stored on Glacier, it is not immediately available. If an end user attempts to access content, she or he is notified in the Status column of the **Background Requests** tab.

Using Multi-Part Capabilities to Improve SIP Upload Performance

Users can upload SIP and table data in AWS S3 using a multi-part upload strategy, in which a file is uploaded in multiple parts. Two properties specific to that strategy are: part size and multi-part upload threshold, which are provided by users while configuring a `StorageEndPoint` object.

If the file size exceeds the specified threshold, the multi-part upload strategy will be used. AWS S3 splits the file into chunks, each having the size specified in the `StorageEndPoint`'s `partSize` property.

If any of the aforementioned two properties are unspecified (`part_size` or `multi-part_upload_threshold`), IA Server uses a suitable default value for the missing field. And, if the specified value is invalid, then an error is issued and the creation of a `StorageEndPoint` will be unsuccessful.

Configuring Custom Storage

This section illustrates the process of:

- Configuring custom storage
- Implementing an InfoArchive content store
- Setting up the environment

1. Configure the custom store.

To configure a custom store, a bag of properties and the name of the Spring service, which is used by InfoArchive to find the customer implementation of ContentStoreFactory, can be specified by the clients via the REST API. It can be done by following the REST relation "<http://identifiers.emc.com/custom-storages>", which is available under the Services link ("<http://identifiers.emc.com/services>"). The properties can be used by the clients in implementing the custom store.

2. Customize the implementation. The following three interfaces can be used by customers to define the implementation:

- **com.emc.ia.content.store.ContentStore**: Use this interface to define methods that perform CRUD operations on unstructured content.

The implementation should also set the location of an object through Content#setPath(String path), where the path contains the location of the object. If the aforementioned method is not used in combination with Content#getPath(), the implementation can never change the algorithm for path construction, since it needs to locate previously stored content. If the implementation uses Content#setPath() at the end of a write, then on a read, it can use Content#getPath() to retrieve the stored path to locate the object to read. If not, the implementation will have to construct the path on each read using the other Content fields.

Note: The getPath()/setPath() feature was not available in the 4.2 Custom ContentStore API. Hence, for any read of Content objects (in a Custom ContentStore) created with IA 4.2, Content#getPath() will always be blank. The implementation will need to take that into account and fall back to resolving against its initial path construction scheme for such Content objects. InfoArchive cannot migrate this for you, since it has no knowledge about the path construction scheme of your Custom ContentStore implementation.

- **com.emc.ia.content.store.ContentStoreFactory**: It is implemented as a Spring service. A service name, custom storage name and properties (optional) are provided to InfoArchive while configuring a custom store. The following is an example of what the payload of adding a custom store looks like:

```
{
  "name" : "customStorageName",
  "factoryServiceName" : "CustomContentStoreFactoryImpl",
  "properties": {
    "key1" : "foo",
    "key2" : "bar"
  }
}
```

The following is an example of defining ContentStoreFactory as a service:

```
@Service("CustomContentStoreFactoryImpl")
public class ContentStoreFactoryImpl implements ContentStoreFactory {
  @Override
  public ContentStore newStore(Bucket bucket, Map<String,
  String> properties) {
    // return instance of your implementation of ContentStore.
  }
}
```

In the above example, the service annotation contains "CustomContentStoreFactoryImpl". It should be identical to what is specified during custom storage configuration (see the custom store's payload example).

- **com.emc.ia.content.store.ContentStoreRetention:** It can be implemented if the customer wants to push retention to the hardware level. The following two interfaces can be used to get information regarding the content store that might be useful in implementing custom store:
 - **com.emc.ia.content.store.Bucket:** It returns the name of the bucket as specified during configuration of content store (see step 1).
 - **com.emc.ia.content.store.Content:** It returns information that can be used by customers in implementing the custom store.
3. Set up the environment.

Customers have to define a Spring configuration class with a condition: Its package name must be prefixed with "com.emc.ia". This enables InfoArchive to find customer-defined Spring services that also includes one for the ContentStore. For example:

```
package com.emc.ia.my.content;
@Configuration
@ComponentScan(basePackages = { "com.my.content.store" })
public class CustomStoreConfig { }
```

The above configuration can be used by customers to include packages that contain implementation classes. In addition to the configuration class, InfoArchive requires you to add implementation (as a jar) on its class-path. To achieve that, the jar will be copied into the external directory of the InfoArchive distribution (dir: /infoarchive/lib/external/).

Creating a Space Using the Web Application

Space is a configuration object that ties the storage system to a particular application. It can have one or more storage systems, but only one database library.

The **Spaces** tab allows the Administrator to create a space, which holds records and content for an application. Before you create a space, complete the following:

- [Create an application](#)
- [Register a federation](#)
- [Link a database](#)
- [Add a storage system](#)

Spaces are displayed in a table that contains the following information:

Column	Description
Space Name	Indicates the name of the space.
Database Libraries	Indicates the database library associated with the space.
Storage System	Indicates the storage system associated with the space.

Column	Description
In Use	<p>Indicates whether the space is currently being used by at least one of the following objects:</p> <ul style="list-style-type: none"> • SpaceRootFolder • SpaceRootObject • SpaceRootXdbLibrary <p>If a space is not in use, user is able to edit or delete it.</p> <p>Along with updating the Space Name, user is able to delete a database library and/or storage system. To add a new database library and/or storage system, click +.</p> <p>If a Space is in use, user is able to only edit it</p>
Side panel	<p>Indicates the details of the selected space, including:</p> <ul style="list-style-type: none"> • Database library of the space • The storage system for the space • The file system root path for the space

1. In the IA Web App, on the **Spaces** tab, click +.
2. Enter the following information:

Field	Description
Application	Select the application that will be associated with the space. The cache size is displayed if the application is a table archive type.
Space Name	Enter a name for the space.
Structured Data Database	Select the database that will be associated with the space and used for preserving of structured data.
Structured Data Database Library	Enter a name for the database library used for structured data preservation.
Search Results	The files allow user to configure a dedicated database library that is used for search result preservation. It is highly recommended to use a dedicated library but, at the same time, decide whether or not to use the same library for structured data preservation.
Search Result Database	Select the database that will be associated with the space and used for preserving of search results.

Field	Description
Search Result Database Library	Enter a name for the database library used for search results preservation.
Storage System	Complete the following steps to assign storage to a space: a. Select the storage system type for the space (for example, file storage). b. Select the desired storage for the space. c. Click CREATE .

3. Click **CREATE**.

The space now appears in the table on the **Spaces** tab.

Adding a Store Using the Web Application

A store is a storage configuration object that contains properties for linking a space with a File System Folder or Bucket. Stores hold records in the context of an application.

The **Stores** tab allows the Administrator to add and configure a storage area for different binary content.

Existing stores are displayed in a table that contains the following information:

Column	Description
Store Name	Indicates the name of the store.
Space Name	Indicates the space that is connected to the store.
Status	Indicates whether the store is: <ul style="list-style-type: none">• Online• Online – Read Only• Offline

Column	Description
Storage Type	Indicates whether the Storage Type is: <ul style="list-style-type: none">• Local file system• Smart lock• xDB• ECS• S3• S3 (proxy)• S3 to Glacier• S3 to Glacier (proxy)• CAS• Custom
Type	Indicates whether the Store type is: <ul style="list-style-type: none">• Regular• Result• Delivery Channel
Used for confirmations	Indicates whether the store is currently being used as a confirmation store.
In Use	Indicates whether the store is currently being used by at least one of the following objects: <ul style="list-style-type: none">• Content store• Delivery channel store• Order item store• Backup store If a store is not in use, user is able to edit or delete it. If a store is in use, user is able to only Edit it.
Side panel	Indicates the details of the selected store.

1. Click '+' to add a store.
2. Enter the following information:

Field	Description
Configuration Label	Enter a name for the store being created.
Application	Select the parent application of the store being created.
Space	Select the space that will be connected to the store being created.
Space Root	<p>Select the specified object of space.</p> <p>There are two types of space roots:</p> <ul style="list-style-type: none"> • Space Root Folder • Space Root Object <p>Each folder or object has its own child object. For a space root folder, it is file system. For a space root object, it is a bucket.</p>
File System Folder/Bucket	<p>Select the specified object of the space root.</p> <p>Click the add button to create a new file system folder or bucket. If adding a new file system, enter a name of the new object and click CREATE.</p> <p>Follow these rules to create a Domain Name System (DNS)-compliant bucket name:</p> <ul style="list-style-type: none"> • Bucket names must be at least three and no more than 63 characters long. • Bucket names must be a series of one or more labels. Adjacent labels are separated by a single period (.). Bucket names can contain lowercase letters, numbers and hyphens. Each label must start and end with a lowercase letter or a number. • Bucket names must not be formatted as an IP address (for example, 192.168.5.4).
Use as Confirmation Store	Indicate if the store is used as confirmation. Available only for filesystem stores.
Status	Indicate the store status.

3. Click **CREATE**.

Editing and Deleting Administration Configuration Objects

This section includes a list of objects along with their respective and it's properties.

When InfoArchive is running (for example, ingesting, searching, etc.), it is very dangerous to update certain fields of existing objects, such as the XfbFederation object's bootstrap field. It is recommended that you test what you are going to do in a development environment before moving to production.

To ensure that references are not broken, objects can be deleted only when "Not In Use".

Whether you can edit a property depends on whether the object is in use or not. After updating a property, InfoArchive provides the ability to test the connection with the new configuration prior to saving your changes.

The following table outlines which properties allow or prohibit the edit and delete functions:

Legend:

- "+" — Object is allowed to be edited or deleted without any issues. Typically, this can be done for unused objects.
- "+/-" — Object is allowed to be edited or deleted with some restrictions or concerns. For example, incorrect configuration may cause a "data unavailability" exception to be issued.
- "-" — Object is not allowed to be edited or deleted in order to not:
 - Break references between objects
 - Lose data availability
 - Move data

Object	Field	Delete		Edit		Notes	
		Not In Use	In Use	Not In Use	In Use		
Federations							
Federation		+	-	+	+	Role: Administrator	
	Federation Name	Not applicable		+	+	Role: Administrator	
	Superuser Password	Not applicable		+	+	Role: Administrator	
	Connection URL	Not applicable		+	+	Role: Administrator	
Databases							
Database		+	-	+	+	Role: Administrator	
	Database Name	Not applicable		+	+	Role: Administrator	
	Admin Password	Not applicable		+	+/- (test connection)	Role: Administrator Change of the password does not change physical xDB password. To achieve this, use xDB Admin.	

	xDB Federation	Not applicable		+	- (read-only property)	Role: Administrator To change the xDB federation, remove the current database from current federation and create another database for the required federation.
Storages						
Isilon		+ (only when no SpaceRootFolders are associated)	-	+	+	
	Storage Name	Not applicable		+	+	Role: Administrator
	Description	Not applicable		+	+	Role: Administrator
	Folder Path	Not applicable		+	+/- (allowed when there is no data)	Role: Administrator Remove the data first.
Local File System		+ (only when no SpaceRoot -Folders are associated)	-	+	+	
	Storage Name	Not applicable		+	+	Role: Administrator
	Description	Not applicable		+	+	Role: Administrator
	Folder Path	Not applicable		+	+/- (allowed when there is no data)	Role: Administrator Remove the data first.

ECS		+ (only when no credentials are associated)	-	+	+	
	Storage Name	Not applicable		+	+	Role: Administrator
	Description	Not applicable		+	+	Role: Administrator
	URL	Not applicable		+	+/- (test connection)	Role: Administrator
	Enable Proxy	Not applicable		-	-	Role: Administrator Required only when proxy enabled
	Proxy URL (required only when proxy enabled)	Not applicable		+	+/- (test connection)	Role: Administrator
	Credentials (Multiple)	+ (only when no Space is associated)	-	+	+/- (test connection)	Role: Administrator
	Credential Name	Not applicable		+	+/- (test connection)	Role: Administrator
	Credential Description	Not applicable		+	+	Role: Administrator
	Access Key	Not applicable		+	+/- (test connection)	Role: Administrator
	Secret Key	Not applicable		+	+/- (test connection)	Role: Administrator

ECS (continued)	Proxy (required only when proxy enabled)	Not applicable	+	+/- (test connection)	Role: Administrator Required only when proxy enabled
	Proxy User Name (required only when proxy enabled)	Not applicable	+	+/- (test connection)	Role: Administrator Required only when proxy enabled
	Proxy User Password (required only when proxy enabled)	Not applicable	+	+/- (test connection)	Role: Administrator Required only when proxy enabled

S3		+ (only when no credentials are associated)	-	+	+	
	Storage Name	Not applicable		+	+	Role: Administrator
	Description	Not applicable		+	+	Role: Administrator
	URL	Not applicable		+	+/- (test connection)	Role: Administrator
	Enable Glacier	Not applicable		-	-	Role: Administrator
	Enable Proxy	Not applicable		-	-	Role: Administrator
	Proxy URL	Not applicable		+	+/- (test connection)	Role: Administrator Required only when proxy enabled
	Credentials (Multiple)	+ (when no Space uses Credential)	-	+	+/- (test connection)	Role: Administrator
	Credential Name	Not applicable		+	+/- (test connection)	Role: Administrator
	Credential Description	Not applicable		+	+/- (test connection)	Role: Administrator
	Access Key	Not applicable		+	+/- (test connection)	Role: Administrator
	Secret Key	Not applicable		+	+/- (test connection)	Role: Administrator

S3 (continued)	Proxy	Not applicable		+	+/- (test connection)	Role: Administrator Required only when proxy enabled
	Proxy User Name	Not applicable		+	+/- (test connection)	Role: Administrator Required only when proxy enabled
	Proxy User Password	Not applicable		+	+/- (test connection)	Role: Administrator Required only when proxy enabled
Custom Storage		+ (when no space is associated)	+	+ (only when not in Use)	+	
	Storage Name	Not applicable		+	+	Role: Administrator
	Description	Not applicable		+	+	Role: Administrator
	Factory Service Name	Not applicable		+	+/- (test connection)	Role: Administrator
	Properties (Multiple)	Possible adding and deleting of properties		+	+/- (test connection)	Role: Administrator
	Name	Not applicable		+	+/- (test connection)	Role: Administrator
	Value	Not applicable		+	+/- (test connection)	Role: Administrator

CAS (Centera)		+ (only when not in use)	+	+	+	
	Storage Name	Not applicable	+	+	Role: Administrator	
	Description	Not applicable	+	+	Role: Administrator	
	Connection String	Not applicable	+	+/- (test connection)	Role: Administrator	
	PEA (multiple)	Adding and deleting of property	+	+/- (test connection)	Role: Administrator	
	Variable	Not applicable	+	+/- (test connection)	Role: Administrator	
	Content	Not applicable	+	+/- (test connection)	Role: Administrator	
Spaces						
Space	+ (only when none of the following objects refer to it: SpaceRoot -XdbLibrary, SpaseRoot -Folder, or SpaceRoot -Object)	-	+	+	Role: Administrator, Developer Space can be removed as long as the following objects do not refer it: <ul style="list-style-type: none">• SpaceRootXdb Library,• SpaseRoot Folder,• SpaceRoot Object	
	Application	Not Applicable	-	-	Role: Administrator, Developer	

	Space Name	Not Applicable	+	+	<p>Role: Administrator, Developer</p> <p>Application for the space cannot be changed. To change the application, remove the space and create a new space for the required application.</p>
	Database Library	Not Applicable	+	+	<p>Role: Administrator, Developer</p> <p>To change the database, remove the attached database, then add the new one. Attached database can be removed only when SpaceRootXdb - Library object that represents connection is not in use.</p>

	Storage System	Not Applicable	+	+	<p>Role: Administrator, Developer</p> <p>Connection with additional storage systems can be added at any time.</p> <p>Connection with existing storage systems can be removed only when the object that presents connection is not in use. The following two types of objects are used, depending on the storage type: SpaceRootFolder and SpaceRootObject .</p>
Stores					
Store		+ (only when no objects use the store)	-	+	+
	Store name	Not Applicable	+	+	Role: Administrator, Developer
	Application	Not Applicable	-	-	Role: Administrator, Developer
	Space	Not Applicable	+	-	Role: Administrator, Developer

	Space Root	Not Applicable	+	-	Role: Administrator, Developer
	File System Folder /Bucket	Not Applicable	+	-	Role: Administrator, Developer
	Status	Not Applicable	+	+	Role: Administrator, Developer
	S3 to Glacier Settings, if applicable	Not Applicable	+	+	Role: Administrator, Developer
Encryption					
CryptoObject		+	-	+	Role: Administrator, Developer Crypto objects can be removed and edited from the user interface only when not in use. It is risky to change any crypto configuration when the object is in use.
	Name	Not Applicable	+	-	Role: Administrator, Developer
	Service Provider	Not Applicable	+	-	Role: Administrator, Developer
	Key Size	Not Applicable	+	-	Role: Administrator, Developer
	Encryption Algorithm	Not Applicable	+	-	Role: Administrator, Developer
	Encryption Mode	Not Applicable	+	-	Role: Administrator, Developer

	Padding	Not Applicable	+	-	Role: Administrator, Developer
	Service provider = Germalto	Not Applicable	+	-	Role: Administrator, Developer
	NAE Name	Not Applicable	+	-	Role: Administrator, Developer
	NAE Username	Not Applicable	+	-	Role: Administrator, Developer
	NAE Password	Not Applicable	+	-	Role: Administrator, Developer
	NAE Group	Not Applicable	+	-	Role: Administrator, Developer
	NAE Alias	Not Applicable	+	-	Role: Administrator, Developer
	NAE Keystore Password	Not Applicable	+	-	Role: Administrator, Developer

Configuring a Holding

The following section is applicable for SIP applications only.

A holding is a logical destination where data that shares common characteristics is archived. Some example of common characteristics include:

- Data from the same source application;
- Data in the same format, such as audio recordings;
- The same type of data (for example, communication such as e-mail, chat, faxes, etc.); and
- Data that belongs to the same business entity.

An application can contain multiple holdings. Multiple holdings can exist for a single data type, which means you are able to apply different access rights or use a different storage area.

A holding is also the central configuration object in SIP-based archiving. The following is defined in a holding:

- storage area
- retention policy

- ingestion sequence
- the AIP mode and xDB mode being used

When creating a holding, consider the types of data that will be archived, as well as the data segregation/isolation restrictions.

In general, ingestion and search performance depends on the application configuration, holding configuration and the following external IT factors:

- Number of AIUs per SIP
- Ingestion mode
- xDB indexes
- Partitioning keys

This section illustrates how to configure a holding. By the end of the section, it will be possible to ingest data, perform a search and retrieve content from IA Web App.

The preconditions to start are:

- Know what is to be archived, and have defined and generated an XML schema.
- Know what is to be searched.
- Know how many SIPs per day are expected to be archived.
- Know how many AIUs per SIP are expected to be preserved.
- Know the retention date range and how long data should be stored.
- Know the average volume of the SIP package (in Mb).
- Know the average volume of the pdi.xml file inside the SIP package (in Mb).

Refer to [Improving Performance for a SIP Archive](#) for further information.

Configuring a Holding Using the Holding Configuration Wizard

The **Holdings** tab contains a list of existing holdings appear in a table that contains the following information:

Column	Description
Holding Name	Indicates the name of the holding.
Holding Description	Contains a brief description of the holding and its purpose.
In Use	<p>Indicates whether the holding is currently being used.</p> <p>In use is calculated by looking for any AIP that is ingested and if the holding is used in any search.</p> <p>If a holding is not in use, user is able to delete it.</p> <p>If the holding has been created via the Holding Wizard, and the holding is not in use, then the holding can be edited via the Holding Wizard. The action to edit a holding is available in the context menu for every holding. The steps for editing the holding via holding wizard is the same as creating a new holding.</p>

A panel on the right side of the screen contains the custom properties of a selected holding. The following tabs appear in the panel:

Tab	Description
Summary	Contains general information about the holding.
Store	Contains information about stores that are connected to the holding.
Retention	Contains information about retention classes.
xDB	Contains information about xDB databases and libraries.
Permissions	Contains a list of restrictions for the holding.
Confirmations	Contains information about confirmation options and store.

The **Holdings** tab includes a wizard that helps the developer to create and configure a holding for an application. The InfoArchive holding configuration wizard helps you build holding configuration from the XSD of the holding data structure. This wizard exposes the most common options, such as the selection of an AIU, indexes and partitioning keys, ingestion mode, retention mode, etc.

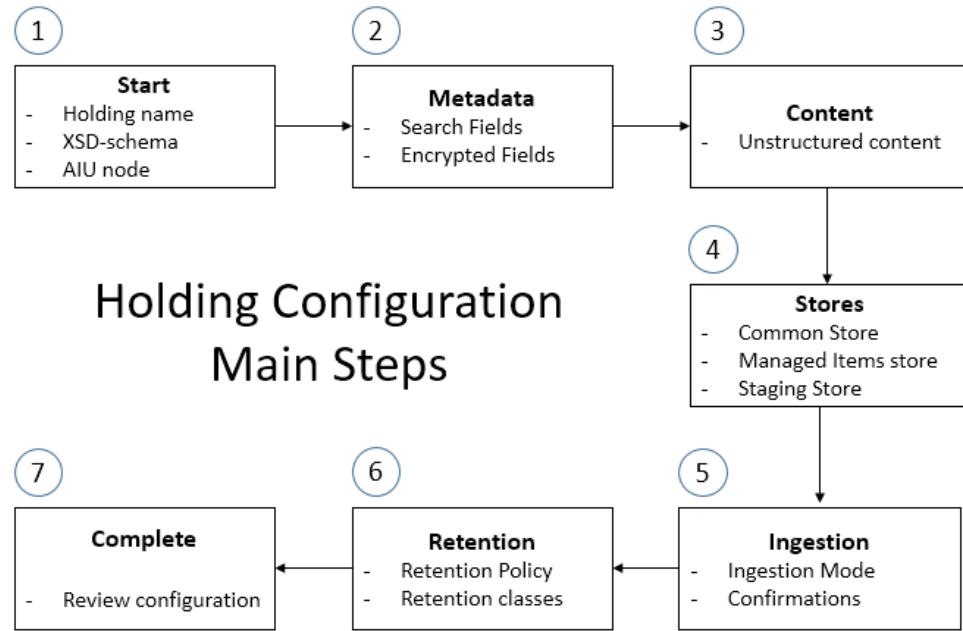
After completing all of the steps in the wizard, the following actions can be performed:

- The wizard lets you save an unfinished holding configuration and complete it at a later time. You can load the saved configuration file later to continue with the configuration. When you load an existing holding configuration that you saved earlier, you are returned to the **Holdings** tab and resume from there;

An unfinished holding configuration can also be deleted.

- Install a holding configuration when it is fully created;
- Export a holding configuration into a declarative format.

The following illustrates the main steps you have to complete to create a holding. The numbers relate to the steps in the procedure that



follows:

To create a holding, complete the following steps:

1. On the [APPLICATION NAME] > Holdings tab, click + > **Create Holding** to initiate the wizard.
 - a. The **Start** step of the wizard allows you to enter the following preliminary information:

Field	Description
Holding Name	<p>Specify a descriptive name that uniquely identifies the holding.</p> <p>The first character of the name cannot contain the following characters: # < > ! = \$ ` & * () + \ " ' : ? @ % { } [] , -</p> <p>The name cannot contain the following characters: # < > ! = \$ ` & * () + \ /</p> <p>A holding is a logical destination archive where to ingest and store data, usually of the same type that share common characteristics. For example, you can create a holding to archive data from the same source application (such as ERP data), or of the same format (such as audio recordings), or belonging to the same business entity.</p> <p>An application can contain multiple archive holdings. The SIP descriptor (<code>eas_sip.xml</code>) contains the name of the holding to be used for data archiving.</p>
Description	Enter a brief description of the holding that outlines its purpose.
Application Space	Select the space associated with the application. If only one space has been configured for the application, it will appear by default.

Field	Description
Schema File	<p>There is an option of uploading a new XSD file or selecting an existing schema file. Refer to Mapping XSD Data Types for further information.</p> <p>Select the schema that formally describes the structured data in the information packages to archive. The specified schema will be imported into the repository as the content of the schema configuration object.</p> <p>The user has the option to upload several schemas.</p>
Archival Information Unit (AIU) Node	<p>Select the AIU node. From the schema diagram, select the node that represents the archival information unit (AIU). This diagram is a graphical representation of the structured data (PDI) schema.</p> <p>Use the following navigational operations to locate the AIU node:</p> <ul style="list-style-type: none"> • Click the plus sign (+) on a node to expand it; click the minus sign (-) to collapse a node • Click the plus sign (+) and the minus sign (-) on the left side of the diagram to expand/collapse the nodes all at once <p>Also, at the top there is slider that allows zoom in\out actions.</p> <p>Note: Make sure you select the correct node that represents the AIU. The wizard does not validate your selection. If you select the wrong node, ingestion will fail.</p> <p>An archival information units (AIU) is, conceptually, the smallest archival unit (like an information atom) of an information package. Each AIU corresponds to a record or item of the archived data. A single customer order, a patient profile, or a financial transaction record in an information package is an AIU.</p> <p>The structured data (PDI) file (<code>eas_pdi.xml</code>) in a SIP describes all the AIUs in the package. An AIU in the <code>eas_pdi.xml</code> file consists of an XML block in the file containing its structured data, and optionally, references to one or more associated unstructured content files.</p> <p>There is a possibility to delete the selected AIU node after it has been selected.</p>

- b. Click **NEXT**.

2. The **Metadata > Search Fields** step of the wizard allows you to select the fields that you want available for search composition. The developer should select at least one search field.

- Click **Select from Schema** to select the desired fields.

Use the following navigational operations to locate the desired fields:

- Click the plus sign (+) on a field to expand it; click the minus sign (-) to collapse a field
- Click the plus sign (+) and the minus sign (-) on the left side of the diagram to expand/collapse the fields all at once

Also at the top there is slider that allows zoom in\out actions.

- Click **SELECT**.
- Set the **Index Type** and **Partition Key Method** for each field:

Index Type	<p>Creating xDB indexes speeds up searches but consumes more storage space. Refer to Index Types in the XQuery Optimization section for further information.</p> <p>InfoArchive supports path value indexes and full-text indexes:</p> <ul style="list-style-type: none"> Path Value: Most common index type of index that consumes less storage. Full Text: Less common type of index that consumes more storage. <p>Refer to The Indexing Process for further information.</p>
Partition Key Method	<p>A partition key is used in the first tier of the query process to limit the data returned when a search is executed. Partition keys are created during ingestion using XQuery and are stored in the AIP object. Refer to Using Partition Keys for further information.</p> <p>The following partitioning techniques at the time of data ingestion help with search performance:</p> <ul style="list-style-type: none"> None: No partitioning techniques were added to the AIP object prior to ingestion Min/Max: Uses the minimum and maximum number of records in the package. List: Uses the unique values of the records in the package.

It is possible to delete previously selected fields.

- The **Encryption Fields** tab allows you to select the fields from a PDI schema that should be encrypted at the time of ingestion.
 - Select the **Data encryption object** from the drop-down list.
 - Click **Select from Schema** to select the desired fields.

Use the following navigational operations to locate the desired fields:

- Click the plus sign (+) on a field to expand it; click the minus sign (-) to collapse a field
- Click the plus sign (+) and the minus sign (-) on the left side of the diagram to expand/collapse the fields all at once

Also at the top there is slider that allows zoom in\out actions.

- iii. Click **SELECT**.
- e. Click **NEXT**.

The **NEXT** button is disabled in two cases:

 - If an encryption object was selected and no field has been selected; or
 - If at least one field is selected and no encryption object has been selected.
3. The **Content** step of the wizard allows you to define a content field.
 - a. Select one of the following:
 - **Unstructured content is not associated with record.** If selected, click **NEXT**.
 - **Unstructured content is associated with record.** If selected, complete the following sub-steps:
 - i. Select a **Content encryption object** (optional).
 - ii. Select the field from schema that represent file names for unstructured content (for example, documents, images or PDFs). If the desired file format is not listed, click **Add new file format**:
 - Select the **Mime Type** from the list.
 - Select the **Mime Sub Type** from the list.
 - Click **ADD**.
 - iii. Click **Select from Schema**.

Use the following navigational operations to locate the desired fields:

 - Click the plus sign (+) on a field to expand it; click the minus sign (-) to collapse a field
 - Click the plus sign (+) and the minus sign (-) on the left side of the diagram to expand/collapse the fields all at once

Also at the top there is slider that allows zoom in\out actions.
 - iv. Click **SELECT**.
 - v. Select File Format form the list of existing formats or add a new one.
 - vi. Specify Additional Settings, if needed:
 - i. Click **Additional Setting**.
 - ii. Set Byte Offset and Byte Length by selecting nodes on the schema.

This fields are mainly for PDF content to set first page and page count of a document.

- iii. Click to allow or disallow whether the field is eligible for:
- **Encryption:** Click to allow encryption of the unstructured content files using the selected encryption object. Only enabled if you select an encryption object from the drop-down list.
 - **Compression:** To save storage space, click to allow unstructured content files to compress with the gzip compression algorithm.
- iv. Click **NEXT**.
4. The **Stores** step of the wizard allows you to configure the application store settings.
- a. Complete the following, as required:

Keep SIP file after ingestion	If selected, a SIP file that has been successfully ingested will be stored
Keep SIP file upon any invalidation	If selected, a SIP file that has not been successfully ingested will be stored
Keep PDI XML after commit	If selected, the PDI (<code>eas_pdi.xml</code>) file will be stored once data has been committed.

Enable Log store	If selected, reception and ingestion logs will be stored, either in a common store with other data or in a separate store.
Common store for all unstructured data	<p>If selected, all unstructured data will be stored in a common store. Select values for the following:</p> <ul style="list-style-type: none"> • Common Store • Managed Item Store • Staging Store <p>If not selected, select values for the following:</p> <ul style="list-style-type: none"> • Reception Store: This store is used to save the original SIP (<code>sip_zip</code>). o The SIP is archived as other content if the customer decides to keep the SIP file after the ingestion. <p>The reception store is used if the reception and the ingestion are performed in two steps. At the end of the reception, the SIP file is uploaded to the reception store to be picked up later during the ingestion.</p> <p>The reception store is not a working directory like <code>data/tmp</code>.</p> <ul style="list-style-type: none"> • Log Store: Saves log files generated during the reception, ingestion and invalidation. • XML Store: Saves the SIP XML, as well as the the original PDI XML file (gzip format), if the customer decides to keep the PDI after the ingestion. • Content Store: Saves the Cis. This includes the CI Container and the RI XML files. • xDB Library Store: Saves the xDB library backup containing the metadata • Rendition Store: Saves the analytic rendition (optional). The analytic rendition can be generated during the ingestion or requested after. An analytic rendition can be a flat file (CSV), which is easily consumed by Hadoop and company frameworks. • Managed Item Store: Saves the retention xDB library backup (optional). • Staging Store: This optional store is used with the <code>contentstore</code> and <code>renditionstore</code> stores.

- c. Select Aggregation store, which indicates the store for packages to be staged for aggregation during ingestion.
- d. Click **NEXT**.
5. The Ingestion > Ingestion Mode step of the wizard allows you to indicate which ingestion mode will be applied to the holding and which events will be confirmed.

- a. According to the ingestion mode selected, the wizard presents the corresponding options for you to configure. Refer to SIP ingestion modes for further information.

For the Pooled and Aggregated modes, when an AIP is ingested, InfoArchive searches for an available xDB pooled library in which to store its structured data. An available xDB pooled library must be one that is not closed and has not reached its close condition or storage quota. If no such pooled library is found, a new pooled library is created.

If Pooled or Aggregated is selected, specify the settings for the library pool.

Field	Description
Maximum number of packages in library	An xDB library stores archived data. Specify the maximum number of AIPs that can be stored in a library.
Maximum number of records in a package	Specify the maximum number of records that an AIP can contain.
Library closing	Specify the close condition for the library: <ul style="list-style-type: none"> • None • Creation Date: CLOSE PERIOD + XDB LIBRARY CREATION_DATE • Last Modified Date: CLOSE PERIOD + XDB LIBRARY LAST MODIFIED DATE
Library partitioning	Specify the time-based partitioning logic (for example, by week, month, quarter, or year). For example, in a quarter partitioning assignment policy, AIPs archived during different quarters are assigned to their respective quarter-based xDB pooled libraries. Multiple pooled libraries can be open at a single point in time for a given library pool. To archive data from in 2018 Q3 through Q4 with a quarter partitioning policy, the corresponding xDB pooled libraries must remain open. When there is no more data to be archived for a particular quarter, the library corresponding to that quarter can be closed.
Private Mode Switch > Private Mode Criteria	Specify whether you want to switch to the private ingestion mode for certain packages by entering the applicable criteria. If it is possible to store a huge number of AIUs (for example, 100,000) in the SIP package , then it is fine to use the private mode. In this mode, indexes are created during ingestion. Refer to SIP ingestion modes for further information.

- b. The **Ingestion > Confirmations** step of the wizard allows you to select the confirmation store and the event types for which you want to generate confirmation message tab.

A confirmation is a message generated in response to an AIP event to acknowledge that the event has occurred and to capture the information of the relevant AIP.

The wizard only provides the most basic confirmation configuration options. You can perform additional manual configurations to:

- Generate multiple confirmations for a single event and configure the content of the messages.
- Apply the scope of confirmation generation to a specified set of AIPs.
- Configure where to output confirmation messages: xDB or a file system location.

- c. Click **NEXT**.
6. The **Retention > Retention Policy** step of the wizard allows you to configure at ingestion if retention should be applied to the package or to records.
 - a. Select the type of retention:
 - Do not apply retention
 - Package – Retention will be applied to entire package
 - Granular – Retention will be applied to each record
 - b. If **Package** was selected, then select whether to use:
 - No default retention policy for the holding;
 - The application's default retention policy; or
 - Another retention policy as a default for the holding. Select the policy from the drop-down list.

Event and mixed retention policies are not shown in the list.
 - c. If **Granular** was selected:
 - Select retention policy and base date. Only duration based retention polices can be used.
 - Define Search and Search Set names if they are defined and you want to generate a purge candidates list. Purge list generation depends on a search template to display columns for records.
 - d. The **Retention > Retention Classes** step of the wizard allows you to add or remove retention classes. This step is only shown if type of retention is Package. Retention classes can be referred to by a SIP to help decide which policy to apply to the package.
- To add a new retention class:
 - i. Click 
 - ii. Enter a name in the **Retention Class Name** field.
 - iii. Select a policy from the **Retention Policy** list.
- e. The Retention > Retention Partitioning step of the wizard allows you to set up partitioning schema for a package.

To select a partitioning schema, select a value from the drop-down list.

Click **NEXT**.
7. Review the information for the holding. When satisfied, click **FINISH**.

To export the holding configuration in a YAML or declarative ZIP format, click the down arrow beside the **FINISH** button.

Limitations of the Holding Wizard

- The installation of a holding through the wizard fails when a space is included in the holding name.
- The holding name cannot contain the following characters: “/ # < > ! = \$ & () * + \ " ' : ? | @ % { } [] ~ ; , ^ -”

Holding Store Configuration

The **Holdings** tab in IA Web App contains a list of the available holdings.

Furthermore, use the Stores tab to view the available storage.

The following is a breakdown of the holding configuration for a store:

Reception Store	Indicates where the entire received SIP will be stored.
Log Store	Indicates where the log files will be stored.
XML Store	Indicates where the XML files will be stored (PDI XML, SIP XML).
Content Store	Indicates where the content information containers items and RI XML will be stored.
xDB Store	Indicates where the xdb sub-library will be stored.
Analytics Rendition Store	Indicates where analytical rendition and purge list exports will be stored.
Managed Item Store	Indicates where the retention data (managed items data) will be stored.
Staging Store	Indicates where the read/write operations prior to closing the aggregate (reception, ingestion, invalidation, etc.) will happen.

Staging Store and Centera

Some storage, especially Centera, are not designed to save temporary data. The staging store is a temporary store, usually on the local disk, that can be used for all the write operations prior to closing the aggregate (reception, ingestion, invalidation, etc.).

The staging store is defined at the Holding level. When the staging store is enabled, all the write operations will go to this store.

Glacier

Using an offline store, such as Glacier, may impact performance if the content of an AIP, (logs, container, sip.xml, pdi.xml files, etc.) is offline. Consequently, at the UI level in the **Packages** tab, the AIP will not be available. Instead, a restore button will be displayed that allows you to request a restore of the content. For example, if the logStore is a Glacier store, in the **Package** tab, the log files will not be available if they are offline.

- **ciStore:** If a Glacier store is chosen as ciStore, then the search result will, potentially, not have a download link to the content results. Instead, a restore button will be displayed. In case of confirmation, it will take longer if some content is offline, as the system needs to retrieve the content before you are able to perform a confirmation. And, indirectly, the same functionality occurs in the case of invalidation because the AIP invalidation must perform a confirmation. During AIU disposition, the ri and container file are needed, too. If this content is offline, the AIU disposition will take longer.
- **xdbStore:** The back up of the xDB libraries will potentially be offline. In this event, a cache-in of the xdblibrary will take more time because the process must restore the content from Glacier. During a search, if some AIPS of the search are cached out, and the back up is offline, then the search will take more time. This will impact the confirmation, export and the transformation processes because of cache-in request initiated during the process.
- **xmlStore:** If a Glacier store is chosen as an xmlStore, it can potentially impact the transformation and the confirmation processes because they use the pdi.xml file according to their settings.

Pooled Libraries

Configuring the Ingestion Process to Store Multiple AIPs in the Same xDB Library

When working in the aggregation mode, InfoArchive allows you to store metadata into shared libraries, which are also known as pooled libraries.

The effective close date is computed with the following rules:

Close Mode	Rule
NONE	nulldate
LAST_MODIFIED_DATE	CLOSE PERIOD + XDB LIBRARY LAST MODIFIED DATE
CREATION_DATE	CLOSE PERIOD + XDB LIBRARY CREATION_DATE
CLOSE_HINT_DATE	CLOSE PERIOD + XQUERY DATE TIME

The XdbLibraryPolicy is referenced to the Holding: in the following attributes:

- xdbMode
- xdbLibraryPolicy

The following attributes are added at the AIP level:

- xdbMode
- phaseCode
- stateCode

The following attributes are added at the XdbLibrary level to facilitate the library management:

- pkey
- aipCount
- aiuCount
- effectiveCloseDate
- closed
- closedDate
- closeRequested
- xdbMode

Executing the Close Job to Close the xDB Libraries and Perform a Back Up

When you run the Close job for each xDBLibrary:

- An xDB back up is performed;
- The XdbLibrary is set to closed = TRUE; and
- The closeDate value is updated.

The back up is only performed if every AIP of the xdbLibrary is set to COMPLETED.

An xDBLibrary with the xdbMode set to PRIVATE is not eligible to be closed. Only an xDBLibrary in xdbMode POOLED or AGGREGATE can be closed:

```
XdbLibrary where xdbMode in (POOLED,AGGREGATE) and closed = FALSE and
(effectiveCloseDate < (offsetDateTime(now) - {closeDelay}) or closeRequested = true)
```

Note: An exceeded quota never triggers a closure. Only an outdated closing date or a manual close can be taken into account.

The job accepts the following parameters to control the execution:

- phaseToProcess
- closeDelay

Refer to [Close Job](#) for further information.

Partition Keys

A partition key is used in the first tier of the query process to limit the data returned when a search is executed. Partition keys are created during ingestion using XQuery and are stored in the AIP object.

An AIP can have multiple partition keys to satisfy different sets of search criteria.

To improve the search performance and reduce the search scope, define one or more partition keys. When a search criterion is linked to a partition key, perform the query only on AIPs that reference the partition key value.

Each partition key value must be assigned to an AIP attribute. Out of the box, the AIP offers some free slots. You need to take into account the type (STRING, INTEGER, DOUBLE, DATETIME, LIST<STRING>).

Using search without defining partition keys may result in the following consequences:

- SIP-based searches will be sensitive to the amount of ingested data (refer to [How Data is Searched](#) for more information). The response time will increase linearly with increasing data volume.
- Background searches will consume a lot of CPU and, therefore, will have an impact on the overall system.

Note: To significantly improve ingestion performance, all fields used as partitioning keys must also be defined as indexes.

Types of Partition Keys

There are different types of partition keys:

Type	Number of Values	Attribute
DateTime	6	pkeys.dateTime01
String	4	pkeys.string01
List of String	4	pkeys.values01
Integer	4	pkeys.integer01
Long	4	pkeys.long01
Double	4	pkeys.double01

Indexes

Types of Indexes

There are two types of indexes:

- A path value index (path.value.index) is the most common of the two index types. It indexes the value of elements and attributes. Furthermore, the values of multiple elements or attributes can be used in the key to create a composite index.
- A full-text index indexes the values of elements and attributes but tokenizes the values into a number of terms, and each term or element combination is added to the index. While consuming more storage, a full-text index enables users to search for an individual word contained in the indexed values. It also allows for the use of wildcards in a search is less sensitive to misspelling entered as search criteria.

Path Value Indexing	MultiPath Indexing
It can be used for indexing multiple elements, but requires every single element to be explicitly listed in the index definition.	Multipath indexes allow you to specify sub-paths with wildcards that will match more than one element path, so not every element has to be explicitly listed. Making multipath indexes much more flexible and easy to use.
Smaller size means that it is faster to ingest	Large index size
Better performance if you know the query ahead of time along with the number of predicates	Only option for table archiving
B-tree index	Lucene Inverted index

Refer to the *Encryption Guide* to learn about the indexing of encrypted fields.

Configuring Indexes for a SIP Archive

An index is computed during ingestion and covers the data contained in one AIP. Indexes are used in the second tier of a search, whereby the system scans packages for individual results (AIUs) via the use of indexes.

An AIP can have many indexes defined to satisfy different search criteria.

Configuring Indexing

The following outlines the information contained in an index:

```
<data>
<id>pdi.index.creator</id> ①
<key.document.name>xdb.pdi.name</key.document.name>
<indexes>
... ②
```

```
</indexes>  
</data>
```

1. Indicates the ID of the index creator processor, which must be pdi.index.creator.
2. Indicates the indexes to create, path.value.index or full.text.index.

Using a Path Value Index

The path value index is defined to index the AIU based on a value:

```
PATH_TO_AIU[CONDITION<TYPE>]
```

The following is an example of the PDI:

```
<objects>  
  <object>  
    <customerid>abc123</customerid>  
    ...  
  </object>  
</objects>
```

The entity path is /{ns}objects/{ns}object

To create an index, for example, on the value 'customer id', you would use the following:

```
/ {ns}objects/{ns}object[{ns}customerid<STRING>]
```

The following is an example of a path value index:

```
<path.value.index>  
  <name>tweetId</name>  
  <path>  
    /{urn:x-emc:ia:demo:1.0}objects/{urn:x-emc:ia:demo:1.0}object  
    [{ur n:xemc:ia:demo:1.0}id<STRING>]  
  </path>  
  <compressed>false</compressed>  
  <unique.keys>false</unique.keys>  
  <concurrent>false</concurrent>  
  <build.without.logging>true</build.without.logging>  
</path.value.index>
```

The following is an example of a full-text index:

```
<full.text.index>  
  <name>text-fulltext</name>  
  <compressed>false</compressed>  
  <concurrent>false</concurrent>  
  <optimize.leading.wildcard.search>true</optimize.leading.wildcard.search>  
  <index.all.text>true</index.all.text>  
  <include.attributes>false</include.attributes>  
  <support.phrases>false</support.phrases>  
  <support.scoring>false</support.scoring>  
  <convert.terms.to.lowercase>true</convert.terms.to.lowercase>  
  <filter.english.stop.words>false</filter.english.stop.words>  
  <support.start.end.token.flags>false</support.start.end.token.flags>  
  <element.uri>urn:x-emc:ia:demo:schema:tweets:1.0</element.uri>  
  <element.name>text</element.name>  
  <attribute.uri/>  
  <attribute.name/>  
</full.text.index>
```

Configuring Indexes for a Table Archive

Indexing needs to be enabled at the column level in the `metadata.xml` file. Multi-path indexes are created at the end of the ingestion process. The task can be configured in the `tools > build-table.xml` file.

Creating Path Value Indexing

Table archiving only supports multi-path value indexing. To save time, the Developer can use a path value index for specific search queries in table archiving, and is able to specify the path value in the configuration for selected fields during or after ingestion. Then, after the ingestion of data, the Developer:

- Has path value indexes created for specified fields; or
- Can trigger path value indexing manually when appropriate (for example, after all data is ingested and verified).

The following is an example of a path value index:

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata>
<defaultSchema>SAKILA</defaultSchema>
<schemaMetadataList>
<schemaMetadata>
<name>sakila</name>
<tableCount>1</tableCount>
<tableMetadataList>
<tableMetadata>
<name>actor</name>
<recordCount>4</recordCount>
<columnList>
<column>
<name>ACTOR_ID</name>
<ordinal>1</ordinal>
<type>INTEGER</type>
<typeLength>32</typeLength>
</column>
...
</columnList>
<pathValueIndexList>
<pathValueIndex>
<name>optional</name>
<uniqueKey>false</uniqueKey>
<column>ACTOR_ID</column>
<column>FIRST_NAME</column>
<column>LAST_NAME</column>
<fulltext>
<column>LAST_NAME</column>
<lowercase>true</lowercase>
</fulltext >
</pathValueIndex>
</pathValueIndexList>
<anotherIndexList>
<anotherIndex>
...
</anotherIndex>
</anotherIndexList>
</tableMetadata>
```

```
</tableMetadataList>
</schemaMetadata>
</schemaMetadataList>
</metadata>
```

It results to the following index definition: /sakila/actor/ROW[ACTOR_ID<INTEGER> + FIRST_NAME<STRING> + LAST_NAME<SA_ADJUST_TO_LOWERCASE>]

The following is the XSD definition for a table element:

```
<xs:complexType name="tableMetadataType">
<xs:sequence>
<xs:element type="xs:string" name="name" minOccurs="1" maxOccurs="1"/>
<xs:element type="xs:int" name="recordCount" minOccurs="0" maxOccurs="1"/>
<xs:element name="columnList" minOccurs="0" maxOccurs="1">
<xs:complexType>
<xs:sequence>
<xs:element type="columnType" name="column" minOccurs="1" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="pathValueIndexList" minOccurs="0" maxOccurs="1">
<xs:complexType>
<xs:sequence>
<xs:element type="pathValueIndexType" name="pathValueIndex" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="pathValueIndexType">
<xs:sequence>
<xs:element type="xs:string" name="name" minOccurs="0" maxOccurs="1"/>
<xs:element type="xs:string" name="uniqueKey" minOccurs="0" maxOccurs="1"/>
<xs:element type="xs:string" name="column" minOccurs="1" maxOccurs="unbounded"/>
<xs:element name="fulltext" minOccurs="0" maxOccurs="1">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:string" name="column" minOccurs="1" maxOccurs="1"/>
<xs:element type="xs:boolean" name="lowercase" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
```

The optional path value index definition:

```
<pathValueIndexList>
<pathValueIndex>
<name>optional name if absents then it is named by IA server</name>
<uniqueKey>false</uniqueKey> // optional
<column>ACTOR_ID</column> // LIST of the table columns participating in the index
<column>FIRST_NAME</column>
<column>LAST_NAME</column>
<fulltext> // optional
<column>LAST_NAME</column> // POINTS to one of the above columns (ACTOR_ID, FIRST_NAME or LAST_NAME)
<lowercase>true</lowercase>
</fulltext >
</pathValueIndex>
</pathValueIndexList>
```

Confirmation

A confirmation is a message generated in reaction to an AIP event. Confirmation messages are generated during a package's lifecycle transitions.

The confirmation mechanism acknowledges to the source applications or notifies other business applications, such as a portal, that new objects have been archived. For example, once a SIP has been ingested and committed, a storage confirmation message is generated and can be passed back to the source application to confirm that the data has been correctly archived. The message can be used to trigger the deletion of the data from the source application or to register the content IDs.

A confirmation message can be generated from the SIP descriptor or the PDI metadata files. The PDI metadata is only accessible for two events:

- Storage
- Purge

To perform a query on the PDI metadata, the AIP needs to be online. If the AIP is not online, the system requests a cache in. In this case, the confirmation is delayed. To perform a query on the SIP descriptor, the file needs to be immediately accessible in the XML store. If the file is archived to Glacier, a restoration is requested and the confirmation is delayed.

The following event types can trigger a confirmation:

Type	Description	SIP Query	PDI Query	Priority
Receipt	Indicates that the SIP has been received.	Available	Not available	Custom SIP XQuery > Default SIP XQuery
Storage	Indicates that the AIP ingestion has been ingested and committed.	Available	Available	Custom PDI XQuery > Custom SIP XQuery > Default SIP XQuery
Reject	Indicates that the AIP has been rejected.	Available	Not available	Custom SIP XQuery > Default SIP XQuery
Invalid	Indicates that the AIP has been invalidated.	Available	Not available	Custom SIP XQuery > Default SIP XQuery
Purge	Indicates that the AIP has been disposed.	Available	Available	Custom PDI XQuery > Custom SIP XQuery > Default SIP XQuery

The Confirmation Job and the Aggregates

The Confirmation job identifies open or closed aggregates:

- Open aggregates are not confirmed. An aggregate is considered open under the following conditions:
 - the xdbMode is set to AGGREGATE; and
 - the Phase is set to Phase.AGGR; and
 - the isPartOfAggregate is set to false.If these conditions are met, the AIP is skipped.
- Child AIPs are processed as regular AIPs, and go through the current confirmation as regular AIPs.
- Closed aggregates AIP:

A closed AIP has the following status:

```
xdbMode == AGGREGATE) && (Aip.State == State.COM OR State.INV_WPROC) &&  
Aip.isPartOfAggregate == false )
```

Closed AIPs can either be completed or invalidated.

- SIP Confirmation:

A closed aggregate contains all the children's sip.xml in a tar.gz file. The confirmation is completed on each individual sip.xml in the tar.gz archive.
- PDI confirmation:

A closed aggregate library contains all the PDI files. The confirmation will be completed for all PDI files.

Cache-In/Cache-Out Feature

To enable the cache-in/cache-out feature for an application, two things need to be done:

Cache Size of the Application

The cache size of the SIP applications must be set with the limit size of data that the user wants in the cache.

Default value is 0 and means **unlimited** (= Automatic cache-in/cache-out disabled).

Cache-Out Job

The Cache-Out job is responsible to automatically cache-out libraries from the metadata repository to ensure the size of the metadata of the application stays within the size limit.

This job can be found on the **Administration > Jobs** tab of IA Web App.

By default, the job is scheduled to execute every 15 minutes. As with every other InfoArchive job, its schedule must be started manually the first time.

In addition to the standard InfoArchive job options, there is a specific parameter *maxLibraryPerApplication* in the Edit Job screen.

This field is the maximum number of xDB libraries that will be removed from the cache for an application for one job execution. The default value is **1000**. For example, if this field is set to N:

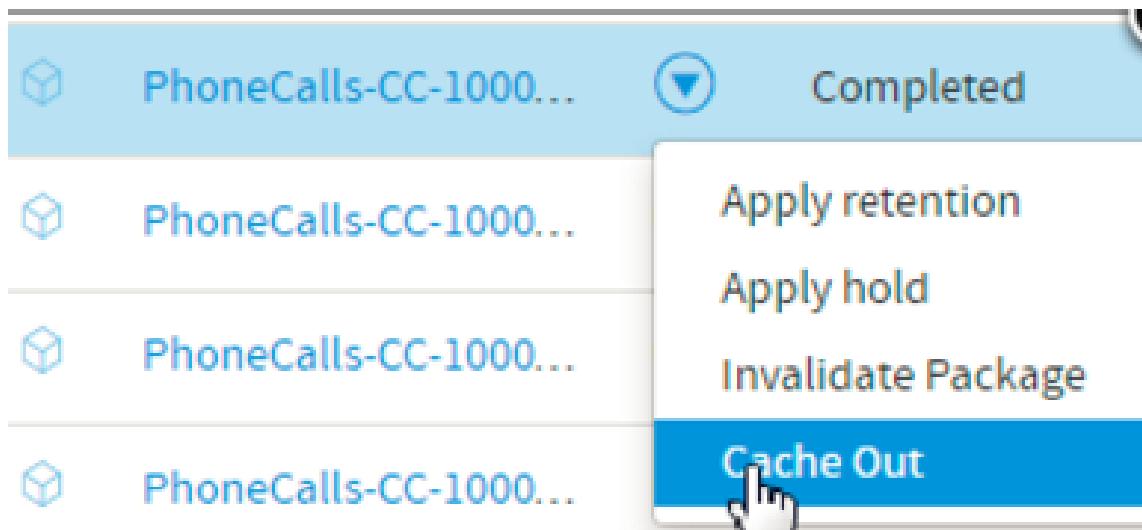
- There is a SIP application for which the total size of metadata in the metadata repository is 200MB above the metadataCacheSize set for this application. For example, assume there is a SIP application for which the total size of metadata in the metdata repository is 200MB above the limit.
- When the Cache-Out job executes:
 - It sees that this application is over its limit.
 - It then starts to cache-out libraries based on statistics.
 - After each cache-out, the job checks if the size is now under the limit. If not, it continues.
 - Even if, after caching out N libraries, the size is still over the limit, this job execution will stop caching out libraries for this application.
- With subsequent executions, the size will eventually go under the limit.

Troubleshooting

Manual Cache-In/Cache-Out Requests

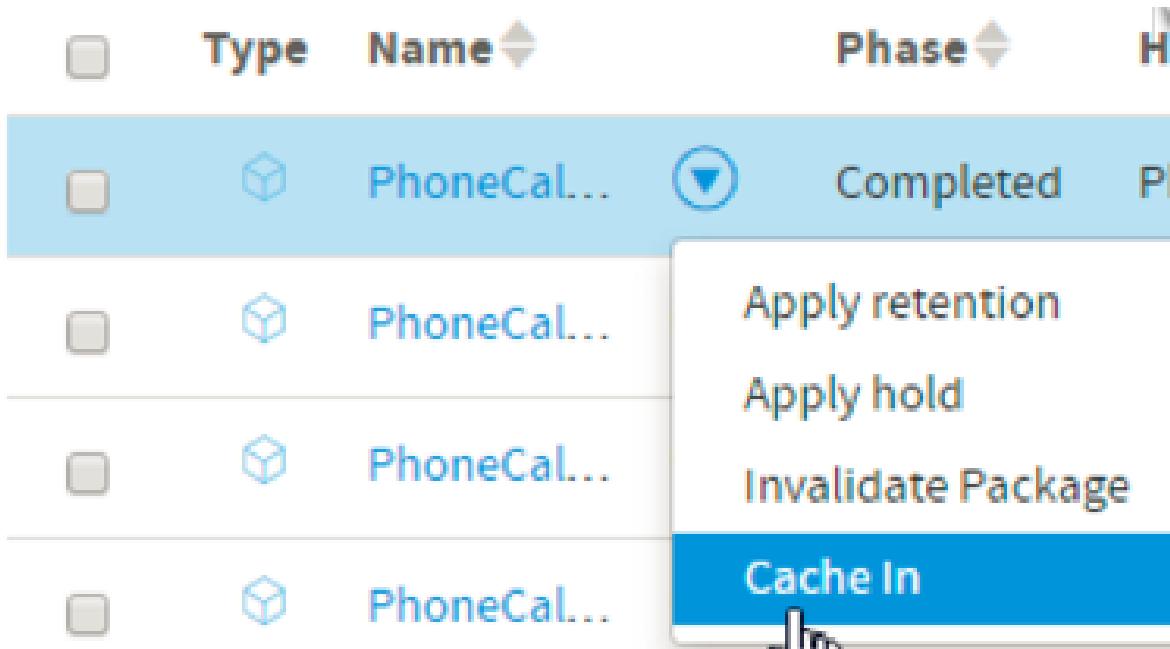
If needed, manual cache-in/cache-out requests can be made.

1. In IA Web App:
 - a. Select a SIP application.
 - b. Navigate to the **Packages** tab. A check mark in the Online column indicates that an AIP is cached-in. If a check mark does not appear in the Online column, the AIP is cached-out.
2. Click the button next to the name of a cached in AIP and select **Cache-Out** to create a cache-out request:



This creates a corresponding background request. Once this request is Complete, the AIP has been successfully cached-out.

You can also click on the button next to the name of a cached-out AIP and select **Cache-In** to create a cache-in request:



Reset Library Access Statistics

The Access Statistics of XdbLibraries can be reset, if needed.

This can be made by the following REST call at the SIP application level:

HTTP Method	Link Identifier	Verb	Example
PUT	http://identifiers.emc.com/reset-metadata-cache-stats	reset-metadata-cache-stats	http://localhost:8765/systemdata/applications/93e4aa25-b291-4959-8299-2676b9519d20/reset-metadata-cache-stats

Response Code: 200 OK

Result: Each cached-in xDB library of the application will now have its creation date as its last access date.

Frequently Asked Questions – Caching

How does system decide which AIPs and how many to cache-out?	<p>The AIP selection is based on the read access statistics.</p> <p>At the application level, you have a property <code>metadataCacheSize</code> to indicate how many metadata in bytes you want to keep in the metadata repository. By default, the value is 0 for unlimited.</p> <p>When the quota is exceeded, the system will try to remove from the cache the AIPs with the oldest last access dates until the cache quota is satisfied.</p>
How does system know that something is cached-out and it may have records from the search criteria?	<p>For each AIP, we have the information if the associated xDB library is online or offline.</p> <p>The search is executed in two steps. During the first step, we establish an AIP list that potentially matches the search criteria. To do that, we use the partition keys. During the second step, based on this list, we identify the xDB libraries where we want to execute the XQuery.</p>
Are there any data organization best practices for CICO utilization?	Today, the cache size is defined at the application level. If you want to apply different cache size per holding, you need to put the holdings into different applications.
How can I ensure that the system is not caching in too many AIPs and runs out of database disk space?	The cache size is a trend not a red line. The cache-out is performed by the Cache-Out job. If this job is never executed, or not often, the cache size can be higher than the expected value. The cache size is declarative, so the administrator must set a consistent value against the available space on the disk.

I know there is a manual capability to cache-out/cache-in AIPs. Is there job in place for such automation?	The Cache-Out job is available to do that automatically. The manual capability is more for demonstration purposes.
Has disposition taken the cache-out AIPs into consideration?	Yes. This is also the case for the confirmation, transformation, background searches.
Have storage dashboard calculations taken CICO into consideration?	Yes, but only the online xDB library is considered.

Note: You will not see retained sets for packages if the retention was applied automatically during ingestion. Packages ingested before upgrading to the new release will not be shown.

Configuring Back Up and Restore

Backing Up and Restoring the Managed Item Database

Managed items (for example, managed items, policy and hold applications) are stored in a separate database. As part of a disaster recovery scenario, they need to be backed up and kept in sync with the system data (retention policies, holds, etc.).

Unlike AIPs, managed items can change as policies are applied, holds are added and removed, so a different disaster recovery strategy is required.

If configured, as managed items change, InfoArchive will automatically back up the changes to a configured store.

Configuring the Managed Items Database

The managed items (HoldApplication, PolicyApplication, and ManagedItem objects) are partitioned over multiple xDBLibraries, with the AIP/table as the partition key. So, for example, all HoldApplications for AIUs in one AIP are together in a separate xDB library.

Depending on the expected size of the managed items database, you may want to move them to a different data node.

If you plan to use fine grained (AIU/record-level) retention, or expect a large number of holds on individual AIUs/records, then the managed items database may grow to even the size of the structured data. At that point, it is no longer feasible to back up the managed items as part of system data back ups. The back up will simply take too long. By mapping the managed items to a different data node and configuring stores for managed item back ups under Holdings/Databases, the back up process becomes:

- The managed items for each AIP/table are backed up separately, after each change (new or removed hold, new retention policy application, etc.).
- This happens immediately after the transaction where the changes took place.

Configuring the Managed Item database in the `application.yml` file. For example:

```
managedItemData:  
  xdb:  
    dataNode:  
      storeStackTraceInLock: false  
      name: mainFederation  
      bootstrap: xhive://localhost:8080  
      superuser:  
        password: test  
    database:  
      name: managedItemDatabase  
      admin:  
        password: secret
```

Backing Up the Managed Items Database

The managed items can be set to automatically back up when they are changed in InfoArchive. Setting the store for the managed items is controlled by application, and each application needs to have the managed item store configured for InfoArchive to back up managed items. By default, the managed item store is not configured. If the store is not set, InfoArchive will not back up the managed items.

The back up strategy for the managed items takes into account how you are backing up the system data and how you are applying retention and holds. The system data database is where the configuration objects (retention policies, holds) and the retained and hold sets are located.

No matter which method is being used to back up the managed items (InfoArchive or a scheduled back up), ensure the back ups to the system data and the managed item data should be completed at the same time.

InfoArchive uses a store for writing the managed items back up files. Ensure that there is sufficient space on the store and that the content can be removed (during disposition of the underlying managed item, hold application and policy application).

For more information, see [Adding a Store Using the Web Application](#).

Setting the Managed Item Store

For SIP archiving, you have two options when setting the managed item store:

1. Use IAShell to update the holding for the application; or
2. Use the REST interface to update the holding for the application.

The attribute on the holding object is called 'managedItemStore' and is a store object reference.

For table archiving, you have two options when setting the managed item store:

1. Use IAShell to update the database for the application; or
2. Use the REST interface to update the database for the application.

The attribute on the holding object is called 'managedItemStore' and is a store object reference.

XdbLibrary

For every managed item, hold application and policy application that is backed up, a corresponding XdbLibrary object is created in the system data database to represent the back up of the managed item. This object has a reference to the content object that was backed up in the managed item store. Since all managed items, hold applications and policy applications for a particular AIP/table are partitioned into a single xDB library, they are represented by an XdbLibrary object. For every partition, there can be three XdbLibrary objects representing the three types of objects in the managed item database (managed item, hold application and policy application).

The XdbLibrary object has a type defined for it called MANAGED_ITEMS. All managed item XdbLibrary objects are set to this type.

The XdbLibrary object uses the name of the object to uniquely identify it and is used to determine what partition (AIP/table) is associated with the managed item and what type of object it represents.

For a managed item:

```
ManagedItem_<partitionkey>
```

For a hold application:

```
HoldApplication_<partitionkey>
```

For a policy application:

```
PolicyApplication_<partitionkey>
```

Restoring the Managed Database

When restoring the managed item database, you must first decide what needs to be restored. Do only a few of the managed items need to be restored or do all the managed items need to be restored?

The method used to perform a restoration depends on the state of the managed item database. If the desire to restore the database back to a previous state due to a corrupted managed item database, use the xDbLibrary list to restore. If you wish to restore just a few managed items that, for whatever reason, you want to roll back, use the managed item list itself to restore.

Using the xDbLibrary

A REST client is required for the following procedure.

Identify which managed items you want to restore.

To restore the entire database to an empty managed item database (that you have recreated), complete the following procedure (a REST client is required for the following procedure):

1. For each application, search for all xDbLibraries that have the type = MANAGED_ITEMS.

2. For each xDbLibrary, select the restore link and execute a POST.

Complete the following steps to restore the entire database :

- To a managed item database that has the managed items still in the database, and
- You want to replace it with the back up

1. Under the Services link, search for all xDbLibraries that have the type = MANAGED_ITEMS.
2. For each xDbLibrary, select the detach link and execute a POST.
3. For each xDbLibrary, select the restore list and execute a POST.

For example:

```
http://localhost:8080/systemdata/applications/6b20bb40-671e-4fbe-a582-98526372283b/
xdb-libraries?type=MANAGED_ITEMS
```

You would get the following:

```
"_embedded": {
  "xdbLibraries": [
    {
      "id": "37061d55-4c2f-4fed-b74e-9b02fc6e4849",
      "name": "ManagedItem_48a2ea68-7f6b-4633-a44b-87d814bd3ddf",
      "type": "MANAGED_ITEMS",
      "size": 204800,
      "indexSize": 0,
      "version": 3,
      "lastModifiedBy": "sue@iacustomer.com",
      "lastModifiedDate": "2016-09-13T12:41:18.274-04:00",
      "createdBy": "sue@iacustomer.com",
      "createdDate": "2016-09-13T11:59:50.538-04:00",
      "detached": false,
      "readOnly": false,
      "detachable": false,
      "concurrent": false,
      "cacheSupport": true,
      "cacheInCount": 0,
      "aipCount": 0,
      "aiuCount": 0,
      "closed": false,
      "closeRequested": false,
      "xdbMode": "PRIVATE",
      "links": [
        {
          "self": {
            "href": "http://localhost:8080/systemdata/xdb-libraries/37061d55-4c2f-
4fed-b74e-9b02fc6e4849"
          }
        },
        {
          "request-close": {
            "href": "http://localhost:8765/systemdata/xdb-libraries/37061d55-
4c2f-4fed-b74e-9b02fc6e4849/request-close"
          }
        }
      ]
    }
  ]
}
```

-
}

To detach, call the following:

```
http://localhost:8080/systemdata/xdb-libraries/37061d55-4c2f-4fed-b74e-9b02fc6e4849/detach
```

To restore, call the following:

```
http://localhost:8080/systemdata/xdb-libraries/37061d55-4c2f-4fed-b74e-9b02fc6e4849/restore
```

Selected Managed Items

You have identified the selected managed items that need to be restored. The partition key to identify the xDbLibrary will be based on the AIP/table ID. You have to restore all managed items for a particular AIP or table. You cannot restore individual managed items (for example an AIU) under a particular AIP. The entire AIP's managed items would have to be restored.

Complete the following steps to complete the restoration:

1. Find the three xDbLibraries that correspond to the partition key. To find them, you would first have to get the AIP or table ID for the partition that you want to restore and issue the following commands to get the three xDbLibraries (for example, the partition key is 6b20bb40-671e-4fbe-a582-98526372283b):

a.

```
http://localhost:8080/systemdata/applications/6b20bb40-671e-4fbe-a582-98526372283b/xdb-libraries?name=ManagedItem_ 6b20bb40-671e-4fbe-a582-98526372283b
```

b.

```
http://localhost:8080/systemdata/applications/6b20bb40-671e-4fbe-a582-98526372283b/xdb-libraries?name=PolicyApplication_ 6b20bb40-671e-4fbe-a582-98526372283b
```

c.

```
http://localhost:8080/systemdata/applications/6b20bb40-671e-4fbe-a582-98526372283b/xdb-libraries?name=HoldApplication_ 6b20bb40-671e-4fbe-a582-98526372283b
```

2. Once you have the xDbLibrary object, issue the `detach` and `restore` commands:

a.

```
http://localhost:8080/systemdata/xdb-libraries/37061d55-4c2f-4fed-b74e-9b02fc6e4849/detach
```

b.

```
http://localhost:8080/systemdata/xdb-libraries/37061d55-  
4c2f-4fed-b74e-9b02fc6e4849/restore
```

Using the Managed Items

If the managed item list is available from an application, there are links off the managed item to detach and restore a managed item. The following example illustrates how to restore a managed item:

1. Get the list of managed items from an application.
2. Each of the managed items that you want to restore, the links will be on the managed item object:
 - a.

```
http://localhost:8080/systemdata/managed-items/8d2edb7a-01d0-4354-bf8a-  
65aa0e309073/detach
```

b.

```
http://localhost:8080/systemdata/managed-items/8d2edb7a-01d0-  
4354-bf8a-65aa0e309073/restore?xdbLibraryId=58cf9369-5d3b-  
4c2d-b2ad-272cde64ef4c
```

For each of the policy applications that is off the managed item, there are links off the policy application to detach and restore a managed item. The following example illustrates how to restore a managed item:

1. Get the list of policy applications from the managed item
 - a.

```
http://localhost:8080/systemdata/managed-items/8d2edb7a-01d0-4354-  
bf8a-65aa0e309073/retention-applications
```

2. Each of the managed items that you want to restore, the links will be on the managed item object:
 - a.

```
http://localhost:8080/systemdata/retention-applications/d9396a58-  
a2f9-456d-8268-c6af5d389fba/detach
```

b.

```
http://localhost:8080/systemdata/retention-applications/d9396a58-  
a2f9-456d-8268-c6af5d389fba/restore?xdbLibraryId=ddaea662-b79a-  
4055-bd0d-5e142440fbe3
```

For each of the hold applications that is off the managed item, there are links off the hold application to detach and restore a managed item. The following example illustrates how to restore a managed item:

1. Get the list of policy applications from the managed item:
 - a.

```
http://localhost:8080/systemdata/managed-items/8d2edb7a-01d0-4354-
bf8a-65aa0e309073/hold-applications
```

2. Each of the managed items that you want to restore, the links will be on the managed item object:

a.

```
http://localhost:8080/systemdata/hold-applications/d9396a58-a2f9-
456d-8268-c6af5d389fba/detach
```

b.

```
http://localhost:8080/systemdata/hold-applications/d9396a58-a2f9-
456d-8268-c6af5d389fba/restore?xdbLibraryId=ddaea662-b79a-4055-
bd0d-5e142440fbe3
```

To restore the managed items, identify what items will need to be restored. Currently, the only way to initiate the restoration is through the REST interfaces. The restoration of the system data at the same time needs to also be done to ensure that the configuration data and the retained and hold sets stay in sync with the managed item database.

If you want to manage the back ups separately from InfoArchive, do not configure the managed item store and back up the managed item and system data xDB databases separately.

Backing Up and Restoring Table Databases

This procedure only backs up the structured content and not the unstructured content.

Assuming you have the Baseball application installed, wait for the indexing service to rebuild the indices after ingestion.

1. Access IA Shell (refer to the *InfoArchive Shell Guide* for more information) and enter:

```
ia-shell> connect -user adam@iacustomer.com -psw password
Connected to "http://localhost:8765/services" as adam@iacustomer.com
iashell> cd applications/Baseball/databases/Baseball-sql-db
iashell> db-backup
OK
```

You will be waiting between 15 to 20 seconds. The operation is synchronous.

2. Verify that the back up was completed.

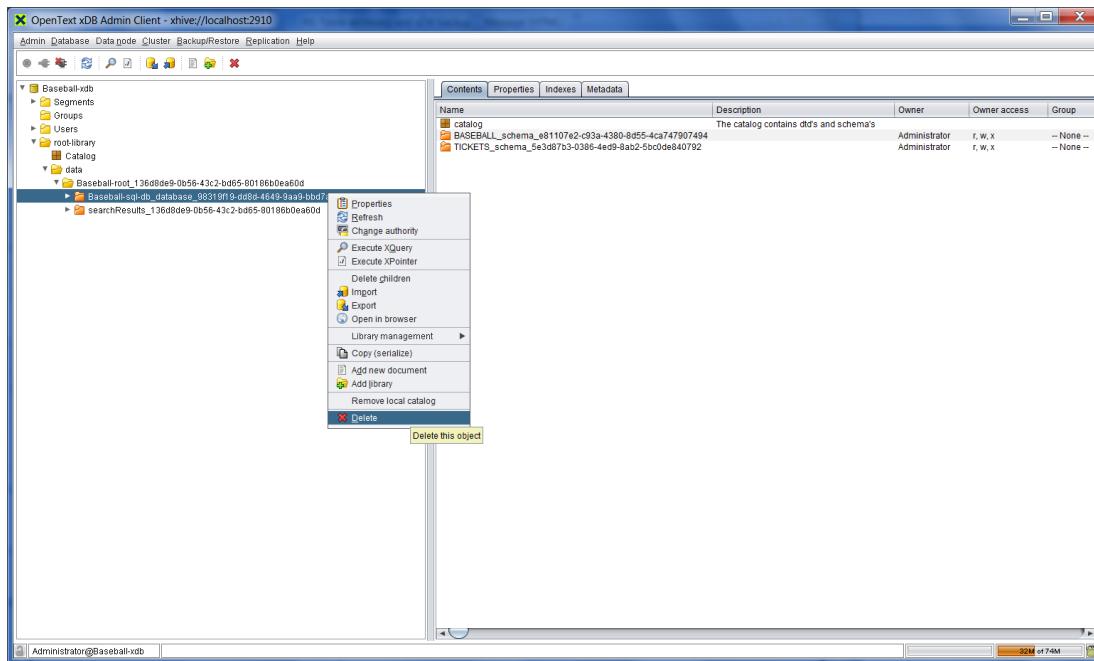
Use Windows Explorer to run a search in the store. If working correctly, you should see a gzip file that is for the database (one file).

3. Test the back up.

It is recommended that this step is completed in a staging environment.

- a. From xDB Admin, login to the database for Baseball-xdb.

Expand to root-directory node, find the `Baseball-sql-db_database` file and delete it:



This operation is confirmed.

- Searches will not work. Collapse and expand the node in xDB to prove that the the Baseball-sql-db_database file has been deleted.
- Restore the content with IA Shell. Access IA Shell and enter:


```
iashell> connect -user adam@iacustomer.com -psw password
      Connected to "http://localhost:8765/services" as adam@iacustomer.com
      iashell> cd applications/Baseball/databases/Baseball-sql-db
      iashell> db-restore
      OK
```
- Login as a Developer and do a search to verify that you get the results back for the search.

Configuring an Audit

InfoArchive allows the auditing of many events and on different levels. There are events related to the:

- system,
- tenant, as well as
- individual applications.

All audit-types that can be configured for an individual application are also available at the tenant level.

Audited events are stored in the xDB database and can be configured in the application.yml file for the server.

In IA Web App, Administrators can access the Administration > Audit tab to select which events are to be audited.

Audited events can be searched once the Audit sample application is installed and the Archive Audits job is executed. The job ingests events from the xDB database into the database associated with the Audit application. The Audit application applies a retention policy to events that it ingests, which allows audited information to be purged.

Turning on audits for retrieve events can impact performance, whenever a user gets a page of resources, an audit is generated for each item that was returned in the page.

Configuring Audit Events Using the Audit Tab

Being able to prove that certain actions have been performed is important, particularly when it comes to compliance. The **Audit** tab allows the Administrator or Developer to configure which events are audited. InfoArchive's auditing process allows the system to control the amount of content being generated.

Administrator permission is required to manage audits.

Audits for enabling or disabling events are now audited at the system level (versus the tenant).

The Application filter allows several levels of access to audit events for the selected application:

- System level: Contains a list of audit-events that correspond to services. Audits for changing events are audited at the system level.
- Tenant level: Contains a list of tenant-level audit-events.
- Application level: Contains a list of audit-events that correspond to particular application.

The following table outlines the general definition of fields used to toggle between an **Application** and an **Event Category**:

Name	Description	Notes
Audited Object	When applicable, indicates the ID of the object	Not always set, especially for login events.
Application	Name of the application	If possible, the name of the application. Not set for most tenant audits, such as retention policies.
Event Source	Name of the resource	Not always set. Set to the user name for a login event.

The Event Category filter allows an Administrator to determine which specific events are currently being audited. The filters depend on the level of access chosen. A check mark indicates that a particular audit event is enabled to generate an audit. Click a box to add or remove a check mark. The following table outlines the applications and the event categories that can be audited:

Application	Description
System	Provisioning Events Other
Tenant	Provisioning Events Compliance Events Ingestion Other
Customer-Created Applications	Provisioning Events Compliance Events Ingestion Other

Events are categorized in the following manner:

- **Provisioning Events**
- **Compliance Events**
- **Ingestion Events**
- **Other**

It is recommended that you configure system- and tenant-level audits that impact all applications first. If you want to only enable certain audits for a particular application, disable the audit at the tenant level and enable the audit for each application that you want the event enabled for.

If an audit is defined at the tenant- and application-level, both audits must be disabled, otherwise, the audit is generated.

The following event types are enabled by default. It is recommended that customers review all audits and choose which ones they want to enable:

Type	Name
aic	<ul style="list-style-type: none"> • create
aip	<ul style="list-style-type: none"> • commit • confirmation • confirmation completed • create • dispose • ingest • invalid • partial dispose • receive • reject • rollback
application	<ul style="list-style-type: none"> • create • dispose
audit events	<ul style="list-style-type: none"> • enable • disable
audits	<ul style="list-style-type: none"> • archive • disable • enable • purge
hold	<ul style="list-style-type: none"> • apply • remove
login	<ul style="list-style-type: none"> • failed • successful
purge list	<ul style="list-style-type: none"> • approve
retention policy	<ul style="list-style-type: none"> • apply
table	<ul style="list-style-type: none"> • dispose

Configuring System-Level Audits for Provisioning Events

Provisioning events are standard operations for managing resources. Most resources support provisioning events. Audits for changing events are now audited at the system level (versus the tenant).

Enable an audit based on one of the following actions to provisioning events at the system-level:

Action	Description
Create	Audit for when the resource is created.
Delete	Audit for when the resource is deleted. For items that are disposed, the dispose audit is generated instead of this audit.
Retrieve	Audit for when the resource is retrieved. This audit is often generated when getting the list of a resource.
	 Caution: Activating this audit may degrade performance.
Update	Audit for when the resource is modified. The name of the attributes changed in the resource are included in supplemental data.

The following are the system event types:

Event Type	Description
Crypto Object	Represents a crypto object, which can be referenced by applications.
Custom Storage	Refers to a custom storage mechanism.
Configuration	Represents when changes are made to configuration through declarative configuration. Configuration can be imported or exported. When importing a configuration, a start and end audit can be configured.
ECS Credential	If using ECS storage, set up an audit to be kept informed of changes made to the ECS Credential object, which is stored.
ECS Endpoint	Not currently used.
Federation	A federation refers to an xDB data node that can contain multiple xDB databases.
File System Root	Indicates a root location on a disk for file storage.
Job Definition	Configuration that defines a job include scheduling information, and parameters required for the job to function.
Job Instance	Represents a scheduled or past instance of a job.
LDAP Configuration	Stores LDAP configuration for managing groups.
Login	Indicates a login, which can be tracked through the gateway service. Includes operations done via IA Shell or through ant scripts.

Event Type	Description
Logout	Indicates a logout. This typically only is done via IA Web App.
Storage Endpoint	Configuration representing a storage endpoint. This configuration may refer to a URL (for example the URL of an S3 account).
Storage Endpoint Credential	Configuration storing credentials for a storage endpoint. This configuration stores credentials for accessing an S3 account.
Tenant	Configuration object that represents a customer. There are two tenants: InfoArchive and the System tenant. Tenants include applications.
xDB Cluster	Not currently used.
xDB Database	A database in xDB (not to be confused with databases for table applications).

Configuring System-Level Audits for Other Events

Audits for changing events are now audited at the system level (versus the tenant)

The following event types for the system support other events:

- Audit Events
- Configuration
- Job Definition
- Job Instance
- Login
- Logout

Apply an audit based on one of the following actions to other events at the system-level:

Configuring Tenant-and Application-Level Audits for Provisioning Events

Provisioning events are standard operations for managing resources. Apply an audit based on one of the following actions to provisioning events at the tenant- and application-level:

Action	Description
Create	Audit for when the resource is created.
Delete	Audit for when the resource is deleted. For items that are disposed, the dispose audit is generated instead of this audit.

Action	Description
Retrieve	Audit for when the resource is retrieved. This audit is often generated when getting the list of a resource. ⚠️ Caution: Activating this audit may degrade performance.
Update	Audit for when the resource is modified. The name of the attributes changed in the resource are included in supplemental data.

The following are the event types that you can apply a tenant- or application-level audit against. Most of these types support the provisioning audits event types, unless otherwise noted:

Event Type	Description
AIC	Specific to a SIP-based application, stores information for doing queries.
AIP	Specific to a SIP-based application, an AIP is a package that contains archived data. An AIP is a SIP that has been archived. AIPs contain AIUs.
AIU	Specific to a SIP-based application, an AIU is a record inside an AIP.
Application	An application is either SIP- or table-based, and stores information about that application. Only available as a tenant-level audit.
Application Category	Used to categorize applications. Only available as a tenant-level audit.
Audits	The archiving of an audit record.
Batch item	If an order item will act on a large number of items, batch items represent a chunk of work to do.
Bucket	Related to ECS, a bucket defines a location within an ECS store.
Collection	A collection refers to a search result and can be associated with a matter to protect records.
Configuration Helper	Specific to a SIP-based application, used for configuring queries.
Confirmation	Used to confirm various lifecycle transitions for the AIP.
Database	Representation of an archived (SQL) database. Databases contain one or more schemas.
Delivery Channel	Defines what to do when the search is run, the results of a search are sent to a delivery channel.
Encryption Server Configuration	Not currently used.
Event	Events can be used to determine the date to start aging and can also be updated by jobs.

Event Type	Description
File System Folder	Represents a file system folder for storing unstructured content.
Hold	Prevents disposition and deletion of items, even if the retention policy indicates that the items no longer need to remain. Only available as a tenant-level audit.
Holding	Specific to a SIP-based application, a holding governs how AIPs are ingested, and can provide retention instructions.
Holding Composition	Used by the holding wizard, it allows a developer to create a holding that is destroyed once the configuration has been completed.
Holding Crypto	Specific to a SIP-based application, defines the cryptography setting for the holding.
Ingest	Reserved for future use. Part of a SIP application.
Ingest Node	Specific to a SIP-based application, an ingest node is part of a holding.
Managed Item	Represents an item being managed for compliance.
Matter	A matter is typically associated with a legal hold, while collections are associated with matters to protect records. Only available as a tenant-level audit.
Order Item	Represents a background task, both jobs and user generated requests use this mechanism.
PDI	Specific to a SIP-based application, PDI is a OASIS term (preservation description information). Defines information for the indices for better query performance.
PDI Crypto	Specific to a SIP-based application, this defines cryptography settings for the PDI.
PDI Schemas	Specific to a SIP-based application, defines the fields that will be in the record.
Purge List	When items are eligible for disposition, a purge list groups related items into a list that can be approved.
Query	Defines the query for the search. For table-based applications, this is the XQuery that does the search.
Query Module	Part of a table application, an XQuery module can contain XQuery variable and function declarations. You can reference this XQuery module from an XQuery. This way, multiple XQueries can reuse module function and variable declarations.
Query Quotas	Specific to a SIP-based application, used to limit the maximum number of records returned in a query.
Receiver Nodes	Specific to a SIP-based application, an receiver node is part of a holding.

Event Type	Description
Result	A result is the stored query results. Results can be exported.
Result Master	Defines the columns to be returned for searches.
Retention Policy	Defines a policy for how long to keep items. Only available as a tenant-level audit.
Rule	A rule is used to apply retention, apply holds, or trigger events. Rules can be triggered upon ingestion or via jobs.
Schema	Specific to a table-based application, a schema contains a set of tables.
Search	A search contains the components necessary to define a search. A search contains a result master, XForm, and query.
Search Group	Part of search.
Space	Represents all storage from an application, in xDB (structured data), and on file systems (ECS, S3, etc (unstructured data).
Space Root Folder	A space root folder ties a file system root and space.
Space Root Object	Part of a space that defines the credentials and storage information.
Space Root XDB library	A space root xDB library defines the xDB library for storing structured content.
Store	A store defines a location to store unstructured content.
Table	Specific to a table-based application, a table is grouped within a schema.
Table Row	Specific to a table-based application, a record within a table.
Transformation	Part of a SIP application, transformations allow you to convert from one schema to another using XSLT.
Type Alias	Used to define alias for use in IA Shell.
xDB Pool Library	Part of a SIP application, defines an xDB pool library specific to storing your application.
XForm	An XForm is part of search that defines the fields specified requested to narrow a search.

Configuring Tenant-and Application-Level Audits for Compliance Events

Refer to [What are Compliance Events?](#) to learn what actions you can apply tenant- and application-level audits against.

The following are the compliance event types that you can apply a tenant- or application-level audit against, unless otherwise noted:

Event Type	Description
AIP	Set up an audit to be kept informed whenever an AIP has been disposed or partially disposed.
AIU	Set up an audit to be kept informed whenever an AIP has been disposed.
Application	Set up an audit to be kept informed whenever an application has been disposed. Only available as a tenant-level audit.
Hold	Set up an audit to be kept informed whenever a hold has been applied or removed.
Matter	Set up an audit to be kept informed whenever a legal matter has been applied, opened, closed, or removed.
Purge List	Set up an audit to be kept informed whenever a purge candidates list has been generated, approved, cancelled, or revoked.
Retention Policy	Set up an audit to be kept informed whenever a retention policy has been applied, removed, or requalified.
Table	Set up an audit to be kept informed whenever a table has been disposed or partially disposed.
Table Row	Set up an audit to be kept informed whenever a table row has been disposed.

Configuring Tenant-and Application-Level Audits for Ingestion Events

The following is a list of the types of ingestion audits:

- AIPs
- Audits
- Purge List

What are Ingestion Events

Ingestion events are about ingesting content into the archive and tracking the state of the artifacts. Apply an audit based on one of the following actions to ingestion events at the tenant- and application-level. Both tenant- and application level event types include AIPs and purge lists. Tenant-level audits can also include the audit event type:

Name	Description
Archive	Audit for when audits are archived. Archiving is completed via the Archive Audit job. When the job is executed, the archived audits are no longer available through the audit REST APIs, but are available using the search templates in the Audit application. Only available as a tenant-level audit.
Change Retention	Audit for when retention is changed. Retention can be changed by either adding a new retention policy or via re-qualification.
CI Content	Audit when CI content is downloaded.
Commit	Audit for when the AIP is committed. This is the final part of ingestion when the AIP has been confirmed to be valid and any retention required, based on the retention class or default retention policy on the application, is applied.
Confirmation	Audit of when confirmation is done. This action is done via the Confirmation job.
Confirmation Completed	Audit for when the confirmation is completed.
Delete	Audit for when the AIP is deleted. This can only be done through REST if no retention is applied to the package (or any records), and the application is still in test mode. This audit is not executed if disposition is complete.
Delete Content	Audit for when the AIP is deleted. This can only be done through REST if no retention is applied to the package (or any records) and the application is still in test mode. As mentioned earlier, this audit is not done if disposition is completed.
Download Content	Audit for when content of an AIP is downloaded.
Export	Audit for export. Only supported for purge lists and configuration. For this audit, the following values can be set: <ul style="list-style-type: none">• Key: Scope Name of the parent that the unstructured content was viewed from. Possible values are Tenant, Application, Holding, or Search.• Key: Name Name of the exported configuration object.
Get Content	Audit for when the content of an AIP is downloaded.
Ingest	Audit for the ingestion of an AIP.
Invalid	Audit for if an AIP is marked as invalid. An AIP is invalidated when the wrong SIP was submitted and you want to resubmit the correct SIP with the same identifier. The audit is generated before the job processes it.

Name	Description
Purge	Audit for when audits are purged. This audit is only for audits, and is completed whenever the Archive Audits job is executed. Only available as a tenant-level audit.
Rebuild	Audit for when an AIP is rebuilt.
Receive	Audit for when an AIP is received.
Reject	Audit for when an AIP is rejected. An AIP is rejected if all AIPs that belong to the same collection are invalidated. When you reject an AIP, you cannot resubmit an AIP with the same DSS as long there is one or more rejected AIPs in the repository. The audit is generated before the job processes it.
Rollback	Audit for when an AIP is rolled back.

Configuring Tenant-and Application-Level Audits for Other Events

The following is a list of the Other Events:

- AIC
- AIP
- AIU
- Batch Item
- Order Item
- Purge List
- Query
- Rule
- Search
- Table
- Table Row

What are the Other Events

Apply an audit based on one of the following actions to other events at the tenant- and application-level, unless otherwise noted:

Action	Description
Active	Audit for when a job definition is activated. This event happens when a suspending job definition is resumed
Brava_download	Audit for when content is downloaded in the Brava! viewer.

Action	Description
Brava_print	Audit for when content is printed in the Brava! viewer.
Brava_view	Audit for when content is viewed in the Brava! viewer. This is distinct from the preview action, supported for AIPs and tables.
Disable	Audit for disabling. Currently, only for audits, specifically for disabling an audit event type.
Enable	Audit for enabling. Currently, only for audits, specifically for enabling an audit event type.
End	Audit for when processing has been ended for an operation, and applies to the following objects: configuration, job_instance, and order_item. This audit does not happen if an error occurred during processing.
Failed	Audit for a failed login attempt. Currently, only for the login.
Fetch	Audit for when the resource is fetched. Currently, only AICs, AIUs and table rows support this event.
Inactive	Audit for when a job definition is suspended.
Native_view	Audit for when content is viewed using a native viewer. This is specific to unstructured data viewed from a set of search results.
Preview	Audit for when unstructured content is previewed.
Run	Audit for when a job definition is executed. This audit records information for scheduled jobs or jobs that run manually.
Search	Audit for when a search is done against the resource. This is supported for AICs and queries.
Skip	Audit for when a job is skipped.
Start	Audit for when processing has been started for an operation, and applies to the following objects: configuration, job_instance, and order_item. This audit may not be generated if an error occurred during processing. For example, for job instances, even if the job fails, the start and end audits are created. For declarative configuration, however, if the operation is rolled back, the start audit is not created.
Successful	Audit for when a user either logs in or logs out. Audits can be configured independently for login and logout
Unskip	Audit for un-skipping.
View Log	Audit when the diagnostic logs are viewed. Available for job instances, order items and batch items.

Supplemental Data

Action	Description
Updated Fields	Names of the fields that were updated. On an update, only values that can be changed, and that were changed from the previous value, are updated.
Name	Name of the object that was changed. Often also set as the event source for the audit.

Testing that Audit Changes Work

The following example uses the PhoneCalls application to demonstrate how to configure an audit for a search.

1. In the **Administration > Audit** tab, select the PhoneCalls application.
2. In the Event Category drop-down list, select **Other**.
3. For the AIC event type, check the Search box and click **Save**. This enables an AIC search.
Note: For the table version, the audit is against the query type instead of an AIC.
4. Return to the **Applications** tab and select the PhoneCalls application.
5. In the **Record Search** tab, execute the FirstName_Operator search.
6. Navigate to the **Administration > Jobs** tab. Execute the Archive Audits job.
Unless you archive the audits, you cannot search an audit.

You are able to execute searches on the AIC type.

When looking at the supplemental data, you are able to view the name of the AIC and the search criteria that was used.

Audit Troubleshooting

Issue	Resolution
I have turned off an audit for an event type at the application level but the audit is still being generated.	Check to see if the audit is enabled at the tenant level. If it is, disable it at the tenant level and enable the audit for applications that still want that audit event type to be generated. The setting to disable an audit for the application is ignored if set at the tenant level.
When I search the audits, I do not get any results.	Run the Archive Audits job periodically so that the audits are archived.

Language Support

InfoArchive has the ability to dynamically add a new language translation for IA Web App.

The IA Web App login page allows users to select a language from the drop-down list with seven language choices. The supplemental language pack installs all primary languages. For the supported language to work, a corresponding <locale_name>.json file should be populated into the classpath. The classpath is automatically modified by supplemental language pack installation.

InfoArchive provides ease of configuration for new language translation.

The language support is improved to be more dynamic. In the new approach:

1. The languages drop-down list values are decided at run-time based on the presence of supporting language files.
2. The user can configure more languages by following the [configuration](#) steps.

Also, the language drop-down list control on the login page shows only those languages for which there are corresponding <locale_name>.json files either in classpath or in the [configurable](#) location. It gives a precise picture of languages supported and avoids confusion when a user selects a language for which a corresponding <locale_name>.json is not present.

Adding New Language Support

Complete the following steps to facilitate support for a new language translation other than the seven primary languages.

To add support for a new language, the corresponding <locale_name>.json file should be generated first. Customers should use the en.json file from lib > infoarchive-webapp.jar under WEB-INF > classes > static > languages folder as a reference for translation into new language.

The following example illustrates how to add support for the Dutch language.

1. Create a folder named customization under the <INFOARCHIVE_ROOT>/examples/legacy-ant-tenants/infoarchive folder in the distribution.
2. Create a folder named languages in the customization folder that you created in the previous step.
3. Create a file named languages.json in the languages folder that you created in the previous step.
4. Edit the languages.json file and add the following content in the file:

```
{  
    "nl": "Dutch"  
}
```

5. Drop the nl.json file (equivalent of en.json, zh.json for Dutch language) into the customization > languages folder.
6. Refresh the login page (using Ctrl + R or F5). There will be an option to select the Dutch language in the languages drop-down menu.

If you want to add support for the Russian language, complete the following steps:

1. Add an entry for the Russian language in the `languages.json` file:

```
{  
  "nl": "Dutch",  
  "ru": "Russian"  
}
```

2. Drop the `ru.json` file into the `customization > languages` folder.
3. Refresh the login page (using Ctrl + R or F5). There will be an option to select the Russian language in the languages drop-down menu.

For more information, see [Setting the Customization Location when Deploying to Tomcat or Other Containers](#).

Customizing Branding

InfoArchive supports limited, drop-in branding customization. Customers are able to define the view and display of the font, color, images and styling of IA Web App.

Configuration to Add Sample Branding Customization

A sample branding customization is provided to the customer in the following directory:

`<INFOARCHIVE_ROOT>/examples/legacy-ant-tenants/infoarchive/customization
/branding.`

To review the sample branding, copy the customization folder and paste it into the following directory: `<INFOARCHIVE_ROOT>/config/iawebapp`.

Configuration to Add New Branding Customization

To replace the image file of IA Web App, complete the following steps:

1. Open the following directory in Windows Explorer: `<INFOARCHIVE_ROOT>/examples
/legacy-ant-tenants/infoarchive/branding/images`.
2. Review the image requirements and directions in the `README` file.
3. Copy and replace the image file to the folder. Do not change the name of the image file.

To replace the styling and CSS rule of IA Web App, complete the following steps:

1. Open the following directory in Windows Explorer: `<INFOARCHIVE_ROOT>/examples
/legacy-ant-tenants/infoarchive/branding/css`.
2. Review the information about current CSS rules in the `README` file.
3. Open the `custom.css` file with a text editor and edit the styling rules using CSS syntax.

Setting the Customization Location when Deploying to Tomcat or Other Containers

The previously described steps work when IA Web App is run as a standalone Springboot application using the infoarchive > bin > infoarchive-webapp command. When it is deployed to an external Tomcat container, however, the situation is different.

For example, assuming the external Tomcat container is installed at:

```
C:\apache-tomcat-8.0.32
```

And IA Web App .war file is deployed at:

```
C:\apache-tomcat-8.0.32\webapps\infoarchive-webapp.war
```

When deployed, the Tomcat expands the .war file to the following folder:

```
C:\apache-tomcat-8.0.32\webapps\infoarchive-webapp
```

Now, assume that you copied the customization folder to the following directory:

```
C:\apache-tomcat-8.0.32\webapps\infoarchive-webapp\config\iawebapp\customization
```

Then, one way to set the customization location is to edit the application.yml file located at:

C: \apache-tomcat-8.0.32\webapps\infoarchive-webapp\WEB-INF\classes\application.yml
and change the key:

```
:
infoarchive:
  gateway:
    :
  customization:
    location: "file:///C:/apache-tomcat-8.0.32/webapps/infoarchive-webapp/
config/iawebapp/customization"
:
```

Alternately, another method to set the customization location is by creating a web.xml file at:

```
C:\apache-tomcat-8.0.32\webapps\infoarchive-webapp\WEB-INF\web.xml
```

Set the contents to:

```
<web-app version="3.0"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">
<context-param>
  <description>INfoarchive Customization location</description>
  <param-name>infoarchive.webapp.customization.location</param-name>
  <param-value>file:///C:/apache-tomcat-8.0.32/webapps/infoarchive-
  webapp/config/iawebapp/customization/</param-value>
</context-param>
</web-app>
```

Verifying and Viewing the Branding Customization

To verify if the web customization, complete the following steps:

1. Open IA Web App in an Internet browser.
2. Refresh and reload the web page.
3. The customized branding style should be displayed.

Troubleshooting

If the new customization is not displayed, clear the browser cache.

InfoArchive JDBC Driver

InfoArchive provides the ability to retrieve table-based archived data via SQL queries. The bundled JDBC driver can be configured with any open source SQL client (*i.e.*, Squirrel SQL).

SQL Support

For table archiving, the SQL JDBC Driver provides ANSI-standard read-only SQL-based reporting access to relational table data stored as XML in InfoArchive/xDB databases.

JDBC driver features include:

- Standard result set processing (forward processing only).
- Querying database metadata for schemas, tables, columns, and available supported SQL functions.

Supported SQL Functions

The JDBC driver supports the following SQL functions with the same arguments as standard SQL:

- abs avg
- ceil ceiling concat count curdate current_date current_timestamp
- floor
- lcase lower
- max min
- now nvl
- replace round
- substr substring sum

- to_date to_timestamp
- ucase upper

Supported SQL Features

The SQL JDBC Driver supports most of the common SQL “select” query patterns to read archived table data from xDB:

Supported SQL Feature	Example
Direct column selection	select a,b from c
Column Aliases	select a as b from c
Like	select a from b where a like ‘c%’
In	select a from b where a in ('1', '2')
between	select a from b where c is between d AND e
is null	select a from b where a is null
order by	select a from b order by a
not (can be combined with not, in, between, is null, like)	select a from b where a not in ('1', '2')
All columns (select *)	select * from b
Table name alias	select a.b from c a
Theta Style inner join	select ta.b,tc.d from ta,tc where ta.b = ta.d
Standard Math operations	select a from b where c*2 > 5 select a*2 from b Note: Standard math syntax is supported.
Schema Selection	select a from schema1.table1 Note: If no schema is defined, then default schema will be used as specified in database metadata.
ANSII Style inner join	select ta.b,tc.d from ta join tc where ta.b = ta.d
ANSII Style left outer join	select a from b left outer join c on b.d = c.d
ANSII Style right outer join	select a from b
Group by	select a, count(*) from b group by a
Aggregate functions	select count(*),max(*),min(*),max(*),sum(*)
Fetching first n rows	select a from b fetch first 2 rows only
Fetching rows starting at specific point in query	select a from b offset 2 fetch first 5 rows only

Supported SQL Feature	Example
Case insensitive column/table selection	<pre>select AnyCaseColumn from AnyCaseTable</pre> <p>Note: Specific column/table will be determined at runtime based on metadata.</p>
select on in clause	<pre>select a from b where c in (select d from e)</pre>
union all	<pre>select a from b union all select c from d</pre> <p>Note: Union is not supported.</p>
distinct	<pre>select distinct a from b select distinct a,b from c select distinct * from a select a, count(distinct b) from c group by a</pre>
having clause	<pre>select a, count(*) from b group by a having count(*) > 2</pre>
Column names referencing tables is not required	<pre>Select a, b from c join d on c.key=d.key</pre> <p>Note: If the column name appears in both tables, you must reference a specific table.</p>
column alias use in group by, order by, and having clause	<pre>select a c, d from d group by c order by c having c > 2</pre>
sub select keywords – any/all/some	<pre>SELECT * FROM AlbumSales WHERE album_gross > ALL (SELECT album_costs FROM AlbumProduction);</pre>
exist	<pre>SELECT * FROM suppliers WHERE EXISTS (select * from orders where suppliers.supplier_id = orders.supplier _id);</pre>
exist	<pre>SELECT * FROM suppliers WHERE EXISTS (select * from orders where suppliers.supplier_id = orders.supplier _id);</pre>
Select from other select	<pre>select a,b from (select a,b from c where c > 2)</pre>

Supported SQL Feature	Example
Select on column selection	select a, (select b from c) from d
left/right outer join	<p>select a from b join c on a.b = c.d right join d on a.c = a.d</p> <p>Restrictions: do not use * or *.table, distinct, grouping and aggregate functions, subsequent columns with duplicated names (all joining tables together) are omitted</p>

Unsupported SQL Features

Unsupported SQL Feature	Example
Updating existing content	update ...
Inserting new content	insert into ...
Full outer join	select a from b full outer join c on b.d = c.d
Natural join	select a from b natural join c
Multiple selections	only a single statement will be processed select a,b from c; select c,d from e is not valid
union	<p>select a from b union select c from d</p> <p>Note: Union all is supported, which is all rows from the joined select clauses.</p>
operator to join columns	<p>select a b from c This can be done using the concat function</p>
minus	<p>select a from b minus select c from d</p>
intersect	<p>select a from b intersect select c from d</p>

SQL Type and XQuery Translation

InfoArchive translates SQL statements into XQuery. It is important to understand the SQL data type and its XQuery translation detail (see the following table):

SQL Type	JDBC Representation	XQuery Translation
BOOLEAN	BOOLEAN	boolean
BIT	BOOLEAN	boolean
CHAR	NVARCHAR	string
VARCHAR	NVARCHAR	string
VARCHAR2	NVARCHAR	string
NVARCHAR	NVARCHAR	string
TINYINT	INTEGER	integer
SMALLINT	INTEGER	integer
INT	INTEGER	integer
INT4	INTEGER	integer
INTEGER	INTEGER	integer
BIGINT	BIGINT	long
FLOAT	DOUBLE	double (*)
REAL	DOUBLE	double (*)
DOUBLE	DOUBLE	double (*)
NUMERIC	DECIMAL	decimal
DECIMAL	DECIMAL	decimal
TIMESTAMP	TIMESTAMP	dateTime
DATETIME	TIMESTAMP	dateTime
DATE	DATE	date
TIME	TIME	time

(*) XQuery has its own rules for casting numbers to data types. For instance, when a number contains an 'E' or 'e', it is automatically converted to xs:double. If it contains a '.' but not an 'E' or 'e', it is converted to an xs:decimal. The xDB XQuery optimizer will try to convert the number to the corresponding index data type. For SQL types FLOAT, REAL and DOUBLE, this is DOUBLE, for SQL types NUMERIC and DECIMAL, this is DECIMAL.

The translation aspect should be considered during the extraction process so the ETL tools can rightfully capture such details for the table-based ingestion process. The data types detail needs to be captured in the `metadata.xml` file required for table ingestion.

Connection Setup

The Java Database Connectivity (JDBC) driver authentication is done using a user name and password.

1. Specify the following properties:

```
user=user-name
```

```

password=user-password
clientId=infoarchive.jdbc
clientSecret=infoarchive.jdbc-client-secret (it is configured in the
InfoArchive web application)

```

2. Using JDBC API, specify the following properties:

```

com.emc.ia.sql2xquery.jdbc.JdbcDriver:
final JdbcDriver jdbcDriver = new JdbcDriver();
DriverManager.registerDriver(jdbcDriver);
final Properties info = new Properties();
info.setProperty("user", "user-name");
info.setProperty("password", "user-password");
info.setProperty("clientId", "infoarchive.jdbc");
info.setProperty("clientSecret", "secret");
final Connection connection = DriverManager.getConnection(connectionString, info);

```

3. Using SQuirreL SQL Client: Modify the selected alias | Properties | Driver properties.

4. Appending to a connection string:

```

jdbc:ia://localhost:8080/restapi?tenant=INFOARCHIVE&application=
Tickets&database=Tickets-sql-db&user=connie@iacustomer.com&password=
password&clientId=infoarchive.jdbc&clientSecret=secret

```

The authEndpoint property points to an authentication end-point host[:port]/path/to/authentication /endpoint.

For example, in the case of the JDBC connection string:

```

jdbc:ia:///hostname:8080/infoarchive-webapp/restapi?tenant=
INFOARCHIVE&application=Baseball&database=Baseball-sql-db
'authEndpoint'=localhost:8080/infoarchive-webapp/oauth/token.

```

The following is an example of the procedure:

After you have ingested sample data in the InfoArchive repository, you need to set up the JDBC (Java Database Connectivity) driver.

InfoArchive supports the Oracle Java Development Kit (JDK).

The following instructions use the Baseball data as an example. Squirrel is one method used to connect. It is not required:

1. Download the SQL Client Squirrel from <http://squirrel-sql.sourceforge.net/> and install it.
2. Click the **Drivers** tab, then click the **Create New Driver** (+) icon.
3. In the **Name** field, type `sql2xq-jdbc-driver` The JDBC driver can be found in the `lib` folder of IA Server distribution
4. In the **Example URL** field, type `jdbc:ia://localhost:8080?tenant=INFOARCHIVE&application=Baseball&database=Baseball-sql-db`
5. Click the **Extra Class Path** tab, then click the **Add** button.
6. Select the JDBC driver (`sql2xq-jdbc-driver.jar`) in the `lib` folder of IA Server distribution.
7. Click **List Drivers**.

8. Click **OK**.
9. Click the **Aliases** tab, then click the **Create New Alias** (+) icon.
10. In the **Name** field, type **infoarchive-baseball**.
11. Select **sql2xq-jdbc-driver** from the **Driver** drop-down list.
12. Click the **Auto logon** checkbox.
13. Click the **Properties** button, then click the **Driver properties** tab.
14. Click the **Use driver properties** checkbox, then click the **Specify** checkbox and add the properties you want to add, if desired.
15. Click **OK**, then double-click the **infoarchive-baseball** alias to open a connection.

Configuring OpenText Directory Services

InfoArchive uses the OAuth 2 authorization framework for authorization. It delegates the authentication to a limited set of external authentication mechanisms, Lightweight Directory Access Protocol (LDAP) or Active Directory (AD). It uses JSON Web Token (JWT) for a stateless authentication context. This is implemented in the Gateway component of InfoArchive.

OpenText Directory Services (OTDS) is a repository of user and group identity information and a collection of services to manage this information for OpenText components. OTDS manages the integration of many authentication systems, such as single sign-on (SSO). You must prime OTDS to be used with InfoArchive. This section illustrates how to further prepare it for use with InfoArchive.

Use the OTDS Administration web client to manage the following components:

Component	Description
Resource	Resources represent multi-user systems, or components, that users can access. Essentially, a resource is a representation of such a component in OTDS. For example, throughout this section, <code>infoArchive</code> is used as the resource name to represent the InfoArchive instance.
Access Role	Access roles are used to control which resources users can access. A default access role is for a resource that is automatically created. For example, the <code>infoarchive</code> resource has the access role <code>infoarchive</code> . Any partitions that are members of the <code>infoarchive</code> access role will be able to access the <code>infoarchive</code> resource. This allows mapping for all the groups defined in those partitions to be mapped to the InfoArchive roles.
Partition	<p>Partitions are self-contained copies of user information that allow you to organize users into a structured hierarchy of users, groups and organizational units. A user partition in OTDS is represented by a unique name. Content can be imported and synchronized with Active Directory (AD) and/or Lightweight Directory Access Protocol (LDAP), and can be managed fully within OTDS. OTDS supports multiple, concurrent user partitions.</p> <ul style="list-style-type: none"> • Synchronized User Partition: Partitions are synchronized with an identity provider, such as AD or LDAP. A synchronized user partition contains users, groups and organizational units that are imported from the identity provider when the user partition is created. A synchronized user partition can be automatically kept up-to-date with its source directory. Users who are imported from an identity provider into a synchronized user partition are authenticated by the identity provider. • Non-synchronized User Partition: These are created and maintained manually. Unlike a synchronized user partition, a non-synchronized user partition does not have an identity provider from which its users and groups are imported. Users and groups in a non-synchronized user partition are maintained entirely through the OTDS Web Client. Users who are created and maintained manually in a non-synchronized user partition are authenticated by Directory Services. Configurable password policies are available for non-synchronized user partitions. For example, InfoArchive may optionally provide a way to create a non-synchronized partition that has some out-of-box users and groups similar to the <code>OOB_IN_MEMORY</code> users we have today.

When configuring OTDS for use with InfoArchive, you have the option of using a simple or full integration. A simple integration is also referred to as the authentication provider mode. A full integration is also referred to as **Single Sign On** (SSO) mode. Of course, the primary difference between the two types of integration is that the full integration allows the system to include SSO functionality.

Refer to [Using OTDS in Authentication Provider Mode](#) and [Using OTDS in SSO Mode](#) for further information.

Downloading and Installing OTDS

OTDS only needs to be downloaded if you are not currently using any other OpenText components. If you already have other OpenText components that are integrated with OTDS, you must prime configure OTDS to be used with InfoArchive.

Download OTDS from <https://knowledge.opentext.com/knowledge/cs.dll/Open/OTDS>.

For Windows, the installer is a .msi file. During the installation, set the JDK and Tomcat locations.

For Linux, the installer is a .tar file. During the installation, set the JDK and Tomcat locations.

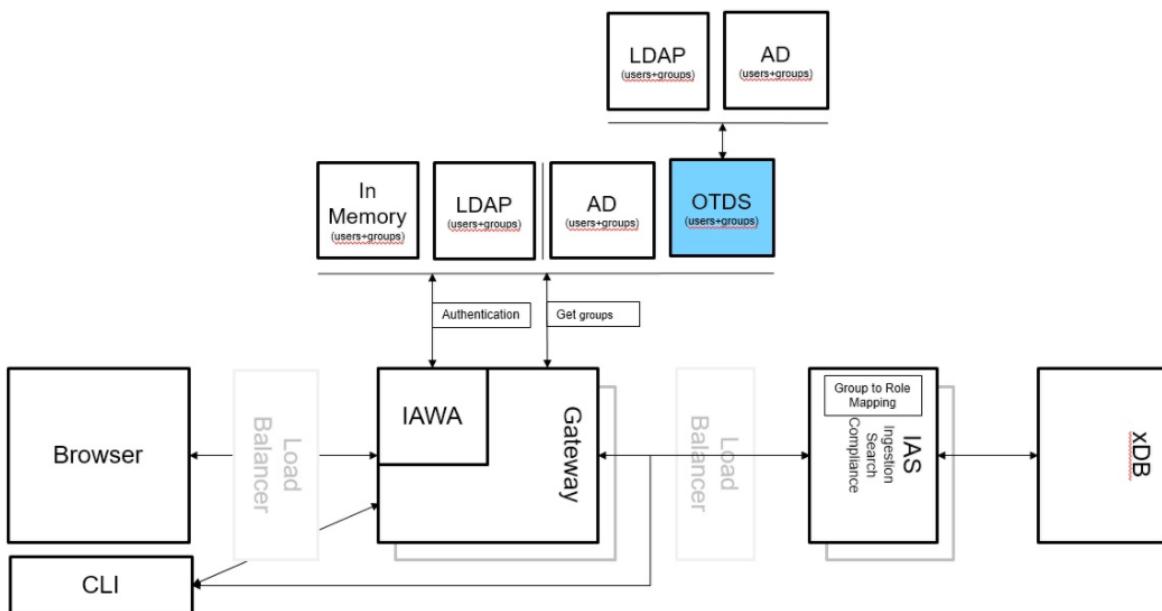
Prior to installing OTDS, be sure to review the OTDS [Installation and Administration Guide](#), which contains the installation prerequisites, as well as important configuration information.

Once installed, apply the following patch on top of the OTDS installation: <http://mimage.opentext.com/support/ecm/secure/patches/otds/16.2/otds-1622-hotfix001.zip>

Using OTDS in Authentication Provider Mode

This mode allows for authentication using user name- password-based authentication. OTDS acts the same as other authentication providers, such as in-memory user accounts, Lightweight Directory Access Protocol (LDAP) or Active Directory (AD). Because OTDS allows configuration of many Identity Providers (IDPs), however, it is possible to use multiple IDPs via OTDS, which opens up lot more possibilities. In the authentication provider mode, InfoArchive Gateway is still in charge of issuing the JWT tokens that are used for the stateless access.

The following diagram illustrates the architecture of OTDS integration when using OTDS in the authentication provider mode:



InfoArchive uses OTDS as an external authentication system in the same manner it uses LDAP and AD. The use of OTDS will be activated using a Spring Profile. Once activated, the users defined in

OTDS who are able to access the resource associated with the InfoArchive instance will be able to log in. Prior to any user logging in, however, it is important that the groups users belong to should be mapped to their respective InfoArchive roles via the Administration > Groups tab in the web application. The InfoArchive Gateway will query OTDS for groups from all partitions that are members of the access role associated with the InfoArchive resource and make them available for mapping on the Administration > Groups tab.

Configuring OTDS for Authentication Provider Mode

This section illustrates how to prepare it for use with InfoArchive.

This section assumes that OTDS is deployed to Tomcat running at port 8090 on localhost. Therefore, the OTDS administration URL will be `localhost:8090/otds-admin/`.

The Rest API endpoint is `localhost:8090/otdsws/rest`.

The OAuth2 authentication endpoints require that the HTTPS protocol is used. This section assumes that the Tomcat has been configured to run HTTPS connector at port 8443, and the appropriate keys and certificates have been configured.

To configure OTDS as the Authentication Provider, complete the following:

1. Start OTDS if it is not already started.
2. Normally, you would access the OTDS Administration web client to create the InfoArchive resource, access role, partitions and OAuth2 Clients. Instead, use the OTDS InfoArchive Initializer Utility, which is located in the `<INFOARCHIVE_ROOT>/bin` directory, to easily create the InfoArchive resource, access role, partitions and OAuth2 Clients:

Note: The configuration of OTDS connectivity are located in file `<INFOARCHIVE_ROOT>/config/otds-ia-init/application-infoarchive.otds.initializer.profile.OTDS.yml` file.

- For Windows, enter `enterotds-ia-init.bat`.
- For Linux, enter `otds-ia-init`.

Note: You can set the environment variable `OTDS_IA_INIT_OPTS` prior to running the `otds-ia-init` utility to pass parameters and system properties to the JVM. For example, use the following to pass the system properties to configure OTDSInfoArchiveInitializer to talk to OTDS over HTTPS:

```
OTDS_IA_INIT_OPTS=-Djavax.net.ssl.trustStore=.../config/
webapp/gatewayTrustStore.jks -Djavax.net.ssl.trustStorePassword=
abcdefg -Djavax.net.ssl.trustStoreType=JKS
```

At this point OTDS is primed for use with InfoArchive. Proceed to activate the OTDS integration.

At this point you can enable the `infoarchive.bootstrap` partiotion in OTDS using the OTDS Admin Web Client. You may also configure additional partitions for your IDPs - LDAP or Active Directory. Refer to OTDS documentation to learn how to complete these tasks. Make sure to add those partitions to the access role called `infoarchive`.

Note: The OAuth2 Clients can be deleted or not created in the first place, as they are only used in the SSO mode.

To activate the OTDS integration:

InfoArchive Gateway

1. Configure the active profiles by updating the active property in the config/iawebapp/application.yml file, as illustrated below:

```
spring:  
  application:  
    name: infoarchive.gateway  
  profiles:  
    # The infoarchive.gateway.profile.AUTHENTICATION_IN_MEMORY is not  
    # recommended for production.  
    # include:  
    - infoarchive.gateway.profile.AUTHENTICATION_IN_MEMORY  
    - infoarchive.gateway.profile.AUTHENTICATION_OTDS  
    :  
      :
```

This tells the system that you are opting to use OTDS in the authentication provider mode and activates the properties found in the <INFOARCHIVE_ROOT>/config/iawebapp/application-infoarchive.gateway.profile.AUTHENTICATION_IN_MEMORY.properties file.

2. Configure the access to OTDS in the <INFOARCHIVE_ROOT>/config/iawebapp/application-infoarchive.gateway.profile.AUTHENTICATION_OTDS.yml file, as illustrated below:

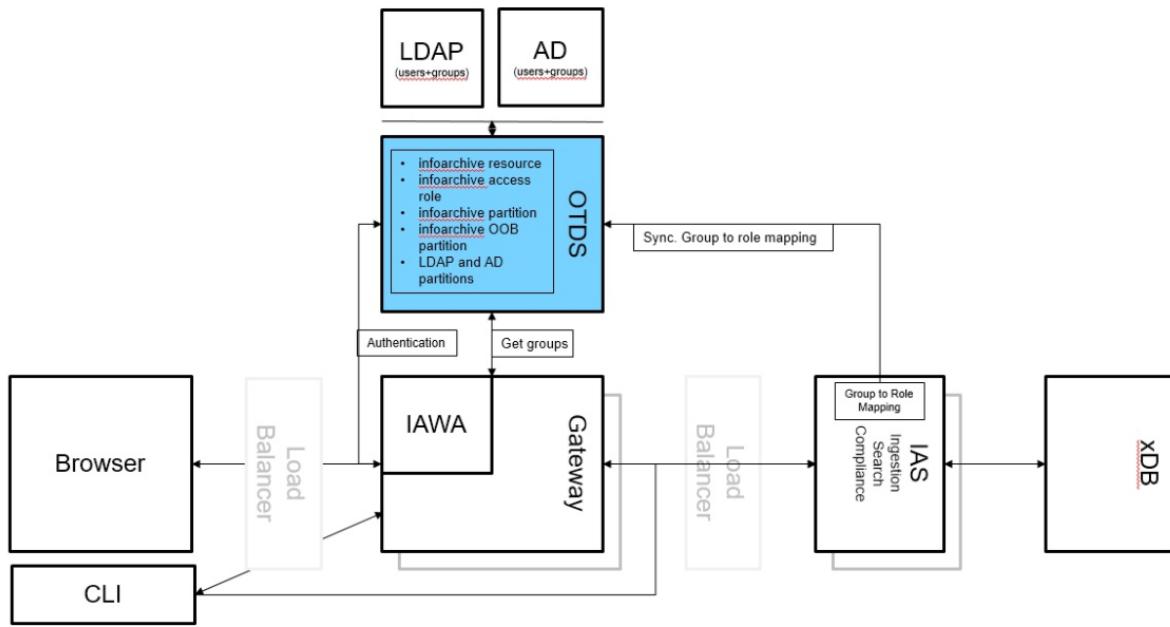
```
OTDS:  
  LOCATION:  
    host: localhost  
    port: 8090  
    httpsPort: 8443  
    path: /otdssws/rest  
    url: http://${OTDS.LOCATION.host}:${OTDS.LOCATION.port}/${OTDS.LOCATION.path}  
    # url: https://${OTDS.LOCATION.host}:${OTDS.LOCATION.httpsPort}/${OTDS.LOCATION.path}  
  username: otadmin@otds.admin  
  password: otadmin  
  infoarchive:  
    resource:  
      id: infoarchive # The name of the OTDS resource for this installation of InfoArchive
```

3. Sign in into the IA Web App with a user with the ADMINISTRATOR role.

Navigate to the Administration > Groups tab to map the groups discovered from OTDS to the appropriate roles.

Using OTDS in SSO Mode

In this exclusive mode, OTDS takes over the authentication and issuing of the JWT tokens completely. This enables OTDS to also provide SSO functionality.



Therefore, it is necessary to store the Group > Role mapping in OTDS. The InfoArchive roles are reflected as OTDS groups in a specially named, non-synchronized partition inside OTDS.

While it is possible to complete the mapping process using the OTDS Web Administration Client, it is much simpler to use the IA Web App, in which you map specific groups to the applicable user roles. This is because the IA Server synchs with OTDS to record any changes in group-role mapping. Refer to [Managing Groups](#) for further information.

Configuring OTSD for SSO Mode

This section assumes that the OTDS product has already been installed. This section illustrates how to further it to prepare it for use with InfoArchive.

To configure OTDS for SSO mode, complete the following:

1. Start OTDS if it is not already started.
2. Normally, you would access the OTDS Administration web client to create the InfoArchive resource, access role, partitions and OAuth2 Clients. Instead, use the OTDS InfoArchiveInitializer Utility, which is located in the `<INFOARCHIVE_ROOT>/bin` directory, to easily create the InfoArchive resource, access role and partitions:

Note: The configuration of OTDS connectivity are located in file `<INFOARCHIVE_ROOT>/config/otds-ia-init/application-infoarchive.otds.initializer.profile.OTDS.yml` file.

- For Windows, enter `otds-ia-init.bat`.
- For Linux, enter `otds-ia-init`.

Capture the output of the utility, as the client secrets are required later when you configure the SSO integration mode.

At this point OTDS is primed for use with InfoArchive. Proceed to activate the OTDS integration.

At this point, you can enable the `infoarchive.bootstrap` partiotioin in OTDS using the OTDS Admin Web Client. You may also configure additional partitions for your IDPs - LDAP or Active Directory. Refer to OTDS documentation to learn how to complete these tasks. Make sure to add those partitions to the access role called `infoarchive`.

To activate the OTDS SSO mode integration:

InfoArchive Gateway

- Configure the active profiles by updating the `active` property in the `config/iawebapp/application.yml` file, as illustrated below:

```
spring:  
  application:  
    name: infoarchive.gateway  
  profiles:  
    include:  
      - infoarchive.gateway.profile.OTDS
```

This tells the system that you are opting to use OTDS in the SSO mode and activates the properties found in the `<INFOARCHIVE_ROOT>/config/iawebapp/application-infoarchive.gateway.profile.OTDS.yml` file.

Deploying to Tomcat

- The Gateway may be deployed to Tomcat by dropping the `infoarchive-webapp.war` to Tomcat's `webapp/` folder. The Gateway will be deployed at the `/infoarchive-webapp` context. This needs to be taken into account. Assuming that Tomcat is running at port 8090, to activate the OTDS SSO mode integration, configure the active profiles in `webapps/infoarchive-webapp/WEB-INF/classes/application.yml`, as illustrated below:

```
spring:  
  :  
  profiles:  
    include:  
      - infoarchive.gateway.profile.OTDS  
    : ...  
    : ...  
  infoarchive:  
    gateway:  
      : ...  
      port: 8090  
      contextPath: /infoarchive-webapp/  
      : ...  
      client:  
        ssl:  
          trust-store: "D:/apache-tomcat-8.0.32/webapps/infoarchive-webapp/WEB-INF/classes/gatewayTrustStore.jks"  
          trust-store-password: abcdefg  
          trust-store-type: JKS
```

Note: The value of `infoarchive.gateway.contextPath`. It is the context path of the IA Web App as deployed to Tomcat. It must be terminated with as trailing / (slash).

The `trust-store` should point to the fully qualified file path of the trust store into which the Tomcat's (which is hosting the OTDS at HTTPS port 8443) has been imported. This is needed by the OAuth2 authentication endpoints in OTDS. Be sure to enter the correct password for the trust store.

2. Configure access to OTDS in the <INFOARCHIVE_ROOT>/config/iawebapp/application-infoarchive.gateway.profile.OTDS.yml file, as illustrated below:

```

OTDS:
  location:
    # OTDS context path - Do not change
    # Port where OTDS is running - if HTTP port: 8090
    # Host where OTDS is running - HTTPS
    httpsHost: ${OTDS.location.host}
    # Port where OTDS is running - if HTTPS httpsPort: 8443
    # OTDS context path - do not change contextPath: /otdsws
    path: ${OTDS.location.contextPath}/rest
    # Use this if OTDS can be accessed via HTTP protocol
    url: http://${OTDS.location.host}:${OTDS.location.port}${OTDS.
      location.path}
    # Use this if OTDS can be accessed via HTTPS protocol
    # url: https://${OTDS.location.host}:${OTDS.location.httpsPort}$
      {OTDS.location.path}
  # OTDS logout url
  logoutUrl: https://${OTDS.location.host}:${OTDS.location.httpsPort}
${OTDS.location.contextPath}/logout

  # Use this if OTDS can be accessed via HTTPS protocol
  # url: https://${OTDS.location.host}:${OTDS.location.httpsPort}${OTDS.location.path}
  # OTDS logout url
  logoutUrl: https://${OTDS.location.host}:${OTDS.location.httpsPort}
${OTDS.location.contextPath}/logout
  # OTDS username
  username: otadmin@otds.admin
  # OTDS password - this may be encrypted using standard InfoArchive
  # password encryption mechanism
  password: otadmin
infoarchive:
  gateway:
    protocol: http
  resource:
    # The name of the OTDS resource for this installation of InfoArchive
    # - do not change
    id: infoarchive
  clients:
    gateway:
      clientId: infoarchive.gateway
      # Enter infoarchive.gateway client secret issued by OTDS obtained
      # by running tools/bin/OTDSInfoArchiveInitializer utility
      clientSecret: "NOTSET"
      scope: otds:groups
    iawa:
      clientId: infoarchive.iawa
      # Enter infoarchive.iawa client secret issued by OTDS obtained
      # by running tools/bin/OTDSInfoArchiveInitializer utility
      clientSecret: "NOTSET"
      scope: otds:groups
      logoutUrl: ${OTDS.infoarchive.gateway.protocol}://${infoarchive.gateway.host}
${infoarchive.gateway.port}${infoarchive.gateway.contextPath}logout

  security:
    sessions: stateless
    oauth2:
      client:
        accessTokenUri: ${OTDS.infoarchive.gateway.protocol}://${infoarchive.
          gateway.host}:${infoarchive.gateway.port}${infoarchive.gateway.contextPath}
        oauth/token
        userAuthorizationUri: https://${OTDS.location.host}:${OTDS.
          location.httpsPort}/otdsws/oauth2/auth
        clientId: ${OTDS.infoarchive.clients.iawa.clientId}

```

```
clientSecret: ${OTDS.infoarchive.clients.iawa.clientSecret}
scope: ${OTDS.infoarchive.clients.iawa.scope}
authentication-scheme: form
preEstablishedRedirectUri: ${OTDS.infoarchive.gateway.protocol}:
://${infoarchive.gateway.host}:${infoarchive.gateway.port}${infoarchive.
gateway.contextPath}login
useCurrentUri: false
resource:
jwt:
keyValue: NOTUSED

zuul:
debug:
request: false
routes:
authorize:
path: /oauth/authorize/**
url: https://${OTDS.location.host}:${OTDS.location.httpsPort}
${OTDS.location.contextPath}/oauth2/auth
token:
path: /oauth/token/**
url: https://${OTDS.location.host}:${OTDS.location.httpsPort}
${OTDS.location.contextPath}/oauth2/token
sensitiveHeaders: ""
otdslogin:
path: /otdsws/**
url: https://${OTDS.location.host}:${OTDS.location.httpsPort}
${OTDS.location.contextPath}
sensitiveHeaders: ""
```

Note: The client secrets for infoarchive.gateway and infoarchive.iawa OAuth2 clients are generated by OTDS when the clients are created. These need to be specified in the file above.

The secrets were issued when the otds-ia-init utility was run.

InfoArchive Server

1. In this mode OTDS issues the JWT token. These tokens are in different format and, therefore, need to be parsed differently. An activation of a OTDS profile is required in the config/iaserver/application.yml file, as illustrated below:

```
spring:
application:
name: infoarchive.ias
profiles:
include: infoarchive.ias.profile.JWT,infoarchive.ias.profile.OTDS
```

This enables IA Server to perform proper decoding of the OTDS issued JWT tokens.

2. Configure the access to OTDS in the <INFOARCHIVE_ROOT>/config/iaserver/application-infoarchive.ias.profile.OTDS.yml file, as illustrated below:

```
OTDS:
LOCATION:
host: localhost
port: 8090
httpsPort: 8443
path: /otdsws/rest
url: http://${OTDS.LOCATION.host}:${OTDS.LOCATION.port}${OTDS.LOCATION.path}
username: otadmin@otds.admin
password: otadmin
infoarchive:
resource:
id: infoarchive
security:
```

```

oauth2:
  resource:
    jwt:
      keyValue: NOTUSED

```

This also enables IA Server to persist the Group > Role mapping back into OTDS, as well as xDB.

InfoArchive IA Shell

1. The Oauth2 client representing the IA Shell is defined in OTDS. Therefore, since OTDS issues the client secrets for OAuth2 clients, you must update the following files so that InfoArchive IA Shell will work in this mode:

Note: The name of OAuth2 Client for IA Shell is `infoarchive.cli`. The secret value for this client should be used below.

In the `<INFOARCHIVE_ROOT>/config/iashell/application.properties` file:

- ```

:
gateway.clientSecret=HYBxUa1c5aPWgbMPSjXVZTaPvoh3nQn7
:
2. In the <INFOARCHIVE_ROOT>/examples/legacy-ant-applications/build
.properties file:

```

```

:
:
gatewaySecret=HYBxUa1c5aPWgbMPSjXVZTaPvoh3nQn7
:
:
InfoArchive username username=sue@iacustomer.com
```

```

InfoArchive password password=Password@123
:
:
```

**Note:** You have to use the passwords for SUPER role user, such as `sue@iacustomer.com` that is configured in OTDS.

**Tip:** A quick method to determine that the OTDS integration was successful is that you will see the following login screen, which is different than the usual login screen:



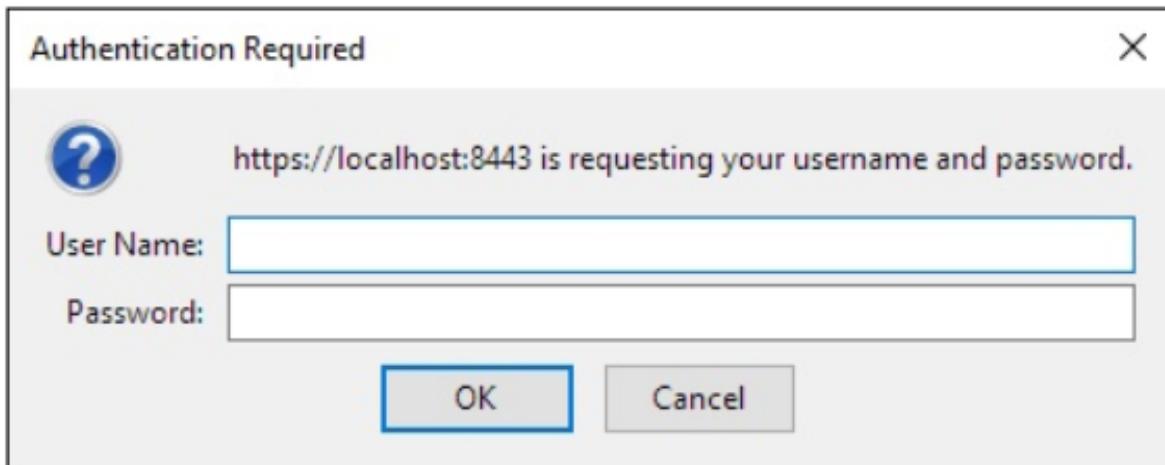
## Troubleshooting OTDS Issues

Ensure that all of the configuration steps are performed (for example, proper spring profiles for both server and gateway).

Ensure that the truststore in the <INFOARCHIVE\_ROOT>/config/iawebapp directory is in place and has the Tomcat certificate.

If you still experience issues, consult the [Troubleshooting](#) chapter of the *OpenText Directory Services: Installation and Administration Guide*.

Depending on the configuration of OTDS and your browser, you may see the following dialog prior to seeing the OTDS login dialog:



Simply **Cancel** the dialog and the OTDS login dialog will be shown. This happens because of the http.negotiate Authentication handler is enabled in OTDS. You can disable that in the OTDS Admin web application here:

| Name                                                | Description                                          | Scope | Priority | Status  | Actions                                            |
|-----------------------------------------------------|------------------------------------------------------|-------|----------|---------|----------------------------------------------------|
| <input checked="" type="checkbox"/> http.negotiate  | Handles "Negotiate" authentication with the browser. |       | 20       | enab... | <a href="#">Properties</a> <a href="#">Disable</a> |
| <input checked="" type="checkbox"/> token.negotiate | Handles token-based "Negotiate" authentication.      |       | 10       | enab... | <a href="#">Properties</a>                         |

You need to leave the http.negotiate Authentication handler as enabled, however, if you are using the Kerberos > NTLM > Windows desktop authentication integration with OTDS. This will also require changes to the browser configuration so that the Authentication Required dialog is not shown. Refer to <http://webapp.opentext.com/piroot/otds/v160202/otds-iwc/en/html/jframe.htm?tr-singlesign> for further information.

## Setting the Login Format

OTDS provides the ability to specify multiple login formats, one of which is where by the domain prefix can be specified. Refer to the [OpenText Directory Services: Installation and Administration Guide](#) for further information.

Basically, OTDS supports via oExternalID4 attribute of the user the following format with Domain name:

```
oExternalID4 = OPENTEXT\franz
```

**Note:** The use of an NT Domain name prefix (for example, CORP > johndoe) makes sense when the synchronized partition's IDP is Active Directory-based. For LDAP and non-synchronized partitions, it is the name of the partition.

## Validating the Integration of SAML 2.0 with OTDS and InfoArchive

As of the InfoArchive 16EP3 release, InfoArchive allows you to configure the following authentication options:

- OTDS to support multiple Lightweight Directory Access Protocol (LDAP)/active directory setups
- Single sign-on (SSO)
- **Security Assertion Markup Language (SAML)**

OpenText Directory Services (OTDS) supports integration with SAML 2.0 Identity Providers (IDPs) out of the box.

This section describes the process of configuration and uses Okta throughout the procedure. Okta is just one example of a SAML 2.0 IDP and is used in this section to help illustrate the configuration process. A similar process will work for other SAML 2.0 IDPs (for example, Ping Identity, OneLogin) with only minor changes.

Also refer to the [OpenText Directory Services Installation and Administration Guide](#) for further information about SAML.

1. Configure integration of OTDS with InfoArchive. Refer to [Configuring OpenText Directory Services](#) to perform this step.
2. Configure the Okta IDP to recognize the OTDS instance as a SAML 2.0 service provider/
  - a. Create an application `otds-localhost` that represents the OTDS instance as a service provider.
  - b. Configure it as a SAML 2.0 application.
    - i. Enter the following details for a SAML 2.0 application:

| Setting Name                          | Value                                             |
|---------------------------------------|---------------------------------------------------|
| Single sign on URL                    | <code>https://localhost:8443/otdssws/login</code> |
| Name ID format                        | EmailAddress                                      |
| GROUP ATTRIBUTE STATEMENTS (OPTIONAL) |                                                   |
| Audience URI (SP Entity ID)           | <code>https://localhost:8443/otdssws/login</code> |
| ATTRIBUTE STATEMENTS                  |                                                   |
| Application username                  | Email                                             |

- ii. In the Advanced section, enter the following details to configure the Single Sign Out:

| Setting Name          | Value                                                                                                        |
|-----------------------|--------------------------------------------------------------------------------------------------------------|
| Single Logout URL     | <code>https://localhost:8443/otdssws/logout</code>                                                           |
| SP Issuer             | <code>https://localhost:8443/otdssws/login</code>                                                            |
| Signature Certificate | Upload the Tomcat certificate. Use the certificate for Tomcat that hosts the OTDS Administration web client. |

- c. Create the following groups:
- OKTA\_ADMINISTRATOR
  - OKTA\_AUDITOR
  - OKTA\_BUSINESS\_OWNER
  - OKTA\_DEVELOPER
  - OKTA\_EDISCOVERY\_ADMINISTRATOR
  - OKTA\_END\_USER
  - OKTA\_IT\_OWNER
  - OKTA\_RETENTION\_MANAGER
3. Create the following users and assign each user to a group:

**Tip:** For the secondary e-mail, use an e-mail address where you can receive the user activation and password reset links.

| User Name                  | Group                         |
|----------------------------|-------------------------------|
| adam@okta.iacustomer.com   | OKTA_ADMINISTRATOR            |
| audrey@okat.iacustomer.com | OKTA_AUDITOR                  |
| bob@okat.iacustomer.com    | OKTA_BUSINESS_OWNER           |
| connie@okta.iacustomer.com | OKTA_DEVELOPER                |
| erica@okta.iacustomer.com  | OKTA_EDISCOVERY_ADMINISTRATOR |
| emma@okta.iacustomer.com   | OKTA_END_USER                 |
| imran@okta.iacustomer.com  | OKTA_IT_OWNER                 |
| rita@okta.iacustomer.com   | OKTA_RETENTION_MANAGER        |

- a. Activate the users. This sends an e-mail that contains an activation link to the primary and secondary e-mail address of each user.
- b. Activate each user once you receive the activation link.
- c. Reset the passwords for each user. The system will send a password reset link to the primary and secondary e-mail addresses. For example:

| User Name                  | Password     |
|----------------------------|--------------|
| adam@okta.iacustomer.com   | Password@123 |
| audrey@okat.iacustomer.com | Password@123 |
| bob@okat.iacustomer.com    | Password@123 |
| connie@okta.iacustomer.com | Password@123 |
| erica@okta.iacustomer.com  | Password@123 |
| emma@okta.iacustomer.com   | Password@123 |

| User Name                 | Password     |
|---------------------------|--------------|
| imran@okta.iacustomer.com | Password@123 |
| rita@okta.iacustomer.com  | Password@123 |

- d. Assign the groups and users to the `otds-localhost` application
4. Copy the metadata URL, which you will need in the next section:

## Configuring Okta as a SAML 2.0 IDP Inside OTDS

This section demonstrates how to:

- [Create a non-synchronized partition called `infoarchive-okta`.](#)
- [Create the SAMPL 2.0 authentication handler named `okta`.](#)

### Creating a Non-synchronized Partition `infoarchive-okta`

This section illustrates how to create a non-synchronized partition called `infoarchive-okta`:

1. Add this partition to the `infoarchive` access role so that the users and groups can be used with OTDS/InfoArchive integration
2. Create the following groups when not using auto-provisioning mode:
  - `OKTA_ADMINISTRATOR`
  - `OKTA_AUDITOR`
  - `OKTA_BUSINESS_OWNER`
  - `OKTA_DEVELOPER`
  - `OKTA_EDISCOVERY_ADMINISTRATOR`
  - `OKTA_END_USER`
  - `OKTA_IT_OWNER`
  - `OKTA_RETENTION_MANAGER`
3. If using auto-provisioning mode of authentication handler (see below), this step can be skipped.  
The groups will be auto-provisioned in OTDS as and when they successfully sign in via Okta.

There is an issue, however, if a brand new group is added due to auto-provisioning, the Group to Role mapping for it will be missing and, therefore, a user may not receive the intended authorization inside InfoArchive. Once the Group to Role mapping for this new group is created inside InfoArchive via the **Administration > Groups** view (or inside OTDS Administration web client), only then this group becomes usable. For that reason, it is recommended that you pre-create the groups in OTDS partition (for example, `infoarchive-okta`).

4. Log into InfoArchive as an administrator (for example, `adam@iacsuomer.com` from the `infoarchive.bootstrap` partition). Go to the **Administration > Groups** view to ensure that the groups you created in previous step are displayed. Map them to the roles.

5. If using the auto-provisioning mode of the authentication handler, you do not need to create the users inside OTDS. The users will be auto-provisioned in OTDS when they successfully sign in via Okta.

If not using auto-provisioning mode, pre-create users.

## Creating the SAMPL 2.0 Authentication Handler

This section illustrates how to create the SAMPL 2.0 authentication handler named `okta`.

1. Associate it with the `infoarchive-okta` partition.
2. Configure this using one of the following modes:
  - a. Always active: In this mode, the users will never see the OTDS login screen, but will always be redirected to Okta login screen.
  - b. No always active: In this mode, the users will first see the OTDS login screen. They can login using a user that is native to OTDS. Alternatively, they can click on the Sign in with Okta button, and they will be redirected to Okta login screen. There they can only sign in using users known to Okta.
3. To verify the configuration, access the IA Web App:
  - If the Okta Authentication Handler is configured as Always Active, the browser redirects you to the Okta login screen.
  - If the Okta Authentication Handler is not configured as Always Active, the browser redirects you to the OTDS login screen. To use Okta for login, you will see a Sign In displayed with an Okta button. Click it to access Okta login screen.
    - a. Sign in with one of the users that you configured in the previous step.  
If the Okta Authentication handler was configured for auto-provisioning of the user, the user is created in the `infoarchive-okta` partition in the OTDS Administration web client.  
You should be able to access the IA Web App with the authorization consistent with the user's groups that are mapped to roles.  
Use **Sign Out of OTDS** to sign out of OTDS and Okta.  
This navigates you to the OTDS or Okta login screen, depending on the Always Active setting of the Okta Authentication Handler.

- b. Sign in with a different user that you configured in the previous step.

# Working with the Deployment Configuration Files

The most important deployment configuration files are stored in the following folders:

- <INFOARCHIVE\_ROOT>/config/iawebapp/application.yml: Refer to [Working with Gateway and InfoArchive Web Application Configuration Files](#) for more information.
- <INFOARCHIVE\_ROOT>/config/iaserver/application.yml: Refer to [Working with the Server's application.yml File](#) for more information.
- <INFOARCHIVE\_ROOT>/xdb/conf/xdb.properties: Refer to [Working with the xdb.properties File](#) for more information.

## Changing the Default Ports for InfoArchive Components

This section illustrates where to update the default ports for the xDB server, IA Server and the IA Web App:

- For the xDB server, update the XHIVE\_SERVER\_PORT and XHIVE\_ADMIN\_PORT properties in the <INFOARCHIVE\_ROOT>/xdb/conf/xdb.properties file.
- For IA Server, update the server.port property in the <INFOARCHIVE\_ROOT>/config/iaserver/application.yml file.
- For IA Shell, update the xdbFederation.bootstrap property in the <INFOARCHIVE\_ROOT>/examples/applications/default.properties file.
- For the applications, update the default.gateway and default.restAPI properties in the <INFOARCHIVE\_ROOT>/config/iashell/application.yml file.
- The IA Web Appn can be deployed using two different methods:
  - It can run as a standalone when the Spring Boot application is run as a process or as a service.
  - It can also be deployed to Tomcat.

Update the infoarchive.gateway.host and infoarchive.gateway.port properties in the <INFOARCHIVE\_ROOT>/config/iawebapp/application.yml file.

This is later used as:

```
server:
 host: ${infoarchive.gateway.host}
 port: ${infoarchive.gateway.port}
```

When deployed as standalone the server.\* properties are used by the embedded Tomcat container. When the web application is deployed to Tomcat, however, the host and port are really controlled by the Tomcat configuration. The web application implementation still needs to know about these values and that is why it is better to use infoarchive.\* properties. Also, if there is a load balancer in front of the web application, these infoarchive.\* properties need to point to it.

# Working with Gateway and InfoArchive Web Application Configuration Files

A version of the configuration files is bundled in the `infoarchive-webapp.(jar|war)` file. The files are also extracted into the `config/iawebapp` folder and are picked up because of the `-Dspring.config.location=file:config/webapp/` parameter to the IA Web App start-up script via one of the following:

```
<INFOARCHIVE_ROOT>/bin/ iawebapp.bat (windows).
<INFOARCHIVE_ROOT>/bin/ iawebapp (linux).
```

The `<INFOARCHIVE_ROOT>/config/iawebapp/application.yml` file is the main configuration file of IA Web App. The configurations in this file further impact and use other configuration files. The out-of-box file contains the following configuration:

| Property                                                                               | Description                                                                                                                                                                                                                                                                                                                                   |
|----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>spring:</code>                                                                   |                                                                                                                                                                                                                                                                                                                                               |
| <code>application:</code>                                                              |                                                                                                                                                                                                                                                                                                                                               |
| <code>name: infoarchive.gateway</code>                                                 | Names the Gateway/IA Web App                                                                                                                                                                                                                                                                                                                  |
| <code>profiles:</code>                                                                 |                                                                                                                                                                                                                                                                                                                                               |
| <code>include : -<br/>infoarchive.gateway.profile<br/>.AUTHENTICATION_IN_MEMORY</code> | List of profiles on each line prefixed with a – (dash). See the profiles section below. <code>infoarchive.gateway.profile</code> <code>.AUTHENTICATION_IN_MEMORY</code> should                                                                                                                                                                |
| <code>cloud:</code>                                                                    |                                                                                                                                                                                                                                                                                                                                               |
| <code>config:</code>                                                                   |                                                                                                                                                                                                                                                                                                                                               |
| <code>enabled: false</code>                                                            | This should be left as false. For future use.                                                                                                                                                                                                                                                                                                 |
| <code>infoarchive:</code>                                                              |                                                                                                                                                                                                                                                                                                                                               |
| <code>gateway:</code>                                                                  |                                                                                                                                                                                                                                                                                                                                               |
| <code>host: localhost</code>                                                           | The host on which the Gateway/web application is running. When running in HTTPS mode, this host name must match with the CN= name used during the creation of the Gateway certificate. This is because there is component in Gateway that acts as its own client. If the host name does not match the CN= name, a status 500 error is issued. |
| <code>port: 8080</code>                                                                | The port at which the Gateway/web application is running.                                                                                                                                                                                                                                                                                     |
| <code>contextPath: /</code>                                                            | The context at which the Gateway/web application is running. Set it only if you want to run Gateway/web applications at context other than /. If running in external Tomcat container this is ignored. Instead the context at which the war is deployed is used internally.                                                                   |
| <code>management:</code>                                                               |                                                                                                                                                                                                                                                                                                                                               |

| Property                                                       | Description                                                                                                                                                                                                                                                                     |
|----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| add-application-context-header: false                          | Disable X-Application-Context response header                                                                                                                                                                                                                                   |
| endpoints:                                                     |                                                                                                                                                                                                                                                                                 |
| enabled: false                                                 |                                                                                                                                                                                                                                                                                 |
| token:                                                         |                                                                                                                                                                                                                                                                                 |
| secret: secret                                                 | <p>The token used to encrypt the JWT token. This must match the same value in the IA Server application.yml file.</p> <p><b>Note:</b> This value must be encrypted if Encrypted password mode is turned on. Secrets are considered like passwords as sensitive information.</p> |
| client                                                         | This is needed when Gateway is a HTTPS or LDAPS client (SSL/TLS Client)                                                                                                                                                                                                         |
| ssl:                                                           |                                                                                                                                                                                                                                                                                 |
| trust-store: "config/iawebapp/gatewayTrustStore.jks"           |                                                                                                                                                                                                                                                                                 |
| trust-store-password: "abcdefg"                                | Refer to that documentation here Using encrypted passwords in configuration files for details.                                                                                                                                                                                  |
| trust-store-type: JKS                                          |                                                                                                                                                                                                                                                                                 |
| # Following entries is required for two-way TLS authentication | This is to enable two-way TLS (HTTPS) communication between Gateway+IAWA and IAS                                                                                                                                                                                                |
| # key-store: "config/webapp/gatewayKeyStore.jks"               |                                                                                                                                                                                                                                                                                 |
| # key-store-password: abcdefg                                  |                                                                                                                                                                                                                                                                                 |
| # key-store-type: JKS                                          |                                                                                                                                                                                                                                                                                 |
| webapp:                                                        |                                                                                                                                                                                                                                                                                 |
| customization:                                                 |                                                                                                                                                                                                                                                                                 |
| location: "file:config/webapp/customization/"                  | "file:config/webapp/customization/"                                                                                                                                                                                                                                             |
| server:                                                        |                                                                                                                                                                                                                                                                                 |
| host: \${infoarchive.gateway.host}                             | <p>DO NOT CHANGE</p> <p>This value must be the same as the infoarchive.gateway.host values.</p>                                                                                                                                                                                 |
| port: \${infoarchive.gateway.port}                             | <p>DO NOT CHANGE</p> <p>This value must be the same as the infoarchive.gateway.host values.</p>                                                                                                                                                                                 |

| Property                                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| contextPath: \${infoarchive.gateway.contextPath} | DO NOT CHANGE<br><br>This value must be the same as the infoarchive.gateway.host values.                                                                                                                                                                                                                                                                                                                                     |
| zuul:                                            |                                                                                                                                                                                                                                                                                                                                                                                                                              |
| routes:                                          |                                                                                                                                                                                                                                                                                                                                                                                                                              |
| restapi:                                         |                                                                                                                                                                                                                                                                                                                                                                                                                              |
| path: /restapi/**                                | DO NOT CHANGE. Mapping for REST API.                                                                                                                                                                                                                                                                                                                                                                                         |
| sensitiveHeaders: ""                             | DO NOT CHANGE.                                                                                                                                                                                                                                                                                                                                                                                                               |
| url: http://localhost:8080/                      | The URL of IA Server. This will be overridden when infoarchive.profile.HTTPS profile is active. The hostname part of the URL must match the CN= name used during the creation of the server certificate. If the CN= and hostname do not match, there will be a Status 500 error and a communication failure will issue a back-end error message.                                                                             |
| addProxyHeaders: true                            | DO NOT CHANGE.                                                                                                                                                                                                                                                                                                                                                                                                               |
| host                                             |                                                                                                                                                                                                                                                                                                                                                                                                                              |
| socket-timeout-millis: 60000                     | DO NOT CHANGE. Timeout in second for requests from Gateway/IAWA to IAS.                                                                                                                                                                                                                                                                                                                                                      |
| maxTotalConnections: 200                         | The setting for maximum number total connections used by Zuul proxy, which routes traffic to IA Server on the /restapi path. Even though changing this is not generally recommended, you may change it if you have many clients such as, IAWA from Browser, CLI, iashell connecting to the Gateway. Technical Detail of this setting for Zuul: The maximum number of total connections the proxy can hold open to back-ends. |
| maxPerRouteConnections: 20                       | Gateway routes traffic to IA Server on the /restapi path. Even though changing this is not generally recommended, you may change it if you have many clients such as, IAWA from Browser, CLI, iashell connecting to the Gateway. Technical Detail of this setting for Zuul: The maximum number of connections that can be used by a single route.                                                                            |
| SimpleHostRoutingFilter:                         |                                                                                                                                                                                                                                                                                                                                                                                                                              |
| route:                                           |                                                                                                                                                                                                                                                                                                                                                                                                                              |
| disable: true                                    | DO NOT CHANGE                                                                                                                                                                                                                                                                                                                                                                                                                |
| log.level.threshold: 'WARN'                      | Logging level.                                                                                                                                                                                                                                                                                                                                                                                                               |

| Property                                                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| # Settings for encryption of passwords appearing in this configuration file | This section is same as the service side. For more information, refer to the Using Encrypted Passwords in Configuration Files section in the <i>Encryption User Guide</i> .                                                                                                                                                                                                                                                                                                                                                                                              |
| passwordEncryption:                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| enabled: false                                                              | Set to 'true' to enable encryption of secrets and passwords.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| noPromptStartup: false                                                      | <p>Set this to true for unattended mode start of Gateway+IAWA when encryption has been enabled (see the previous property). This is required to run Gateway and the web application as a service.</p> <p>If set to true, the user is not prompted to enter a password, but has to follow the procedure documented in the Using Encrypted Passwords in Configuration Files section of the <i>Encryption User Guide</i>.</p> <p>For more information, refer to the Starting the InfoArchive Server without Password Input section in the <i>Encryption User Guide</i>.</p> |
| keyStorePath: "config/iawebapp/keystore.jceks"                              | Basically the location of the keyStorePath should be same as that set in the server's application.yml file.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| keyStoreType: jceks                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| keyID: IA_PASSWORD_ENCRYPTION_KEY                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| encryptionAlgorithm: AES                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| encryptionMode: CBC                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| paddingScheme: PKCS5PADDING                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| securityProvider: SunJCE                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| keySize: 256                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| gemaltoUserName:                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| gemaltoClientCertificat:                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| gemaltoClientCertificate                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| gemaltoPropertiesFile: "config/iawebapp/IngrianNAE.properties"              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| gemaltoGroup:                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| spring:                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| profiles: "infoarchive.profile.HTTPS"                                       | Profile to activate HTTPS mode of Gateway/web application. This is set by the -ssl variant of the startup scripts in bin/ folder.                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| server:                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| ssl:                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

| Property                                             | Description                                                                                                                                          |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| keystore: "file:config/iawebapp/gatewayKeyStore.jks" | Location of keystore. To point to external keystore use file:/ URLs.                                                                                 |
| keyStorePassword: abcdefg                            | Keystore password. For more information, refer to the Using Encrypted Passwords in Configuration Files section in the <i>Encryption User Guide</i> . |
| keyPassword: abcdefg                                 | Key password. For more information, refer to the Using Encrypted Passwords in Configuration Files section in the <i>Encryption User Guide</i> .      |
| trustStoreType: JKS                                  | The type of the trust store.                                                                                                                         |
| infoarchive:                                         |                                                                                                                                                      |
| gateway:                                             |                                                                                                                                                      |
| zuul:                                                |                                                                                                                                                      |
| routes:                                              |                                                                                                                                                      |
| restapi:                                             |                                                                                                                                                      |
| path: /restapi/**                                    |                                                                                                                                                      |
| sensitiveHeaders: ""                                 |                                                                                                                                                      |
| url: "https://localhost:8765/"                       | https:// URL to IAS running in HTTPS mode. Both Gateway/IAWA and IAS must run either in HTTP mode or HTTPS mode.                                     |

Profiles activate and deactivate specific functionality of the IA Web App. The following are authentication related profiles:

- infoarchive.gateway.profile.AUTHENTICATION\_IN\_MEMORY - this is out-of-box active profile and should come later than the other profiles below.
- infoarchive.gateway.profile.AUTHENTICATION\_ACTIVE\_DIRECTORY - use this profile to activate connection with Active Directory.
- infoarchive.gateway.profile.AUTHENTICATION\_EXTERNAL\_LDAP - use this profile to activate connection with external LDAP.
- application-infoarchive.gateway.profile.AUTHENTICATION\_OTDS – use this profile to activate simple integration with OTDS.
- application-infoarchive.gateway.profile.OTDS – use this profile to activate SSO integration with OTDS.

The HTTPS (SSL) Related profile activates the HTTPS(SSL) mode of Gateway/web application. To activate this profile, add it to the spring:application:profiles:include property using a list of profile names on each line prefixed with a – (dash). Once this profile is activated, the clients must access the Gateway/web application using the https ://... url (note the 's' in https).

The <INFOARCHIVE\_ROOT>/config/iawebapp/application-infoarchive.gateway.profile.AUTHENTICATION\_IN\_MEMORY.properties file configures the out-of-box in-memory users. This file is active out-of-box for easy setup, demo-ability and bootstrapping of the customer installation. This file is picked up when the spring:application:profiles:include property contains the profile name infoarchive.gateway.profile.AUTHENTICATION\_IN\_MEMORY:

```
spring:
 application:
 ...
 profiles:
 include:
 - infoarchive.gateway.profile.AUTHENTICATION_IN_MEMORY
```

The following illustrates the out-of-box content of the file:

```
AUTHENTICATION_IN_MEMORY.group.groups[0]=GROUP_ADMINISTRATOR
AUTHENTICATION_IN_MEMORY.group.groups[1]=GROUP_BUSINESS_OWNER
AUTHENTICATION_IN_MEMORY.group.groups[2]=GROUP_DEVELOPER
AUTHENTICATION_IN_MEMORY.group.groups[3]=GROUP_END_USER
AUTHENTICATION_IN_MEMORY.group.groups[4]=GROUP_IT_OWNER
AUTHENTICATION_IN_MEMORY.group.groups[5]=GROUP_RETENTION_MANAGER
AUTHENTICATION_IN_MEMORY.group.groups[6]=GROUP_EDISCOVERY_ADMINISTRATOR
AUTHENTICATION_IN_MEMORY.group.groups[7]=GROUP_AUDITOR

Example of how to add additional roles if needed. This can be done in external
application.properties file.
AUTHENTICATION_IN_MEMORY.group.additionalGroups[0]=GROUP_COMPLIANCE_DEPARTMENT

AUTHENTICATION_IN_MEMORY.user.users[0]=adam@iacustomer.com,password,
${AUTHENTICATION_IN_MEMORY.group.groups[0]}
AUTHENTICATION_IN_MEMORY.user.users[1]=bob@iacustomer.com,password,
${AUTHENTICATION_IN_MEMORY.group.groups[1]}
AUTHENTICATION_IN_MEMORY.user.users[2]=connie@iacustomer.com,password,
${AUTHENTICATION_IN_MEMORY.group.groups[2]}
AUTHENTICATION_IN_MEMORY.user.users[3]=emma@iacustomer.com,password,
${AUTHENTICATION_IN_MEMORY.group.groups[3]}
AUTHENTICATION_IN_MEMORY.user.users[4]=imran@iacustomer.com,password,
${AUTHENTICATION_IN_MEMORY.group.groups[4]}
AUTHENTICATION_IN_MEMORY.user.users[5]=rita@iacustomer.com,password,
${AUTHENTICATION_IN_MEMORY.group.groups[5]}
AUTHENTICATION_IN_MEMORY.user.users[6]=sue@iacustomer.com,password,
GROUP_ADMINISTRATOR|GROUP_BUSINESS_OWNER|GROUP_DEVELOPER|GROUP_EDISCOVERY_ADMINISTRATOR|
GROUP_END_USER|GROUP_IT_OWNER|GROUP_RETENTION_MANAGER
AUTHENTICATION_IN_MEMORY.user.users[7]=erica@iacustomer.com,password,
${AUTHENTICATION_IN_MEMORY.group.groups[6]}
AUTHENTICATION_IN_MEMORY.user.users[8]=audrey@iacustomer.com,password,
${AUTHENTICATION_IN_MEMORY.group.groups[7]}

Example of how to add additional users if needed. This can be done in external
application.properties file.
AUTHENTICATION_IN_MEMORY.user.additionalUsers[0]=dave@iacustomer.com,password,
${AUTHENTICATION_IN_MEMORY.group.groups[2]}
```

The <INFOARCHIVE\_ROOT>/config/iawebapp/application-infoarchive.gateway.profile.AUTHENTICATION\_EXTERNAL\_LDAP.properties file configures the external LDAP systems such as Apache DS. This file is picked up when the **spring:application:profiles:include** Spring property contains the profile name infoarchive.gateway.profile.AUTHENTICATION\_EXTERNAL\_LDAP:

```
spring:
 application:
 ...
 profiles:
 include:
 - infoarchive.gateway.profile.AUTHENTICATION_EXTERNAL_LDAP
```

The following illustrates the out-of-box content of the file:

| Property                                                                                                                                | Description                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| # External LDAP                                                                                                                         | This is just an example for connecting to Apache DS service running on local host. You will have to change the settings for your LDAP.                                                                                                                                                                                                                       |
| # The following two are default administrator and password for Apache DS#AUTHENTICATION_EXTERNAL_LDAP<br>.managerDn=uid=admin,ou=system | Supports non-personal account NPA.                                                                                                                                                                                                                                                                                                                           |
| #AUTHENTICATION_EXTERNAL_LDAP<br>.managerPassword=secret                                                                                | Supports non-personal account NPA.                                                                                                                                                                                                                                                                                                                           |
| AUTHENTICATION_EXTERNAL_LDAP.userDnPatterns<br>=uid=\{0\},ou\=people                                                                    |                                                                                                                                                                                                                                                                                                                                                              |
| AUTHENTICATION_EXTERNAL_LDAP.userSearchFilter<br>=uid=\{0\}                                                                             |                                                                                                                                                                                                                                                                                                                                                              |
| AUTHENTICATION_EXTERNAL_LDAP.userSearchBase<br>=ou\=people                                                                              |                                                                                                                                                                                                                                                                                                                                                              |
| AUTHENTICATION_EXTERNAL_LDAP.groupSearchFilter<br>=(member\=\{0\})                                                                      |                                                                                                                                                                                                                                                                                                                                                              |
| AUTHENTICATION_EXTERNAL_LDAP.groupSearchBase<br>=ou\=groups                                                                             |                                                                                                                                                                                                                                                                                                                                                              |
| AUTHENTICATION_EXTERNAL_LDAP.url=ldap:/<br>/localhost:10389/dc\=infoarchive,dc\=emc,dc\=com                                             | URL + context of local Apache DS LDAP service.                                                                                                                                                                                                                                                                                                               |
| AUTHENTICATION_EXTERNAL_LDAP.url=ldaps:/<br>/localhost:10636/dc\=infoarchive,dc\=emc,dc\=com                                            | Now the secure ldaps:// protocol is supported. For that you have to: <ul style="list-style-type: none"> <li>• Generate or obtain a certificate for LDAP. Populate it in the Keystore and configure the Keystore for LDAP server. This may have been for the LDAP server. Import the certificate for LDAP server into the Gateway+IAWA truststore.</li> </ul> |

| Property                                                                                                            | Description                                                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| # Optionally specify objectClass for groups - defaults to groupOfNames                                              | Optionally specify objectClass for groups - defaults to groupOfNames                                                                                            |
| # AUTHENTICATION_EXTERNAL_LDAP<br>.groupObjectClass=groupOfNames                                                    | Sometimes the name of objectClass is different than the default                                                                                                 |
| # This will be added with (objectClass="groupOfNames"<br>&(...))<br><br># AUTHENTICATION_EXTERNAL_LDAP.groupFilter= | This will be added with (objectClass="groupOfNames"<br>&(...))<br><br>For example, (cn=*INFOARCHIVE*) will only match groups that have INFOARCHIVE in the name. |

This profile can be used in conjunction with other profiles. For example:

```
spring:
 application:
 ...
 profiles:
 include:
 - infoarchive.gateway.profile.AUTHENTICATION_IN_MEMORY,
 - infoarchive.gateway.profile.AUTHENTICATION_EXTERNAL_LDAP
```

The <INFOARCHIVE\_ROOT>/config/iawebapp/application-infoarchive.gateway.profile.AUTHENTICATION\_ACTIVE\_DIRECTORY.properties file configures the connection to an external Active Directory server, such as the Microsoft Active Directory Server. This file is picked up when the spring:application:profiles:active Spring property contains the profile name infoarchive.gateway.profile.AUTHENTICATION\_ACTIVE\_DIRECTORY:

```
spring:
 application:
 ...
 profiles:
 include:
 - infoarchive.gateway.profile.AUTHENTICATION_ACTIVE_DIRECTORY
```

The following illustrates the out-of-box content of the file:

| Property                                                                                     | Description                                                                                                               |
|----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| # External ACTIVE DIRECTORY                                                                  | This is just an example for connecting to an Active Directory server. You will have to change the settings for your LDAP. |
| AUTHENTICATION_ACTIVE_DIRECTORY<br>.managerDn=cn\=Administrator,cn\=users,dc\=iigads,dc\=com |                                                                                                                           |

| Property                                                                                                 | Description                                                                                                                                                                                                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AUTHENTICATION_ACTIVE_DIRECTORY<br>.managerPassword=Password@123                                         | The Password@123 is the just an example. Change it to suit your AD access. Needs to be changed to encrypted form if the secret and password encryption is turned on in application.yml file above. For more information, refer to the Using Encrypted Passwords in Configuration Files section in the <i>Encryption User Guide</i> . |
| AUTHENTICATION_ACTIVE_DIRECTORY<br>.userDnPatterns=cn=\{0\},ou\=Users,ou\=infoarchive,dc\=iigads,dc\=com |                                                                                                                                                                                                                                                                                                                                      |
| AUTHENTICATION_ACTIVE_DIRECTORY<br>.userSearchFilter=sAMAccountName\=\{0\}                               |                                                                                                                                                                                                                                                                                                                                      |
| AUTHENTICATION_ACTIVE_DIRECTORY<br>.userSearchBase=ou\=Users,ou\=infoarchive,dc\=iigads,dc\=com          |                                                                                                                                                                                                                                                                                                                                      |
| AUTHENTICATION_ACTIVE_DIRECTORY<br>.groupSearchFilter=(member\=\{0\})                                    |                                                                                                                                                                                                                                                                                                                                      |
| AUTHENTICATION_ACTIVE_DIRECTORY<br>.groupSearchBase=ou\=Groups,ou\=infoarchive,dc\=iigads,dc\=com        |                                                                                                                                                                                                                                                                                                                                      |
| AUTHENTICATION_ACTIVE_DIRECTORY<br>.url=ldap://10.31.70.140:389/                                         | URL Location of AD server.                                                                                                                                                                                                                                                                                                           |
| # Optionally specify objectClass for groups - defaults to group                                          | Optionally specify objectClass for groups - defaults to group                                                                                                                                                                                                                                                                        |
| # AUTHENTICATION_ACTIVE_DIRECTORY<br>.groupObjectClass=group                                             | Sometimes the name of objectClass is different than the default                                                                                                                                                                                                                                                                      |
| # Optional filter for groups. This will be added with (objectClass="group" &(...))                       | Optional filter for groups. This will be added with (objectClass="group" &(...))                                                                                                                                                                                                                                                     |
| # AUTHENTICATION_ACTIVE_DIRECTORY<br>.groupFilter=                                                       | For example, (cn=*INFOARCHIVE*) will only match groups that have INFOARCHIVE in the name.                                                                                                                                                                                                                                            |

This profile can be used in conjunction with other profiles. For example:

```
spring:
 application:
 ...
 profiles:
 include:
 - infoarchive.gateway.profile.AUTHENTICATION_IN_MEMORY
 - infoarchive.gateway.profile.AUTHENTICATION_ACTIVE_DIRECTORY
 profile.AUTHENTICATION_ACTIVE_DIRECTORY
```

InfoArchive uses OAuth2 for authentication. The <INFOARCHIVE\_ROOT>/config/iawebapp/application-Clients.yml file configures all the out-of-box OAuth2 clients of InfoArchive. Customers may add additional clients to integrate applications with InfoArchive (for example, via REST APIs).

The following illustrates the out-of-box content of the file:

| Property                    | Description                                                                                                                                                                                                                                                                                          |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| clientId                    | ID of client. InfoArchive supports the following values: <ul style="list-style-type: none"> <li>• infoarchive.gateway</li> <li>• infoarchive.iawa</li> <li>• infoarchive.jdbc</li> <li>• infoarchive.cli</li> </ul>                                                                                  |
| authorizedGrantTypes        | Specifies which operations are authorized. Do not change the values for each client.                                                                                                                                                                                                                 |
| authorities                 | The value should be role.administrator for the infoarchive.gateway. Otherwise, the value must be ROLE_TRUSTED_CLIENT.                                                                                                                                                                                |
| scopes                      | Do not change these values because they are specific for each client.                                                                                                                                                                                                                                |
| clientSecret                | Only set for the infoarchive.jdbc and the infoarchive.cli client.<br><br>Needs to be changed to encrypted form if the secret and password encryption is turned on. For more information, refer to the Using Encrypted Passwords in Configuration Files section in the <i>Encryption User Guide</i> . |
| accessTokenValiditySeconds  | Configures how long the access token is valid for. It is recommended that you do not change these values.                                                                                                                                                                                            |
| refreshTokenValiditySeconds | Configure how long the refresh token is valid for. It is recommended that you do not change these values.                                                                                                                                                                                            |

## InfoArchive Server and InfoArchive Web Application Communication Setup

This section describes how to set up IA Server and Gateway/IA Web App for HTTPS communication:

Browser <----- HTTPS -----> Gateway IAWA <----- HTTPS -----> IAS

First, generate or obtain certificates from some CA Root. Once you have the certificates, complete the following procedure:

1. Import the IA Server certificate into the key store.

2. Adjust the <INFOARCHIVE\_ROOT>\config\iaserver\application-ssl.yml file to point to the key store and specify the key store\* passwords.

3. Run the IA Server:

```
> bin\iaserver.bat
```

4. Import IA Server certificate into the IA Server key store.

5. Export the IA Server and IA Web App public certificate into the IA Web App truststore.

6. Adjust the <INFOARCHIVE\_ROOT>\config\iawebapp\application.yml file:

```
spring:
 application:
 name: infoarchive.gateway
 profiles:
 include:
 - infoarchive.profile.HTTPS
 - infoarchive.gateway.profile.
 AUTHENTICATION_IN_MEMORY
 cloud:
 config:
 enabled: false

infoarchive:
 gateway:
 host: localhost
 port: 8080
 contextPath: /
 token:
 secret: secret
 server:
 host: ${infoarchive.gateway.host}
 port: ${infoarchive.gateway.port}
 contextPath: ${infoarchive.gateway.contextPath}

zuul:
 routes:
 restapi:
 path: /restapi/**
 sensitiveHeaders: ""
 url: http://localhost:8080/
 addProxyHeaders: true
 host:
 socket-timeout-millis: 60000

spring:
 profiles: "infoarchive.profile.HTTPS"
server:
 ssl:
 keystore: "location of gatewayKeyStore.jks"
 keyStorePassword: abcdefg (this is a sample)
 keyPassword: abcdefg (this is a sample)
 keyStoreType: JKS
zuul:
 routes:
 restapi:
 path: /restapi/**
 sensitiveHeaders: ""
 url: "https://localhost:8080/"
```

7. Run Gateway/IA Web App:

```
> bin\iawebapp.bat
```

## Self-Signed Certificates-Based Setup

In the following scenario, the distribution has been extracted into:

c:\infoarchive

The IA Server and the Gateway/IA Web App are running on the same machine, which means that the localhost works everywhere. You will have to use the IP addresses if the IA Server and the Gateway/IA Web App are running on different machines.

The IA Server and Gatway/IA Web App must both run in HTTPS mode. You cannot run one in HTTP and the other in HTTPS mode.

1. Create the following directories:

```
> cd c:\infoarchive
> mkdir config\iaserver\https
> mkdir config\iawebapp\https
```

2. Create a self-signed IA Server certificate:

```
> cd c:\infoarchive\config\iaserver\https
> keytool -genkey -noprompt -trustcacerts -keyalg RSA -alias
IA_SERVER_CERT -dname "CN=localhost, OU=JavaSoft, O=Sun, L=Cupertino,
S=California, C=US" -keystore serverCertStore.p12
-storepass abcdefg
```

The generated file is named `serverCertStore.p12`.

3. List the self-signed IA Server certificate:

```
> cd c:\infoarchive\config\iaserver\https
> keytool -list -v -keystore serverCertStore.p12 -storepass abcdefg
-storetype PKCS12
```

4. Create the keystore for IA Server:

```
> cd c:\infoarchive\config\iaserver\https
> keytool -importkeystore -deststorepass abcdefg -destkeypass abcdefg
-destkeystore serverKeyStore.jks -srckeystore serverCertStore.p12
-srcstoretype PKCS12 -srcstorepass abcdefg -alias IA_SERVER_CERT
```

5. Export the IA Server certificate:

```
> cd c:\infoarchive\config\iaserver\https
> keytool -export -noprompt -rfc -alias IA_SERVER_CERT -file server_public_cert.cert
-keystore serverKeyStore.jks -storepass abcdefg -storetype JKS
```

The generated file is named `server_public_cert.cert`.

6. Import the IA Server certificate into Gateway/IA Web App:

```
> copy c:\infoarchive\config\iaserver\https\server_public_
cert.cert c:\infoarchive\config\iawebapp\https
> cd c:\infoarchive\config\iawebapp\https
> keytool -import -noprompt -trustcacerts -alias IA_SERVER_CERT -file
server_public_cert.cert -keystore gatewayTrustStore.jks -storepass abcdefg
```

The generated file is named gatewayTrustStore.jks.

7. List the imported IA Server certificate:

```
> cd c:\infoarchive\config\iawebapp\https
> keytool -list -v -keystore gatewayTrustStore.jks
Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entries

Alias name: server_public_cert
Creation date: Jun 10, 2016
Entry type: trustedCertEntry

Owner: CN=localhost, OU=JavaSoft, O=Sun, L=Cupertino, ST=California,
C=US
Issuer: CN=localhost, OU=JavaSoft, O=Sun, L=Cupertino, ST=California, C=US
Serial number: 35786c0
Valid from: Fri Jun 10 08:32:40 PDT 2016 until: Thu Sep 08 08:32:40 PDT 2016
Certificate fingerprints:
 MD5: F1:05:E4:6A:F4:7C:C4:A7:8A:0E:B6:DE:A9:16:DF:AA
 SHA1: CC:65:A3:74:85:C0:97:DA:95:AE:13:1F:CA:75:16:CA:66:F2:17:87
 SHA256: 22:86:56:6B:16:A1:D5:C1:66:CE:2B:A7:F5:CA:EC:74:45:C5:
 A5:3E:00:DD:F1:28:8F:F3:F8:7D:2B:00:25:A4
 Signature algorithm name: SHA256withRSA
 Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 5D 34 8D B8 34 CF 0B A0 64 05 20 93 E4 BA 92 28]4..4...d.(
0010: CD EB 20 64 .. d
]
]
```

\*\*\*\*\*  
\*\*\*\*\*

8. Create the self-signed Gateway/IA Web App certificate:

```
> cd c:\ia40\infoarchive\config\iawebapp\https
> keytool -genkey -noprompt -trustcacerts -keyalg RSA -alias IA_GATEWAY_CERT -dnname
"CN=localhost, OU=JavaSoft, O=Sun, L=Cupertino, S=California, C=US" -keypass
abcdefg -storetype PKCS12 -keystore gatewayCertStore.p12 -storepass abcdefg
```

The generated file is gatewayCertStore.p12.

9. List the self-signed Gateway/IA Web App certificate:

```
> cd c:\ia40\infoarchive\config\iawebapp\https
> keytool -list -v -keystore gatewayCertStore.p12 -storepass abcdefg -storetype
PKCS12
```

10. Create the keystore for Gateway/IA Web App:

```
> cd c:\infoarchive\config\iawebapp\https
```

```
> keytool -importkeystore -deststorepass abcdefg -destkeypass abcdefg
-destkeystore gatewayKeyStore.jks -srckeystore gatewayCertStore.p12
-srcstoretype PKCS12 -srcstorepass abcdefg -alias IA_GATEWAY_CERT
```

11. Export the Gateway/IA Web App certificate:

```
> cd c:\infoarchive\config\iawebapp\https
> keytool -export -noprompt -rfc -alias IA_GATEWAY_CERT -file gateway_public_cert.cert
-keystore gatewayKeyStore.jks -storepass abcdefg -storetype JKS
```

The generated file is named gateway\_public\_cert.cert.

12. Import the Gateway/IA Web App certificate. This step is required because Gateway is its own client for getting the group to role mapping from IA Server via the REST API from the backend:

```
> cd c:\infoarchive\config\iawebapp\https
> keytool -import -noprompt -trustcacerts -alias gateway_public_cert
-file gateway_public_cert.cert -keystore gatewayTrustStore.jks -storepass abcdefg
```

The generated file is named gatewayTrustStore.jks.

13. List the imported Gateway/IA Web App certificate:

```
> cd c:\infoarchive\config\iawebapp\https
> keytool -list -v -keystore gatewayTrustStore.jks
Keystore type: JKS
Keystore provider: SUN

Your keystore contains 2 entries

Alias name: gateway_public_cert
Creation date: Jun 10, 2016
Entry type: trustedCertEntry

Owner: CN=localhost, OU=JavaSoft, O=Sun, L=Cupertino, ST=California, C=US
Issuer: CN=localhost, OU=JavaSoft, O=Sun, L=Cupertino, ST=California, C=US
Serial number: 78b0bb1b
Valid from: Fri Jun 10 08:41:09 PDT 2016 until: Thu Sep 08 08:41:09 PDT 2016
Certificate fingerprints:
MD5: 21:21:12:82:77:83:06:B9:EC:87:93:D3:07:FD:50:22
SHA1: 56:1C:F2:D2:17:75:9A:4F:FB:F3:D9:C3:89:64:7D:29:52:4E:DC:1B
SHA256: 92:3D:F3:E3:83:98:61:6D:34:02:66:6E:2D:07:60:F6:E9:DD:
3D:BA:AD:AC:31:1C:91:39:76:85:9A:9F:C0:FD
Signature algorithm name: SHA256withRSA
Version: 3
```

Extensions:

```
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 3C 90 5E 2D CF 7E 2A 49 AB 20 DC E1 E5 2A 1E 71 <.^-..*I. ...*.q
0010: 4A 2A 83 A1 J*..
]
]
```

```


```

```
Alias name: server_public_cert
Creation date: Jun 10, 2016
```

```

Entry type: trustedCertEntry

Owner: CN=localhost, OU=JavaSoft, O=Sun, L=Cupertino, ST=California, C=US
Issuer: CN=localhost, OU=JavaSoft, O=Sun, L=Cupertino, ST=California, C=US
Serial number: 357866c0
Valid from: Fri Jun 10 08:32:40 PDT 2016 until: Thu Sep 08 08:32:40 PDT 2016
Certificate fingerprints:
 MD5: F1:05:E4:6A:F4:7C:C4:A7:8A:0E:B6:DE:A9:16:DF:AA
 SHA1: CC:65:A3:74:85:C0:97:DA:95:AE:13:1F:CA:75:16:CA:66:F2:17:87
 SHA256: 22:86:56:6B:16:A1:D5:C1:66:CE:2B:A7:F5:CA:EC:74:45:C5:
 A5:3E:00:DD:F1:28:8F:F3:F8:7D:2B:00:25:A4
 Signature algorithm name: SHA256withRSA
 Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 5D 34 8D B8 34 CF 0B A0 64 05 20 93 E4 BA 92 28]4..4...d.(
0010: CD EB 20 64 .. d
]
]
```

\*\*\*\*\*  
\*\*\*\*\*

#### 14. Import the Gateway/IA Web App certificate into the IA Server:

```

> copy c:\infoarchive\config\iawebapp\https\gateway_public_cert.cert
c:\infoarchive\config\iaserver\https
> cd c:\infoarchive\config\iaserver\https
> keytool -import -noprompt -trustcacerts -alias IA_GATEWAY_CERT -file
gateway_public_cert.cert -keystore serverTrustStore.jks -storepass abcdefg
```

The generated file is named serverTrustStore.jks.

#### 15. Configure IA Server for HTTPS mode. Edit the infoarchive/config/iaserver /application-ssl.yml file and set its content to:

```

server:
 ssl:
 keystore: "file:config/iaserver/https/serverKeyStore.jks"
 keyStorePassword: abcdefg
 keyPassword: abcdefg
 keyStoreType: JKS
 keyAlias: ia_server_cert
 trustStore: "file:config/iaserver/https/serverTrustStore.jks"
 trustStorePassword: abcdefg
 trustStoreType: JKS
 clientAuth: want
```

**Note:** clientAuth: want is critical clientAuth: need to mask the set inside the jar file.

#### 16. Run IA Server in HTTPS mode:

```

> cd c:\infoarchive
> .\bin\infoarchive-server-ssl.bat
```

17. Configure Gateway/IA Web App for HTTPS mode:

```
spring:
 application:
 name: infoarchive.gateway
 profiles:
 include:
 - infoarchive.profile.HTTPS
 - infoarchive.gateway.profile.AUTHENTICATION_IN_MEMORY
 config:
 enabled: false

infoarchive:
 gateway:
 host: localhost
 port: 8080
 contextPath: /
 token:
 secret: secret

server:
 host: ${infoarchive.gateway.host}
 port: ${infoarchive.gateway.port}
 contextPath: ${infoarchive.gateway.contextPath}

zuul:
 routes:
 restapi:
 path: /restapi/**
 sensitiveHeaders: ""
 url: http://localhost:8080/
 addProxyHeaders: true
 host:
 socket-timeout-millis: 60000

 spring:
 profiles: "infoarchive.profile.HTTPS"
 server:
 ssl:
 keystore: "file:config/webapp/https/gatewayKeyStore.jks"
 keyStorePassword: abcdefg
 keyPassword: abcdefg
 keyStoreType: JKS
 trustStore: "file:config/webapp/https/gatewayTrustStore.jks"
 trustStorePassword: abcdefg
 trustStoreType: JKS
 zuul:
 routes:
 restapi:
 path: /restapi/**
 sensitiveHeaders: ""
 url: "https://localhost:8080/"
```

18. Run IA Web App in HTTPS mode:

```
> cd c:\infoarchive
> bin\iawebapp.bat
```

Refer to the section 'Deploying IAS and IAWA with HTTPS Mode on Separate Hosts' in the *Installation Guide* for further information.

## Creating Certificates and Trust Establishment

The following sections provide examples of how to implement one-way and two-way TLS.

### One-Way TLS

When implementing one-way TLS, the server authenticates itself to clients.

To create self signed server certificate:

```
keytool -genkey -noprompt -trustcacerts -keyalg RSA -alias IA_SERVER_CERT -dname "CN=localhost, OU=JavaSoft, O=Sun, L=Cupertino, S=California, C=US" -keystore serverCertStore.p12 -storepass abcdefg -storetype PKCS12
```

To display server certificates in a keystore:

```
keytool -list -v -keystore serverCertStore.p12 -storepass abcdefg -storetype PKCS12
```

To create a public server certificate for other applications:

```
keytool -export -noprompt -rfc -alias IA_SERVER_CERT -file server_public_cert.cert -keystore serverCertStore.p12 -storepass abcdefg -storetype PKCS12
```

To import a certificate into a client keystore:

```
keytool -import -noprompt -trustcacerts -alias server_public_cert -file server_public_cert.cert -keystore client_truststore.jks -storepass abcdefg
```

Add the following properties to the server's application.yml file:

```
server:
 ssl:
 keystore: "serverCertStore.p12"
 keyStorePassword: abcdefg
 keyPassword: abcdefg
 keyStoreType: PKCS12
 keyAlias: ia_server_cert
```

### Two-Way TLS

When implementing two-way TLS, the server and client authenticate themselves to each other.

To create a self-signed client certificate:

```
keytool -genkey -noprompt -trustcacerts -keyalg RSA -alias IA_CLIENT_CERT -keystore clientCertStore.p12 -storepass abcdefg -storetype PKCS12
```

To display client certificates in a keystore:

```
keytool -list -v -keystore clientCertStore.p12 -storepass abcdefg -storetype PKCS12
```

To create a public client certificate for other applications:

```
keytool -export -noprompt -rfc -alias IA_CLIENT_CERT -file client_public_cert.cert
-keystore clientCertStore.p12 -storepass abcdefg -storetype PKCS12
```

To import a certificate into a server keystore:

```
keytool -import -noprompt -trustcacerts -alias client_public_cert -file
client_public_cert.cert -keystore server_truststore.jks -storepass abcdefg
```

Add the following properties to the server's application.yml file:

```
server:
 ssl:
 keystore: "serverCertStore.p12"
 keyStorePassword: abcdefg
 keyPassword: abcdefg
 keyStoreType: PKCS12
 keyAlias: ia_server_cert
 trustStore: "server_truststore.jks"
 trustStorePassword: abcdefg
 trustStoreType: JKS
 clientAuth: need

client:
 ssl:
 trust-store: "location of gatewayTrustStore.jks"
 trust-store-password: abcdefg (this is a sample)
 trust-store-type: JKS
 # Following entries is required for two-way TLS authentication
 key-store: "location of gatewayKeyStore.jks"
 key-store-password: abcdefg (this is a sample)
 key-store-type: JKS
```

## Working with the Server's application.yml File

This section explains what you can and cannot update in the server's <INFOARCHIVE\_ROOT>/config/iaserver/application.yml file.

## Security Profile

The data for the security profile can be updated.

```
spring:
 application:
 name: infoarchive.ias
 profiles:
 include: infoarchive.ias.profile.JWT
 # include: infoarchive.ias.profile.NO_AUTHENTICATION
```

## logging.level

The data for the logging.level can be updated.

```
log.level.threshold: 'WARN'
```

The server should typically be running in WARN level, as that limits the amount of information logged in the system (saves disk space, helps performance, etc.). If there are issues in the system, the logging level should be raised (for example, to DEBUG level). IA Server needs to be restarted for this change to take effect.

## infoarchive Section

The data for the infoarchive section can be updated.

The value of infoarchive.gateway.token.secret should be exactly the same as it is set in the IA Web App application.yml file. Both the IA Server and the IA Web App client need to have this value synchronized to the same value to ensure that JWT tokens can be decrypted correctly.

The collections section controls the defaultPage. When displaying collections through pagination, you typically start with first page, which is called the 'defaultPage'. Collections are zero-indexed, meaning that the first page is at index '0'. If, for some reason, you want to always start at the second page, set the defaultPage to '1' and so on. For most environments, however. this value should be set to '0'.

Collections are paginated. The value for defaultPageSize controls the size of the page. The following example indicates pages of 10 items (each). You can adjust this value, as needed.

**Note:** Modifying the defaultPageSize value impacts the real estate of the pages in IA Web App. This should be properly tuned to ensure all clients can handle a given page size.

The search.defaultTimeOutMs indicates the amount of time a synchronous search is allowed to run before the server will interrupt it and return an error. The value is in milliseconds (MS). In the following example, the value of '8000' indicates a default time out of eight seconds. If the search does not complete, it will be interrupted in eight seconds and the server will return an error. To handle this, the user should try re-running the search in asynchronous mode as a background search.

```
infoarchive:
 gateway:
 token:
 secret: secret
 rest:
 collections:
 defaultPage: 0
 defaultPageSize: 10
 search:
 defaultTimeOutMs: 8000
```

## managedItemData Section

This section needs to be updated if you want the managedItemData in a separate federation.

The data for the managedItemData section can be updated.

```
managedItemData:
 xdb:
 dataNode:
 storeStackTraceInLock: false
 name: mainFederation
 bootstrap: xhive://localhost:8080
 superuser:
 password: test
 database:
 name: managedItemDatabase
 admin:
 password: secret
```

## defaultNames Section

The defaultNames value is for future use and should not be modified.

```
tenant:
 defaultNames:
 - INFOARCHIVE
```

## Storage Paths and OAIS Sections

It is recommended that you do not change the storage paths or the OAIS sections.

```
storage:
 fileSystemRoots:
 -
 name: defaultFileSystemRoot
 description: Default FileSystemRoot
 path: data/root

#locations on the IA Server's local file system
localStorage:
 tempFolder: data/temp

oais:
 defaultReceiverNode: receiver_node_01
 defaultIngestNode: ingest_node_01
```

## job Section

Do not change the data in the job section. In particular, any change made to the readOnly flag for jobs will cause the jobs to malfunction.

```
job:
 definitions:
 -
 name: Commit
```

```
handlerName: CommitJob
description: Commits packages in waiting commit
readOnly: false
applicationScoped: false
tenantScoped: false
maxAttempts: 1
expirationInterval: 60000
rescheduleInterval: 60000
```

## roles Section

Do not update anything in this section.

## Updating the Working Directory

Define the properties of the working directory in the `localStorage` entry of the `application.yml` file. Using the `application.yml` file not only allows you to easily set a different working directory, but also simplifies application configuration.

1. Before updating the working directory, stop all services.
2. Access the `<INFOARCHIVE_ROOT>/config/iaserver/application.yml` file.

The following information is displayed:

```
#locations on the IA Server's local file system
localStorage:
 ..tempFolder: data/temp
```

3. Update the working directory, as desired and save your changes.
4. Restart the services.

## Default File System Root

The IA Server does not create a default file system root if nothing is configured. The `application.yml` file, however, allows for a "fileSystemRoot" entry to be used by IA Server. The specified path has to pre-exist on the file system. IA Server will not create it.

You are able to define a default file system root through the `<INFOARCHIVE_ROOT>\config\iaserver\application.yml` file.

## System and Audit Database

The system database is where all system objects live (tenants, applications, spaces, searches, holdings, tables, etc.).

Audit data is where audit records are stored until they are SIP-ified and archived.

The system database and audit data are both configured in the `.yml` file.

The following is from the application.yml for a system database:

```
Settings for the System User, overall xDB caches, and SystemData repository
system:
 # Maximum number of objects held in cache
 objectCacheSize: 100

 userName: system

 pageSize: 4096

 # Percentage of heap used for xDB cache
 cacheSizeTotal: 50

 # Percentage of cache use for SystemData
 cacheSizeSystem: 50

 # Retention of Rollforward objects, in hours
 rollForwardObjectRetention: 24

SystemData repository
xdb:
 dataNode:
 storeStackTraceInLock: false
 name: mainFederation
 bootstrap: xhive://localhost:2910
 superuser:
 password: test
 database:
 name: mainDatabase
 admin:
 password: 1c742386-a50e-478d-84c2-918329f95f50
 rollForwardDatabase:
 name: rollForwardDatabase
 admin:
 password: 8612ac1a-7cb9-48df-a9fd-c05869606733
 synchronizationDatabase:
 name: synchronizationDatabase
 admin:
 password: d493b068-c0c8-42ec-9370-c0402b05b718
```

The following is from the application.yml for audit data:

```
auditData:
 # Maximum number of objects held in cache
 objectCacheSize: 100

 xdb:
 dataNode:
 storeStackTraceInLock: false
 name: mainFederation
 bootstrap: xhive://localhost:2910
 superuser:
 password: test #GENERATED_PASSWORD_auditData_xdb_dataNode_superuser
 _password_REFERENCE
 database:
 name: auditDatabase
 admin:
 password: 987968b3-e71f-4bc8-8c5b-3e400f460103 #GENERATED_PASSWORD_
 auditData_xdb_database_admin_password_REFERENCE
```

## Configuring the Number of Items Listed in the Search Results

The defaultPageSize value in the `application.yml` indicates the number of result items that appear in the **Record Search** tab. The default value is 10.

In the following example, the user is changing the default value to 5.

1. Before updating the defaultPageSize, stop all services.
2. Access the `<INFOARCHIVE_ROOT>/config/iaserver/application.yml` file.

The following information is displayed:

```
infoarchive
...rest:
.....collections
.....defaultPage: 0
.....defaultPageSize: 10
.....search:
.....defaultTimeOutMs: 8000
```

3. Change the defaultPageSize value to the desired value (in the following example, the desired number of result items is 5):

```
infoarchive
...rest:
.....collections
.....defaultPage: 0
.....defaultPageSize: 5
.....search:
.....defaultTimeOutMs: 8000
```

4. Save the change.
5. Restart the services.

## Configuring the Time Limit for a Background Search

If a search exceeds the time limit specified in the `application.yml` file, it automatically becomes a background search. The default time limit set is set to 8,000 milliseconds (8 seconds). You are, however, able to change the time limit.

In the following example, the user is changing the time limit from the default to 5,0000 milliseconds.

1. Before updating the time limit, stop all services.
2. Access the `<INFOARCHIVE_ROOT>/config/iaserver/application.yml` file.

The following information is displayed:

```
infoarchive
...rest:
.....collections
.....defaultPage: 0
.....defaultPageSize: 5
.....search:
.....defaultTimeOutMs: 8000
```

3. Change the `defaultTimeOutMs` value to the desired time (in the following example, the desired time is 5000):

```
infoarchive
```

```
...rest:
.....collections
.....defaultPage: 0
.....defaultPageSize: 5
.....search:
.....defaultTimeOutMs: 5000
```

4. Save the change.
5. Restart the services.

## Configuring the xDB Segment Size for a Table Archive

An entry in the `application.yml` file allows you to configure the xDB segment size for a table archive.

The value of the `maxSegmentSize` field indicates the maximum segment size (in bytes) for a table archive. If the segment size exceeds this value, a new segment is created. If the `maxSegmentSize` is set to '0', a segment will grow indefinitely.

The default value of the `maxSegmentSize` is 100000000000 bytes (10 GB).

## Limiting the Size of Files Transferred Through REST

The IA Server `application.yml` file allows you to limit the size of files that are transferred through REST during reception.

The following is the section in the `application.yml` that can be updated:

```
Spring settings for multipart requests
multipart:
 enabled: true
 maxFileSize: 2048MB
 maxRequestSize: 2048MB
```

The above values must match the equivalent values in the `<INFOARCHIVE_ROOT>/config/iawebapp/application.yml` file.

The following explains the Spring Boot properties:

| Property                              | Description                                                          |
|---------------------------------------|----------------------------------------------------------------------|
| <code>multipart.maxFileSize</code>    | Specifies the maximum size permitted for uploaded files.             |
| <code>multipart.maxRequestSize</code> | Specifies the maximum size allowed for multipart/form-data requests. |
| <code>multipart.enabled</code>        | Enables the support of multipart uploads.                            |

Specify the values using long values or using more readable variants that accept KB or MB suffixes.

## Working with the xdb.properties File

This file allows you to configure the xDB memory settings.

By default these settings are commented out. Use these settings to set the memory to another value than the default value.

The following memory settings are used by a single command line tool, as in the `-Xmx` parameter to the JVM. For example, these settings are used when running the xDB Admin client.

```
XHIVE_MAX_MEMORY, default value: 256M
XHIVE_MIN_MEMORY, default value: 32M
```

Memory settings for the page server process (`-Xmx`):

```
XHIVE_SERVER_MAX_MEMORY, default value: 4G
XHIVE_SERVER_MIN_MEMORY, default value: 256M
```

## Running Chain-of-Custody Tests for Table Archives

Chain of custody tests check the integrity of the ingested tables in InfoArchive. They are configurable with respect to which tests users want to run. They can be configured pre- or post-data ingestion via an XML file. Refer to the following files in the `<INFOARCHIVE_ROOT>/config/iashell` directory to review two examples of such files: of the IA distribution, named as:

- `chain-of-custody-table.xml`
- `chain-of-custody-schema.xml`

Users can POST the XML file containing the tests' configuration on the REST link contained in the response of the retrieved table (desired table on which these tests need to be executed against). Refer to the Developer's REST Guide for further information. This will return the result back to the users. The tests rely on the information given in the `metadata.xml` of the tables provided by the users. If any discrepancy is found between the aforementioned metadata and the ingested tables after running the chain of custody tests, InfoArchive reports it to the users. Refer to the following table to review the various tests that can be used:

| Test               | Description                                                                                                                                                                         |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Table Count        | Counts the tables that have been loaded into the system and compares the count to the one stated in the <code>metadata.xml</code> file. The test fails if the results do not match. |
| Row Count          | Verifies that the number of records for each table matches the row count, as reflected in the table's <code>metadata.xml</code> file.                                               |
| Metadata Row Count | Verifies that the number of records for each table is stated in the <code>metadata.xml</code> file associated with the table (or table metadata).                                   |

| Test                                                  | Description                                                                                                                                                                                                                                                                        |
|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Test Table Document Containing Table Name and Rows    | Inspects each table document within a table library. The test validates that the document has the name of the table at the proper place with respect to the predefined document structure of the table document. It also verifies that the table document contains row element(s). |
| Trailing Spaces                                       | Inspects table data for trailing spaces. This test is configurable to test the first number of rows in each data file. If any of the data files contain trailing spaces, the test will fail.                                                                                       |
| Leading Spaces                                        | Inspects table data for leading spaces. This test is configurable to test the first number of rows in each data file. If any of the data files contain leading spaces, the test will fail.                                                                                         |
| Table in Data Library is Also in metadata File        | Verifies if every table document in the table data library includes its table metadata.xml. If the table metadata does not exist, the test will fail.                                                                                                                              |
| Table in metadata File has at Least One Data Document | Verifies if every table in the metadata.xml file also has at least one data document. If one of the metadata.xml files does not contain at least one data document, the test will fail.                                                                                            |
| Table metadata and its Data Library                   | Inspects if there is a data library of the table, as stated in its metadata.xml. If a data library cannot be found for a table, the test will fail.                                                                                                                                |
| Schema metadata                                       | Verifies that the schema metadata does not contain table metadata. If there is no table metadata of the schema, the test will fail.                                                                                                                                                |
| Metadata Column Count                                 | Verifies that information about table columns is referenced in the table metadata.xml. If no column information is found in the metadata.xml, the test will fail.                                                                                                                  |
| Metadata Column Type                                  | Verifies whether the type information about the column is stated in the table metadata.xml. If this information is not stated in the metadata.xml, the test will fail.                                                                                                             |
| Column Count                                          | The number of columns of each row are counted and validated against the table metadata.xml column counts. If the counts do not match, the test will fail.                                                                                                                          |
| Data Types                                            | Verifies that the table data matches the data types as specified in the table metadata.xml.                                                                                                                                                                                        |

## Using InfoArchive's Batch Processing Functionality

Batch processing is used to improve the performance of the following long-running operations and jobs, including:

- Applying a hold to an application, table, package, or one or more records in a search result.
- Removing a hold set or removing items from a hold set.

- Applying a hold to records via the Apply Hold Rule to Records job.
- Applying a retention policy to records via the Apply Retention Rule to Records job. This job does not use batch processing.

With batch functionality, an operation is broken into smaller chunks. Even if there is a small number of items to process, at least one batch is created.

If one of the batches fails to be processed, the option to retry the operation is available for the applying a hold operation. The ability to retry an operation is not available for the remove hold operation. The system only restarts the batches that failed and, therefore, recovery is quicker than doing the operation from scratch.

Furthermore, if the server is taken down for maintenance while a batch is running, when the server comes back up, the batches will automatically be retried without user intervention.

The following jobs allow you to manually retry upon failure:

- Generate Purge Candidate List
- Dispose Purge Candidate List
- Apply Retention Policy to Records (this does not support batch processing but does support retry)
- Clean up Purge Candidate Lists and Applications
- Apply Retention Rule to Records
- Apply Hold Rule to Records
- Process Retention Events

The following chart lists all compliance-related operations that are candidates for batch processing.

## Configuring the Batch Size

Configure the following properties in the <INFOARCHIVE\_ROOT>\config\iaserver\application.yml file:

```
configuration for batch sizes using batching framework
the disposition value is not for batch but controls transaction size
batch:
 size:
 applyHold: 1000
 removeHold: 1000
 applyRetention: 1000
 removeRetention: 1000
 disposition: 1000
```

| Property        | Description                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| applyHold       | When applying a hold to a large number of records, this property indicates the default batch size used (the maximum number of records in a batch). There are two methods when this property comes into play: <ul style="list-style-type: none"> <li>• When applying a hold directly to search results.</li> <li>• When applying a hold using the Apply Hold Rule to Records job.</li> </ul> |
| removeHold      | When removing a hold set that contains a lot of records, this property indicates the maximum number of records in a batch.<br><br>It is recommended that the batch size for this property be set to 1000 or more. When the remove hold operation is applied to a large hold set, creation of the batches will take a long time if this property is set too low.                             |
| applyRetention  | When applying a retention policy to a large number of records, this property indicates the default batch size used (the maximum number of records in a batch). This property comes into play when you apply a retention policy with the Apply Retention Rule to Records job.<br><br>The Apply Retention Policy to Records job does not use the batch functionality.                         |
| removeRetention | Used by the Remove Policy Job for batches.                                                                                                                                                                                                                                                                                                                                                  |
| disposition     | This property is being used by the Dispose Purge Candidates List job.                                                                                                                                                                                                                                                                                                                       |
| processEvents   | Used by the Process Events job.                                                                                                                                                                                                                                                                                                                                                             |
| tableIndex      | Used by the Table index job.                                                                                                                                                                                                                                                                                                                                                                |

## Viewing Batch and Log Information

The following example demonstrates how you can view the batch information and logs for a particular job instance. This scenario involves the successful run of the Apply Hold Rule to Records job:

1. When the **Last Run Status** column states **SUCCESS**, click the Apply Hold Rule to Records link.
2. In the **Status** column, click **SUCCESS**.  
A new screen is displayed.
3. In the **Status** column, click **COMPLETE**.

Now you are able to toggle between the batch information and log for this particular job instance. You can also download the diagnostics log.



# Chapter 3

---

## Declarative Configuration

### Configuring a SIP Archive

Applications in InfoArchive provide access to all archived data. Each application represents a single decommissioned or active archive. Typically, most users will access applications via IA Web App. It is also possible, however, to access applications via the REST API, which allows customers to create a custom interface to interact with the applications.

This section illustrates how to configure a new SIP archive with one holding. By the end of the section, you will be able to ingest data, perform a search and retrieve content from IA Web App.

Before you begin, you need to:

- Know what you want to archive and to have defined and generated an XML schema.
- Know what you want to search
- Know how many SIPs per day you expect to have
- Know how many AIUs per SIP you expect to have
- Know the retention date range. How long data should be stored
- Know the average volume of the SIP package (in Mb)
- Know the average volume of the pdi.xml inside the SIP package (in Mb)

This section illustrates how to properly configure:

- [PDI](#)
- [Indexes](#), including:
  - [PDI.INDEX.CREATOR](#)
- [PDI.AIU.CNT](#)
- [PDI.AIU.ID](#)
- [Partition keys](#)
- [RI.INIT](#)
- [A holding](#)

Refer to Improving [Performance for a SIP Archive](#) in Appendix B for more information.

## Configuring PDI

The following example procedure uses a fictitious holding named Tweets, which contains messages from Twitter. Each tweet contains some metadata (text, screen name, date, etc.) and can be associated with 0 to n entities (URL, photo, hashtag, etc.).

The following is the PDI used in the scenario:

```
<tweets xmlns="urn:x-emc:ia:demo:schema:tweets:1.0">
 <tweet>
 <id>731398089370669056</id>
 <text>RT @EMCbigdata: Recorded live at #EMCWORLD: a #BigData podcast
 with our partner @Splunk via #EMCElect https://t.co/FI14e4mmgF
 https://t.co/...</text>
 <createdAt>2016-05-14T10:17:47+02:00</createdAt>
 <retweet>true</retweet>
 <retweetCount>7</retweetCount>
 <user>
 <id>729669013593333760</id>
 <name>Ruth Lutterloch</name>
 <location>London, England</location>
 <screenName>RuthLutterloch</screenName>
 </user>
 <mediaEntity>
 <id>730819244519264256</id>
 <url>http://pbs.twimg.com/media/CiRkOxiXEAAuEh.jpg</url>
 <type>photo</type>
 <start>126</start>
 <end>140</end>
 </mediaEntity>
 <urlEntity>
 <text>https://t.co/FI14e4mmgF</text>
 <start>102</start>
 <end>125</end>
 <url>https://t.co/FI14e4mmgF</url>
 <expandedUrl>http://emc.im/6012Boc5k</expandedUrl>
 <displayUrl>http://emc.im/6012Boc5k</displayUrl>
 </urlEntity>
 <hashtagEntity>
 <text>EMCWORLD</text>
 <start>33</start>
 <end>42</end>
 </hashtagEntity>
 <hashtagEntity>
 <text>BigData</text>
 <start>46</start>
 <end>54</end>
 </hashtagEntity>
 <hashtagEntity>
 <text>EMCElect</text>
 <start>92</start>
 <end>101</end>
 </hashtagEntity>
 </tweet>
</tweets>
```

The following is the PDI schema used in the scenario:

```
<?xml version="1.0" encoding="UTF-8"?>
<xss:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault=
"qualified" targetNamespace="urn:x-emc:ia:demo:schema:tweets:1.0"
xmlns:ns1="urn:x-emc:ia:demo:schema:tweets:1.0">
 <xss:element name="tweets">
 <xss:complexType>
```

```

<xs:sequence>
 <xs:element maxOccurs="unbounded" ref="ns1:tweet"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="tweet">
 <xs:complexType>
 <xs:sequence>
 <xs:element ref="ns1:id"/>
 <xs:element ref="ns1:text"/>
 <xs:element ref="ns1:createdAt"/>
 <xs:element ref="ns1:retweet"/>
 <xs:element ref="ns1:retweetCount"/>
 <xs:element ref="ns1:user"/>
 <xs:element minOccurs="0" maxOccurs="unbounded" ref="ns1:mediaEntity"/>
 <xs:element minOccurs="0" maxOccurs="unbounded" ref="ns1:urlEntity"/>
 <xs:element minOccurs="0" maxOccurs="unbounded" ref="ns1:hashtagEntity"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="createdAt" type="xs:dateTime"/>
<xs:element name="retweet" type="xs:boolean"/>
<xs:element name="retweetCount" type="xs:integer"/>
<xs:element name="user">
 <xs:complexType>
 <xs:sequence>
 <xs:element ref="ns1:id"/>
 <xs:element ref="ns1:name"/>
 <xs:element ref="ns1:location"/>
 <xs:element ref="ns1:screenName"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="name" type="xs:string"/>
<xs:element name="location" type="xs:string"/>
<xs:element name="screenName" type="xs:NCName"/>
<xs:element name="mediaEntity">
 <xs:complexType>
 <xs:sequence>
 <xs:element ref="ns1:id"/>
 <xs:element ref="ns1:url"/>
 <xs:element ref="ns1:type"/>
 <xs:element ref="ns1:start"/>
 <xs:element ref="ns1:end"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="type" type="xs:NCName"/>
<xs:element name="urlEntity">
 <xs:complexType>
 <xs:sequence>
 <xs:element ref="ns1:text"/>
 <xs:element ref="ns1:start"/>
 <xs:element ref="ns1:end"/>
 <xs:element ref="ns1:url"/>
 <xs:element ref="ns1:expandedUrl"/>
 <xs:element ref="ns1:displayUrl"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="expandedUrl" type="xs:anyURI"/>
<xs:element name="displayUrl" type="xs:anyURI"/>
<xs:element name="hashtagEntity">
 <xs:complexType>
 <xs:sequence>

```

```
<xs:element ref="ns1:text"/>
<xs:element ref="ns1:start"/>
<xs:element ref="ns1:end"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="id" type="xs:integer"/>
<xs:element name="text" type="xs:string"/>
<xs:element name="url" type="xs:anyURI"/>
<xs:element name="start" type="xs:integer"/>
<xs:element name="end" type="xs:integer"/>
</xs:schema>
```

1. Prepare your environment by completing the following:

- a. You can choose to either:
  - Define an application from scratch;
  - Make a copy of the application that resembles one you want to create (for example, use the PhoneCalls sample application to use as a template for the new configured application); or
  - Use the [Holding wizard](#) to create an application using the InfoArchive web interface, export the result in the final step, and then use the exported file as a starting point. If this is the starting point, then all of the configuration will be placed in a single YAML file, and references to external files, as described in the remaining sections, will appear (and should be edited) in that single YAML file.
- b. The folder of your application may be placed anywhere, each of the sample applications in <INFOARCHIVE\_ROOT>/examples/applications contain some install scripts that perform the configuration and ingestion of the sample into InfoArchive. If those script files are copied to your new application, the path locations in install.bat and/or install should be updated

The following example procedure uses a fictitious holding named **Tweets**:

- c. If the starting point for the new application is an existing application, start with updating all of the names in the configuration.yml file (for the sample applications, this file resides in the config subdirectory). For example, replace all of the occurrences of PhoneCalls with Tweets.
2. Place the new pdi schema XSD file in the data-model-config subdirectory of your application, and then refer to it in the configuration.yml file. For example

```
pdiSchema:
 name: urn:eas-samples:en:xsd:tweets.1.0
 content:
 format: xsd
 resource: data-model-config/pdiSchema-tweets.xsd
```

3. Create or update the pdi xml file referenced from the pdi section in the configuration.yml file. For example:

```
pdi:
 name: Tweets-pdi
 content:
 format: xml
 resource: data-model-config/pdi-Tweets.xml
```

This XML file contains the configuration that will be used during the ingestion process. In this file, you can:

- Configure the xDB indexes to put on the PDI XML file,
- Extract the partition keys,
- Count the number of AIUs in the PDI, and
- Define where to find the CI.

It is possible to create a new file, or base it on one of the existing samples.

4. Set up the transformation. This object describes how you want to transform the PDI XML when you want to generate an analytic rendition and/or a disposition export.

In the following example of the 250-transformation.xml file, we indicate the XQuery to execute to generate the format of the `rendition.csv.gzip` file when we request the result schema: `urn:x-emc:ia:schema:pdi:export`:

```
transformation:
 name: PhoneCalls-transformation-csv
 format: rendition.csv.gzip
 inputSchema:
 - urn:eas-samples:en:xsd:phonecalls.1.0
 - urn:eas-samples:en:xsd:phonecalls.1.0:crypto
 resultSchema: urn:x-emc:ia:schema:pdi:export
 xquery:
 namespaces:
 - n
 resource: exports/transformation-Tweets.xq
```

The XQuery is defined in a separate file, and contains the following content:

```
declare namespace n = "${pdiSchema}";
declare function local:replace($i as xs:string?) {
 if ($i) then ''' || replace($i,'','','') || ''' else '''
};
declare function local:add-row($input) {string-join
($input,',') || '
'};
(
(: Header :)
local:add-row((
'createdAt',
'userName',
'userScreenName',
'text',
'hashtags'
))

,
(: Rows :)
for $call in /n:tweets/n:tweet
return
local:add-row(
(
local:replace($call/n:createdAt),
local:replace($call/n:userName),
local:replace($call/n:userScreenName),
local:replace($call/n:text),
local:replace(string-join($call/n:hashtagEntity/n:text,','))
))
```

Before disposing the AIP, it is possible to export the metadata with a background export function. This option is available if one transformation object with the `urn:x-emc:ia:schema:pdi:export` result schema is found.

To generate an analytic rendition during the ingestion, add the following configuration into the PDI and specify the desired result schema:

```
<data>
<id>pdi.transformer</id>
<result.schema>urn:x-emc:ia:schema:pdi:export</result.schema>
</data>
```

If you do not need these features, remove the PDI configuration part.

## Configuring Indexes

An index is computed during ingestion and covers the data contained in one AIP. Indexes are used in the second tier of a search, whereby the system scans packages for individual results (AIUs) via the use of indexes.

An AIP can have many indexes defined to satisfy different search criteria.

For table archive applications, indexing needs to be enabled at the column level in the metadata files that for the sample applications reside as XML files in `config` directories. Multi-path indexes are created at the end of the ingestion process,

Refer to the *Encryption Guide* to learn about the indexing of encrypted fields.

## Types of Indexes

There are two types of indexes:

- A path value index (`path.value.index`) is the most common of the two index types. It indexes the value of elements and attributes. Furthermore, the values of multiple elements or attributes can be used in the key to create a composite index.
- A full-text index indexes the values of elements and attributes but tokenizes the values into a number of terms, and each term or element combination is added to the index. While consuming more storage, a full-text index enables users to search for an individual word contained in the indexed values. It also allows for the use of wildcards in a search and is less sensitive to misspelling entered as search criteria.

Path Value Indexing	MultiPath Indexing
It can be used for indexing multiple elements, but requires every single element to be explicitly listed in the index definition.	Multipath indexes allow you to specify sub-paths with wildcards that will match more than one element path, so not every element has to be explicitly listed. Making multipath indexes much more flexible and easy to use.
Smaller size means that it is faster to ingest	Large index size

Path Value Indexing	MultiPath Indexing
Better performance if you know the query ahead of time along with the number of predicates	Only option for table archiving
B-tree index	Lucene Inverted index

## Configuring Indexing

The following outlines the information contained in an index in the pdi.xml file:

```
<data>
 <id>pdi.index.creator</id> (1.)
 <key.document.name>xdb.pdi.name</key.document.name>
 <indexes>
 ...
 </indexes>
</data>
```

1. Indicates the ID of the index creator processor, which must be pdi.index.creator.
2. Indicates the indexes to create, path.value.index or full.text.index.

## Using a Path Value Index

The path value index is defined to index the AIU based on a value:

PATH\_TO\_AIU[CONDITION<TYPE>]

The following is an example of the PDI:

```
<objects>
 <object>
 <customerid>abc123</customerid>
 ...
 </object>
</objects>
```

The entity path is /{ns}objects/{ns}object.

To create an index, for example, on the value 'customer id', you would use the following:

/ {ns}objects/{ns}object[{ns}customerid<STRING>]

The following is an example of a path value index:

```
<path.value.index>
 <name>tweetId</name>
 <path>
 /{urn:x-emc:ia:demo:1.0}objects/{urn:x-emc:ia:demo:1.0}object
 [{ur n:xemc:ia:demo:1.0}]id<STRING>
 </path>
 <compressed>false</compressed>
 <unique.keys>false</unique.keys>
 <concurrent>false</concurrent>
 <build.without.logging>true</build.without.logging>
</path.value.index>
```

The following is an example of a full-text index:

```
<full.text.index>
<name>text-fulltext</name>
<compressed>false</compressed>
<concurrent>false</concurrent>
<optimize.leading.wildcard.search>true</optimize.leading.wildcard.search>
<index.all.text>true</index.all.text>
<include.attributes>false</include.attributes>
<support.phrases>false</support.phrases>
<support.scoring>false</support.scoring>
<convert.terms.to.lowercase>true</convert.terms.to.lowercase>
<filter.english.stop.words>false</filter.english.stop.words>
<support.start.end.token.flags>false</support.start.end.token.flags>
<element.uri>urn:x-emc:ia:demo:schema:tweets:1.0</element.uri>
<element.name>text</element.name>
<attribute.uri/>
<attribute.name/>
</full.text.index>
```

## Creating Path Value Indexing for a Table Archive

Table archiving only supports multi-path value indexing. To save time, the Developer can use a path value index for specific search queries in table archiving, and is able to specify the path value in the configuration for selected fields during or after ingestion. Then, after the ingestion of data, the Developer:

- Has path value indexes created for specified fields; or
- Can trigger path value indexing manually when appropriate (for example, after all data is ingested and verified).

The following is an example of a path value index:

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata>
<defaultSchema>SAKILA</defaultSchema>
<schemaMetadataList>
<schemaMetadata>
<name>sakila</name>
<tableCount>1</tableCount>
<tableMetadataList>
<tableMetadata>
<name>actor</name>
<recordCount>4</recordCount>
<columnList>
<column>
<name>ACTOR_ID</name>
<ordinal>1</ordinal>
<type>INTEGER</type>
<typeLength>32</typeLength>
</column>
...
</columnList>
<pathValueIndexList>
<pathValueIndex>
<name>optional</name>
<uniqueKey>false</uniqueKey>
<column>ACTOR_ID</column>
<column>FIRST_NAME</column>
<column>LAST_NAME</column>
<fulltext>
<column>LAST_NAME</column>
```

```

<lowercase>true</lowercase>
</fulltext >
</pathValueIndex>
</pathValueIndexList>
<anotherIndexList>
<anotherIndex>
...
</anotherIndex>
</anotherIndexList>
</tableMetadata>
</tableMetadataList>
</schemaMetadata>
</schemaMetadataList>
</metadata>

```

It results to the following index definition: /sakila/actor/ROW[ACTOR\_ID<INTEGER> + FIRST\_NAME<STRING> + LAST\_NAME<SA\_Adjust\_To\_LowerCase>]

The following is the XSD definition for a table element:

```

<xs:complexType name="tableMetadataType">
<xs:sequence>
<xs:element type="xs:string" name="name" minOccurs="1" maxOccurs="1"/>
<xs:element type="xs:int" name="recordCount" minOccurs="0" maxOccurs="1"/>
<xs:element name="columnList" minOccurs="0" maxOccurs="1">
<xs:complexType>
<xs:sequence>
<xs:element type="columnType" name="column" minOccurs="1" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="pathValueIndexList" minOccurs="0" maxOccurs="1">
<xs:complexType>
<xs:sequence>
<xs:element type="pathValueIndexType" name="pathValueIndex" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:sequence>
<xs:complexType name="pathValueIndexType">
<xs:sequence>
<xs:element type="xs:string" name="name" minOccurs="0" maxOccurs="1"/>
<xs:element type="xs:string" name="uniqueKey" minOccurs="0" maxOccurs="1"/>
<xs:element type="xs:string" name="column" minOccurs="1" maxOccurs="unbounded"/>
<xs:element name="fulltext" minOccurs="0" maxOccurs="1">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:string" name="column" minOccurs="1" maxOccurs="1"/>
<xs:element type="xs:boolean" name="lowercase" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

The optional path value index definition:

```

<pathValueIndexList>
<pathValueIndex>
<name>optional name if absents then it is named by IA server</name>
<uniqueKey>false</uniqueKey> // optional
<column>ACTOR_ID</column> // LIST of the table columns participating in the index
<column>FIRST_NAME</column>

```

```
<column>LAST_NAME</column>
<fulltext> // optional
<column>LAST_NAME</column> // POINTS to one of the above columns (ACTOR_ID,
FIRST_NAME or LAST_NAME)
<lowercase>true</lowercase>
</fulltext>
</pathValueIndex>
</pathValueIndexList>
```

## PDI.INDEX.CREATOR

Create an xDB index on each search criterion. In the majority of use cases, the path value index is the best candidate. A full-text index can be defined if you want to use a contains operator and/or case insensitive search. The full-text index is less efficient than a path value index.

The AIU path is followed by the element path you want to search with the type (STRING, INTEGER, LONG, DOUBLE, DATE, DATE\_TIME). For example:

```
{urn:x-emc:ia:demo:schema:tweets:1.0}tweets/{urn:x-emc:ia:demo:schema:
tweets:1.0}tweet[{urn:x-emc:ia:demo:schema:tweets:1.0}createdAt<DATE_TIME>]
```

The following is an example of the pdi.index.creator file:

```
<data>
 <id>pdi.index.creator</id>
 <key.document.name>xdb.pdi.name</key.document.name>
 <indexes>
 <path.value.index>
 <name>createdAt</name>
 <path>
 /{urn:x-emc:ia:demo:schema:tweets:1.0}tweets/{urn:x-emc:ia:
demo:schema:tweets:1.0}tweet[{urn:x-emc:ia:demo:schema:tweets:1.0}createdAt<DATE_TIME>]
 </path>
 <compressed>false</compressed>
 <unique.keys>false</unique.keys>
 <concurrent>false</concurrent>
 <build.without.logging>false</build.without.logging>
 </path.value.index>
 <path.value.index>
 <name>userName</name>
 <path>
 /{urn:x-emc:ia:demo:schema:tweets:1.0}tweets/{urn:x-emc:ia:
demo:schema:tweets:1.0}tweet[{urn:x-emc:ia:demo:schema:tweets:1.0}user/
{urn:x-emc:ia:demo:schema:tweets:1.0}name<STRING>]
 </path>
 <compressed>false</compressed>
 <unique.keys>false</unique.keys>
 <concurrent>false</concurrent>
 <build.without.logging>false</build.without.logging>
 </path.value.index>
 <path.value.index>
 <name>userScreenName</name>
 <path>
 /{urn:x-emc:ia:demo:schema:tweets:1.0}tweets/{urn:x-emc:ia:
demo:schema:tweets:1.0}tweet[{urn:x-emc:ia:demo:schema:tweets:1.0}user/
{urn:x-emc:ia:demo:schema:tweets:1.0}screenName<STRING>]
 </path>
 <compressed>false</compressed>
 <unique.keys>false</unique.keys>
 <concurrent>false</concurrent>
 <build.without.logging>false</build.without.logging>
 </path.value.index>
 </indexes>
</data>
```

```

</path.value.index>
<path.value.index>
 <name>hashtag</name>
 <path>
 /{urn:x-emc:ia:demo:schema:tweets:1.0}tweets/{urn:x-emc:ia:
demo:schema:tweets:1.0}tweet[{urn:x-emc:ia:demo:schema:tweets:1.0}hashtagEntity/
{urn:x-emc:ia:demo:schema:tweets:1.0}text<STRING>]
 </path>
 <compressed>false</compressed>
 <unique.keys>false</unique.keys>
 <concurrent>false</concurrent>
 <build.without.logging>false</build.without.logging>
</path.value.index>
<path.value.index>
 <name>retweetCount</name>
 <path>
 /{urn:x-emc:ia:demo:schema:tweets:1.0}tweets/{urn:x-emc:ia:
demo:schema:tweets:1.0}tweet[{urn:x-emc:ia:demo:schema:tweets:1.0}
retweetCount<INTEGER>]
 </path>
 <compressed>false</compressed>
 <unique.keys>false</unique.keys>
 <concurrent>false</concurrent>
 <build.without.logging>false</build.without.logging>
</path.value.index>
<full.text.index>
 <name>text-fulltext</name>
 <compressed>false</compressed>
 <concurrent>false</concurrent>
 <optimize.leading.wildcard.search>true</optimize.leading.
wildcard.search>
 <index.all.text>true</index.all.text>
 <include.attributes>false</include.attributes>
 <support.phrases>false</support.phrases>
 <support.scoring>false</support.scoring>
 <convert.terms.to.lowercase>true</convert.terms.to.lowercase>
 <filter.english.stop.words>false</filter.english.stop.words>
 <support.start.end.token.flags>false</support.start.end.token.
flags>
 <element.uri>urn:x-emc:ia:demo:schema:tweets:1.0</element.uri>
 <element.name>text</element.name>
 <attribute.uri/>
 <attribute.name/>
</full.text.index>
</indexes>
</data>

```

**Note:** It is recommended that you do not create indexes with the option `build.without.logging` value set to `true`.

## PDI.AIU.CNT

To ensure that the SIP contains the number of AIUs defined in the SIP descriptor, it is necessary to count the AIUs into the PDI XML. To accomplish this, an XQuery is executed. Based on the AIU path, it is easy to count the AIUs. Do not forget to indicate the namespace declaration and prefix.

The following is an example of the `pdi.aiu.cnt` file:

```

<data>
 <id>pdi.aiu.cnt</id>
 <select.query>
```

```
<! [CDATA[
 declare namespace n = "urn:x-emc:ia:demo:schema:tweets:1.0";
 count(/n:tweets/n:tweet)
]]>
</select.query>
</data>
```

This query is valid but it does not use indexes, which could be an issue with a large PDI XML. To use an index, include an indexed element into the XQuery. The element must be present for each AIU. Choose the element carefully.

The following is an example of an optimized pdi.aiu.cnt file:

```
<data>
 <id>pdi.aiu.cnt</id>
 <select.query>
 <! [CDATA[
 declare namespace n = "urn:x-emc:ia:demo:schema:tweets:1.0";
 count(/n:tweets/n:tweet[n:createdAt])
]]>
 </select.query>
</data>
```

## PDI.AIU.ID

To identify each AIU, it is necessary to generate an ID for each. To do that, an XQuery is executed to return all of the AIU nodes. Based on this list, an attribute ID will be added with the corresponding AIU ID value. The AIU ID is composed by the AIP ID and the position into the list with this format. For example:

```
 ${AIP_ID}:aiu:${position}
```

The following is an example of the pdi.aiu.id file:

```
<data>
 <id>pdi.aiu.id</id>
 <select.query>
 <! [CDATA[
 declare namespace n = "urn:x-emc:ia:demo:schema:tweets:1.0";
 for $aiu in /n:tweets/n:tweet
 return $aiu
]]>
 </select.query>
</data>
```

## Using Partition Keys

A partition key is used in the first tier of the query process to limit the data returned when a search is executed. Partition keys are created during ingestion using XQuery and are stored in the AIP object.

An AIP can have multiple partition keys to satisfy different sets of search criteria.

To improve the search performance and reduce the search scope, define one or more partition keys. When a search criterion is linked to a partition key, perform the query only on AIPs that reference the partition key value.

Each partition key value must be assigned to an AIP attribute. Out of the box, the AIP offers some free slots. You need to take into account the type (STRING, INTEGER, DOUBLE, DATETIME, LIST<STRING>).

Using search without defining partition keys may result in the following consequences:

- SIP-based searches will be sensitive to the amount of ingested data (refer to [How Data is Searched](#) for more information). The response time will increase linearly with increasing data volume.
- Background searches will consume a lot of CPU and, therefore, will have an impact on the overall system.

## Types of Partition Keys

There are different types of partition keys:

Type	Number of Values	Attribute
DateTime	6	pkeys.dateTime01
String	4	pkeys.string01
List of String	4	pkeys.values01
Integer	4	pkeys.integer01
Long	4	pkeys.long01
Double	4	pkeys.double01

## Configuring Partition Keys

The following is an example of how to configure partition keys:

```
<data>
<proc_id>pdi.pkeys</proc_id> (1.)
<pkey attr="dateTime01"> (2.)
declare namespace n = "urn:x-emc:ia:example.1.0";
min(/n:objects/n:object/xs:dateTime(n:creationTime)) (3.)
</pkey>
... (4.)
</data>
```

In the example above:

1. Indicates the ID of the partition key processor, which must be pdi.keys.
2. One pkey element per attribute can be defined.
3. The query to set the value (minimum or maximum) or the list of values within a single PDI.
4. Any other pkey element configuring the computation of additional partition keys.

The common configuration is to calculate the minimum/maximum for a dateTime. In the following example from the pdi.pkeys file, it is the `createdAt` value. We will set the minimum value to `dateTime01` and the maximum value to `dateTime02`:

```
<data>
```

```
<id>pdi.pkeys</id>
<pkey attr="dateTime01">
 declare namespace n = "urn:x-emc:ia:demo:schema:tweets:1.0";
 min(/n:tweets/n:tweet/xs:dateTime(n:createdAt))
</pkey>
<pkey attr="dateTime02">
 declare namespace n = "urn:x-emc:ia:demo:schema:tweets:1.0";
 max(/n:tweets/n:tweet/xs:dateTime(n:createdAt))
</pkey>
</data>
```

The XQuery defined for the partition key is executed for every AIU inside the SIP PDI. To improve the performance of XQuery execution, the xDB indexes must be used for the field that is used as a partition key. In addition, to use the xDB indexes and improve the performance with large PDI XML write a different XQuery. It is important to know how xDB indexes are sorted. In the following example, if you pick-up the first value of the index, you have the minimum value:

```
<data>
<id>pdi.pkeys</id>
<pkey attr="dateTime01">
 declare namespace n = "urn:x-emc:ia:demo:schema:tweets:1.0";
 (
 for $aiu in /n:tweets/n:tweet[n:createdAt]
 order by $aiu/n:createdAt ascending
 return $aiu/n:createdAt/xs:dateTime(..)
)[1]
</pkey>
<pkey attr="dateTime02">
 declare namespace n = "urn:x-emc:ia:demo:schema:tweets:1.0";
 (
 for $aiu in /n:tweets/n:tweet[n:createdAt]
 order by $aiu/n:createdAt descending
 return $aiu/n:createdAt/xs:dateTime(..)
)[1]
</pkey>
</data>
```

## Three Common Partition Key Queries

The following partition key query allows you to set the minimum number of results returned within a single PDI:

```
min(/n:objects/n:object/xs:dateTime(n:creationTime))
```

The following partition key query allows you to set the maximum number of results returned within a single PDI:

```
max(/n:objects/n:object/xs:dateTime(n:creationTime))
```

The following partition key query allows you to set the list of values within a single PDI:

```
distinct-values(/n:objects/n:object/n:category)
```

For faster minimum and maximum queries use the following:

- For minimum:

```
(
for $aiu in /n:objects/n:object[n:createdAt]
order by $aiu/n:createdAt ascending
return $aiu/n:createdAt/xs:dateTime(..)
```

```

) [1]

• For maximum:

(
for $saiu in /n:objects/n:object[n:createdAt]
order by $saiu/n:createdAt descending
return $saiu/n:createdAt/xs:dateTime(.)
) [1]

```

## RI.INIT

The RI.INIT XQuery allows you to enumerate the unstructured content included into the SIP. The approach is to retrieve all file names into the PDI XML file and to return a list of <content> elements. For each content, specify the MIME type. This MIME type will be returned during the content retrieval. The MIME type can be static or retrieved one by one from the PDI XML.

**Tip:** Ensure that the correct MIME type is associated as part of the ingestion process if the end user is to utilize the viewer functionality. A MIME type is a way of identifying a file on the Internet according to its nature and format. For example, using the "Content-type" header value defined in a HTTP response, the browser can open the file with the proper extension/plugin. The InfoArchive viewer comes with a set list of MIME types that it supports. The browser native viewer, however, depends on the browser being used and the plugins loaded on the browser. If the browser has the plugin for the MIME type, it would be launched and content would be rendered. Therefore, it is important to ensure that the MIME type is correctly ingested in the system.

If content is linked to more than one AIU, use the distinct-values function to avoid archiving a copy of the same content.

For the Tweets holding, the file name corresponds to the ID value of the mediaEntity:  
`/n:tweets/n:tweet/n:mediaEntity/n:id`.

The following is an example of the `ri.init` file:

```

<data>
 <id>ri.init</id>
 <select.query>
 <![CDATA[
 declare namespace n = "urn:x-emc:ia:demo:schema:tweets:1.0";
 for $ci in distinct-values(/n:tweets/n:tweet/n:mediaEntity/n:id)
 order by $ci
 return <content type="application/octet-stream">{ $ci }</content>
]]>
 </select.query>
</data>

```

## XDB.PDI.CI.ID

To facilitate the content retrieval, enrich the PDI XML with the Content ID (CID). To do that, one XQuery is used to return the nodes where we want to add the CID value. In return, each node must be followed by a string value corresponding to the CID. To support all xDB modes, compute the CID with the help of two external variables ID and SEQNO\_START.

```
<data>
 <id>xdb.pdi.ci.id</id>
 <select.query>
 <! [CDATA[
 declare variable $id as xs:string external;
 declare variable $seqno_start as xs:long external;
 declare namespace n = "urn:x-emc:ia:demo:schema:tweets:1.0";
 declare namespace ri = "urn:x-emc:ia:schema:ri";
 let $pdi_uri := root(.)
 let $ri_uri := replace(document-uri($pdi_uri), '\.pdi$', '.ri')
 for $ri in doc($ri_uri)/ri:ris/ri:ri[@key]
 for $n in /n:tweets/n:tweet/n:mediaEntity/n:id[. = $ri/@key]
 return ($n,concat($id,:ci:", sum((($ri/@seqno,$seqno_start)))))
]]>
 </select.query>
</data>
```

If the SIP contains a lot of unstructured content, it is possible to improve the performance by creating an index on the element containing the file name.

```
<path.value.index>
 <name>filename</name>
 <path>
 /{urn:x-emc:ia:demo:schema:tweets:1.0}tweets/{urn:x-emc:ia:
demo:schema:tweets:1.0}tweet/{urn:x-emc:ia:demo:schema:tweets:1.0}mediaEntity/
{urn:x-emc:ia:demo:schema:tweets:1.0}id<STRING>
 </path>
 <compressed>false</compressed>
 <unique.keys>false</unique.keys>
 <concurrent>false</concurrent>
 <build.without.logging>false</build.without.logging>
</path.value.index>
```

## Configuring a Holding

A holding is a logical destination where data that shares common characteristics is archived. Some example of common characteristics include:

- Data from the same source application;
- Data in the same format, such as audio recordings;
- The same type of data (for example, communication such as e-mail, chat, faxes, etc.); and
- Data that belongs to the same business entity.

An application can contain multiple holdings. Multiple holdings can exist for a single data type.

A holding is also the central configuration object in SIP-based archiving. The following is defined in a holding:

- storage area
- retention classes
- ingestion sequence
- the AIP mode and xDB mode being used

When creating a holding, you should consider the types of data that will be archived as well as the data segregation/isolation restrictions.

In general, ingestion and search performance depends on the application configuration, holding configuration and the following external IT factors:

- Number of AIUs per SIP
- Ingestion mode
- xDB indexes
- Partitioning keys

The holding configuration is present in the YAML configuration file as the holding item. For example:

```
holding:
 name: Tweets
 ciStore: default-store
 defaultRetentionClass: default
 ingestNodes:
 - ingest_node_01
 logStore: default-store
 managedItemStore: null
 pdiConfigs:
 - schema: urn:eas-samples:en:xsd:tweets.1.0
 priority: 1
 renditionStore: default-store
 retentionClasses:
 - name: default
 policies:
 - Tweets-policy
 sipStore: default-store
 stagingStore: null
 subPriorities:
 - deadLine: 100
 priority: 0
 - deadLine: 200
 priority: 1
 xdbLibraryParent: Tweets-xdb-library
 xdbStore: default-store
 xmlStore: default-store
 xdbMode: PRIVATE
 ingestConfigs:
 - sipFormat: eas_sip_zip
 - sipFormat: sip_zip
 retentionClasses:
 - name: default
 policies:
 - Documentum-policy
```

## Holding Configuration – Stores

The **Holdings** tab in IA Web App contains a list of the available holdings.

Furthermore, use the **Stores** tab to view the available storage.

The following is a breakdown of the holding configuration for a store:

```
sipStore: store-name (1.)
ciStore: store-name (2.)
xmlStore: store-name (3.)
logStore: store-name (4.)
renditionStore: store-name (5.)
xdbStore: store-name (6.)
```

```
managedItemStore: store-name (7.)
```

1. Indicates where the entire received SIP will be stored.
2. Indicates where the content information containers items and RI XML will be stored.
3. Indicates where the XML files will be stored (PDI XML, SIP XML).
4. Indicates where the log files will be stored.
5. Indicates where analytical rendition and purge list exports will be stored.
6. Indicates where the xdb sub-library will be stored.
7. Indicates where the retention data (managed items data) will be stored.

## Holding Configuration: Hashing

The following is a breakdown for hash validation from the holding in the `configuration.yml` file:

```
ciHashValidationEnabled : true (1.)
pdiXmlHashValidationEnabled: true (2.)
pdiXmlHashEnforced: true (3.)
```

1. Indicates whether content hash validation is enabled or not.
2. Indicates whether pdi hash validation is enabled or not.
3. Indicates whether or not to require that a hash exists and is validated.

## Holding Configuration: Retention

The following is a breakdown of the holding configuration for retention:

```
holding:
 ...
 retentionClasses: (1.)
 - name: default
 policies:
 - Tweets-policy
 defaultRetentionClass: default (2.)
```

1. Indicates the retention class, name and the policy to apply if this retention class is used.
2. Indicates the default retention class. If no retention class is specified in the SIP, this retention class will be used.

# Configuring Ingestion

## Configuring xDB Ingestion Mode

When creating a holding, you can define the xDB ingestion mode. The `xdbMode` property in the holding item allows you to configure the xDB ingestion mode (configurable through the `holding.xdbMode` property). The following values are acceptable

- PRIVATE
- POOLED
- AGGREGATE

Proper ingestion mode is the key aspect for optimal search performance. In general, the ingestion mode depends on the SIP package characteristics. If it is possible to store a huge number of AIUs (for example, 100,000) in the SIP package, then it is fine to use the PRIVATE mode. In this mode, indexes are created during ingestion. If it is possible to put only few AIUs per SIP (for example, 2 to 100), then AGGREGATION mode works better because a lot of small SIPs are aggregated into a single library. And the indexes are created on that library only after the Close job has been executed.

InfoArchive also allows you to dynamically select the xDB mode to use during reception and ingestion.

The dynamic mode is an optimization for the aggregate or pooled mode to switch back to private mode when there are too many records (for example, more than the limit indicated by the `xdbLibraryPolicy.aiuThreshold` property).

When `xdbLibraryPolicy.aiuThreshold` property is not set or set to 0, the dynamic mode is not activated. If the value is more than 0, the dynamic mode is activated for the AGGREGATE and POOLED modes.

## Unitary Archiving and Aggregation

Unitary archiving allows a client application to synchronously archive data in InfoArchive. If the web service call succeeds, archived data can be immediately searched. Unitary archiving executes the same processing as batch ingestion.

Unitary archiving can be completed by an aggregation step to optimize storage and reduce the repository footprint.

## Receiving and Ingesting a SIP in One Request

When ingesting SIPs in batch, it is required to receive all SIP packages and then ingest them into the application.

To achieve this, an exposed REST API can receive and ingest simultaneously. Receiving an ingest a SIP in one request reduces the number of steps you must perform (for example, saving the SIP file). The resource name is `ingest-direct`.

Refer to the *REST Development Guide* for more information.

## Configuring the Ingestion Process to Store Multiple AIPs in the Same xDB Library

When working in the aggregation mode, InfoArchive allows you to store metadata into shared libraries, which are also known as pooled libraries.

The effective close date is computed with the following rules:

Close Mode	Rule
NONE	nulldate
LAST_MODIFIED_DATE	CLOSE PERIOD + XDB LIBRARY LAST MODIFIED DATE
CREATION_DATE	CLOSE PERIOD + XDB LIBRARY CREATION_DATE
CLOSE_HINT_DATE	CLOSE PERIOD + XQUERY DATE TIME

The XdbLibraryPolicy is referenced to the Holding: in the following attributes:

- xdbMode
- xdbLibraryPolicy

The following attributes are added at the AIP level:

- xdbMode
- phaseCode
- stateCode

The following attributes are added at the XdbLibrary level to facilitate the library management:

- pkey
- aipCount
- aiuCount
- effectiveCloseDate
- closed
- closedDate
- closeRequested
- xdbMode

## Executing the Close Job to Close the xDB Library and Perform a Back Up

When you run the Close job for each xDBLibrary:

- An xDB back up is performed;
- The XdbLibrary is set to closed = TRUE; and
- The closeDate value is updated.

The back up is only performed if every AIP of the xdbLibrary is set to COMPLETED.

An xDBLibrary with the xdbMode set to PRIVATE is not eligible to be closed. Only an xDBLibrary in xdbMode POOLED or AGGREGATE can be closed:

```
XdbLibrary where xdbMode in (POOLED, AGGREGATE) and closed = FALSE and
(effectiveCloseDate < (offsetDateTime(now) - {closeDelay}) or closeRequested = true)
```

**Note:** An exceeded quota never triggers a closure. Only an outdated closing date or a manual close can be taken into account.

The job accepts the following parameters to control the execution:

- phaseToProcess
- closeDelay

Refer to [Close Job](#) for further information.

## Configuring the Ingestion Process to Allow Aggregation

The aggregation mode is based on the pooled mode with additional constraints to allow the aggregation at a later time. To activate the aggregation mode, it is necessary to specify at the Holding level the xdbMode AGGREGATE and link a new configuration object XdbLibraryPolicy.

To configure and enable this mode:

1. Define an xDB library policy in the configuration.yml file. For example:

```
xdbLibraryPolicy:
 name: Tweets-xdb-library-policy
 aipQuota: 10
 aiuQuota: 100
 closeHintDateQuery: current-dateTime()
 closeMode: close hint date
 closePeriod: 2
 pKeyQuery: |
 declare namespace n="urn:x-emc:ia:schema:sip:1.0";
 year-from-dateTime(xs:dateTime(/tweets/tweet/createdAt/text()))
```

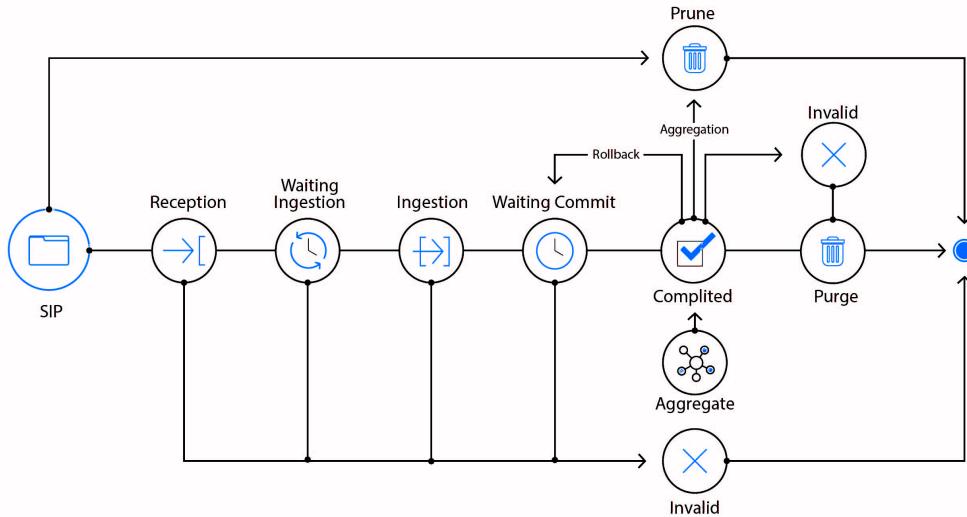
2. On the holding configuration, use this library policy and change the xdbMode value to AGGREGATE:

```
holding:
 ...
 # (if there is only one defined, this will be used as default
 # automatically)
 xdbLibraryParent: Tweets-xdb-library-policy
 xdbMode: AGGREGATE
```

To close and perform the aggregation, execute the Close job. An aggregate can be closed only:

- When the `effectiveCloseDate` is expired; or
- If a manual closing has been requested.

At the end, the children (in COMPLETED phase) will be moved to the PRUNE phase and the AIP AGGREGATE will be moved to the COMPLETED phase.



The `effectiveCloseDate` is computed with the information stored in the `xdbLibraryPolicy` item and the close mode you have chosen on the `xdbLibraryPolicy`:

- `closeMode`: NONE, CREATION\_DATE, LAST\_MODIFIED\_DATE, CLOSE\_HINT\_DATE
- `closePeriod`: number of days
- `closeHintDateQuery`: XQuery to compute a DateTime on the fly based on the SIP XML information

NONE:	nulldate
CREATION_DATE:	CLOSE PERIOD + XDB LIBRARY
LAST_MODIFIED_DATE	CLOSE PERIOD + XDB LIBRARY LAST
CLOSE_HINT_DATE	CLOSE PERIOD + XQUERY DATE TIME

## Reception

If `xdbMode` is set to `AGGREGATE`, the AIP is not put under retention. The AIP should have `isSipLast` set to `true` and `SipSeqNo` set to `1` or it will fail validation. `CommitSync` is always enabled, whatever the value of the holding.

## Using a Staging Store (Optional)

Some storage, especially Centera, are not designed to save temporary data. The staging store is a temporary store, usually on the local disk, that can be used for all the write operations prior to closing the aggregate (reception, ingestion, invalidation, etc.).

The staging store is defined at the Holding level. When the staging store is enabled, all the write operations will go to this store.

## Executing the Confirmation Job to Identify Open and Closed Aggregates

The Confirmation job identifies open or closed aggregates:

- Open aggregates are not confirmed. An aggregate is considered open under the following conditions:
  - the xdbMode is set to AGGREGATE; and
  - the Phase is set to Phase.AGGR; and
  - the isPartOfAggregate is set to false.

If these conditions are met, the AIP is skipped.

- Child AIPs are processed as regular AIPs, and go through the current confirmation as regular AIPs.
- Closed aggregates AIP:

A closed AIP has the following status:

```
xdbMode == AGGREGATE) && (Aip.State == State.COM OR State.INV_WPROC) &&
Aip.isPartOfAggregate == false)
```

Closed AIPs can either be completed or invalidated.

- SIP Confirmation:  
A closed aggregate contains all the children's `sip.xml` in a `tar.gz` file. The confirmation is completed on each individual `sip.xml` in the `tar.gz` archive.
- PDI confirmation:  
A closed aggregate library contains all the PDI files. The confirmation will be completed for all PDI files.

## Retrieving a CI from an Aggregate

When an aggregate is opened, perform the following additional steps to retrieve the child AIP and the entry into the Reference Information (RI):

1. Extract the AIP ID from the Content Information (CI) ID.
2. Load the AIP.

If the AIP is an aggregate (`xdbMode = AGGREGATE`) and is not aggregated (`phaseCode = AGGREGATE` and `partOfAggregate = FALSE`):

3. Retrieve the child AIP with a pseudo query:

- a. Retrieve the SEQNO from the CI ID.
- b. Retrieve the XDB\_LIBRARY\_ID from the AIP aggregate.

It is expected to retrieve a '0' to '1' result.

The query must take into account the permissions at the AIP level (filter = true):

```
from AIP where xdbLibrary = {XDB_LIBRARY_ID} and phaseCode = 'COM' and
partOfAggregate = TRUE and aggregateCiSeqno <= {SEQNO} and
aggregateCiSeqno + ciCount > {SEQNO}
```

4. Compute the new CI ID:

- a. Load the child AIP.
- b. Retrieve the AIP ID.
- c. Adjust the SEQNO (SEQNO - aggregateCiSeqno).

## Configuring a Search

Search configuration can be broken down into the following phases:

- *AIC Configuration*: The Developer indicates which AIPs can be searched as well as specify the search criteria. To get the whole search context, a search should only be defined at the search composition level, not at the AIC level.
- *Query Configuration*: The Developer indicates the business object and sets up the mapping between each search criterion and the corresponding value in the PDI XML file. It is also in this phase that the Developer can specify any transformation of search results, such as decryption.
- *Query Quota*: The Developer indicates:
  - The maximum number of AIPs and AIUs a search can span, and
  - The maximum number of AIUs that a search can return.

**Note:** This section illustrates the more advanced method of configuring a search. The primary and easier method is to utilize IA Web App. Refer to [Composing the Search Form](#) for more information.

## Configuring the AIC

In this phase, the Developer defines the AIC with a collection of AIPs.

The Developer also defines the search criteria that can be used to search those AIPs:

```
aic:
 name: Tweets-aic
 criteria:
 - name: createdAt (1.)
 label: Created At (2.)
 type: datetime (3.)
 pKeyMinAttr: pkeys.dateTime01 (4.)
 pKeyMaxAttr: pkeys.dateTime02
 holdings:
 - Tweets
```

```
predicate: ''
```

1. Indicates the name of the search criterion.
2. Indicates the label of the search criterion.
3. Indicates the data type.
4. Indicates which, if any, partition key can be used.

This item exposes the list of allowed search criteria. For each criterion, specify the type and whether the criterion is a partition key with the name of the attributes to use on the AIP.

**Tip:** When a SIP search does not return the expected results, the search configuration may be wrong (for example, not the suitable type or not the same type between the Query and AIC configuration objects for the `xdbPdiConfigs` operand type and the criteria's type).

- The allowed types are: STRING, INTEGER, LONG, DOUBLE, DATE, DATETIME, ID and CID. The type is used to validate the submitted values.
- If you have defined a minimum/maximum partition key, populate the following fields: `pKeyMinAttr` / `pKeyMaxAttr`.
- If you have defined a single or multi-values partition key, populate the following field: `pKeyValuesAttr`.

## Using an AIC Predicate

Using an AIC predicate restricts visibility to a subset of items (AIPs) matching predefined metadata conditions. The predicate is added to the query for every search and background search created with this AIC. This restriction improves search performance.

The predicate is defined in SpEL (Spring Expression Language) in the ‘predicate’ field of the AIC object. The predicate contains expression on metadata of the AIP. This field is optional (refer to the AIC sample above) but, if it is defined, it should be on valid AIP fields. Otherwise, either the AIC creation fails or the search execution fails.

The following are examples of the predicate:

Predicate	Description
<code>dss.entity == 'Dept1'</code>	Restricts the AIP list to the ones from ‘Dept1’ entity.
<code>(dss.entity == 'Dept1' and dss.holdingName == 'Invoices-1') or (dss.entity == 'Dept2' and dss.holdingName == 'Invoices-2')</code>	The AIP list is restricted to ‘Dept1’ entity for the Invoices-1 holding and to ‘Dept2’ entity for the Invoices-2 holding.
<code>(pkeys.values03 == 'myvalue') and (ingestStartDate &gt; '2017/01/01')</code>	The AIP list is filtered by partition key values03 set to ‘myvalue’ and by ingestStartDate after 1st January 2017.

Not every SpEL expression is allowed:

- The expression must be a selection:

```
dss.entity == 'Dept1'
```

- Relational operators are allowed:

```
>, >=, <, <=, ==, != , matches
```

- Logical operators are supported:

```
and, or
```

- Following method references are supported too:

```
String::toLowerCase(), String::toUpperCase()
```

The rest of the SpEL language is invalid for AIC predicate and dynamic values are also not supported (such as

```
'$session.userName'
, etc.).
```

## Configuring the Query

During this phase, the Developer sets the entity path, which is the path from the root of the PDI XML file to the AIU.

The mapping is between each search criterion and the corresponding value in the PDI XML file:

```
query:
 name: PhoneCalls-query
 xdbPdiConfigs:
 - entityPath: /n:Calls/n:Call
 operands:
 - name: CallStartDate (1.)
 path: n:CallStartDate (2.)
 type: DATETIME (3.)
 indexed: true (4.)
```

1. Indicates the name of the operand, which must correspond to a name of a criterion in the AIC.
2. Indicates the path to the corresponding element in the PDI XML.
3. Indicates the data type.
4. Indicates whether or not there is an index supporting this operand.

Optionally, it can also define a query prolog and a query template.

It is possible to have several query configurations for different use cases.

The following is an example of a query configuration as it may appear in the  
<INFOARCHIVE\_ROOT>/examples/applications/<APPLICATION\_NAME>/config  
/searches/configuration.yml file:

```
query:
 name: PhoneCalls-query
 aics:
 - PhoneCalls-aic
 namespaces:
```

```

- prefix: n
 uri: urn:eas-samples:en:xsd:phonecalls.1.0
 quota: PhoneCalls-quota
 quotaAsync: PhoneCalls-quota
 resultRootNsEnabled: false
 resultSchema: urn:eas-samples:en:xsd:phonecalls.1.0
 xdbPdiConfigs:
 - entityPath: /n:Calls/n:Call
 operands:
 - name: CallEndDate
 path: n:CallEndDate
 type: DATETIME
...

```

This item contains the mapping between the search criteria and the xDB PDI XML file. Enumerate all operands and indicate the criterion name, the path, the type and whether it is indexed or not. The allowed types are: STRING, INTEGER, LONG, DOUBLE, DATE, DATETIME, ID and CID.

To list the operands, specify the entityPath. This path corresponds to the AIU level. All others, paths are defined from this level.

## Query Template

The query template can be used to modify the search results (for example, to add metadata from the AIP properties or to decrypt metadata). The PhoneCalls sample application uses both examples.

To add the `sipProductionDate` property of the AIP, add the following query template:

```

template: |
 let $root := root($aiu)
 let $aipid := xhive:metadata($root, 'aip_id')
 let $value := ia-fun:property-retriever($aipid, 'sipProductionDate')
 return element {node-name($aiu)} {$aiu/@*, $aiu/*,
<n:sipProductionDate>{$value}</n:sipProductionDate>}

```

Then, modify the ResultHelper to add the `sipProductionDate` value to the result configuration (refer to [Configuring the ResultHelper](#) for further information).

To decrypt metadata, add the following query template:

```

template: |
 let $root := root($aiu)
 let $aipid := xhive:metadata($root, 'aip_id')
 return ia-fun:decrypt($aiu, $aipid)

```

## Configuring the Query Quota

The following is a breakdown of the quota item:

```

queryQuotas:
- name: Tweets-quota
 aipQuota: 0 (1.)
 aiuQuota: 0 (2.)

```

```
dipQuota: 0 (3.)
```

1. Indicates the maximum number of AIPs a synchronous search can span. '0' indicates that there is no limit.
2. Indicates the maximum number of AIUs a synchronous search can span. '0' indicates that there is no limit.
3. Indicates the maximum number of AIU returns by a search. '0' indicates that there is no limit.

## Configuring the ResultHelper

The ResultHelper configuration defines what values are displayed in the user interface.

The result configuration XML file is referenced from the configuration YAML file. For example:

```
resultConfigurationHelper:
 name: urn:eas-samples:en:xsd:tweets.1.0
 content:
 format: xml
 resource: data-model-config/resultConfigurationHelper-tweets.xml
 propagateChanges: true
 resultSchema:
 - urn:eas-samples:en:xsd:tweets.1.0
```

For each value you must define an element tag consisting of:

- name
- label
- path
- type
  - STRING, INTEGER, LONG, DOUBLE, DATE, DATETIME, ID, CID

The following outlines the ResultHelper template:

```
<?xml version="1.0" encoding="UTF-8"?>
<resultConfigurationHelper
 xmlns:n="urn:eas-samples:en:xsd:phonecalls.1.0" ①
 xmlns:ia="urn:x-emc:ia:schema:pdi"> ②
 <element>...</element>
 ...
 <group> ...</group>
 ...
</resultConfigurationHelper>
```

1. Indicates the schema(s) of the PDI XML.
2. Indicates the InfoArchive PDI schema if you are referencing content, etc.

The following is a breakdown of each element in the `resultConfigurationHelper.xml` file:

```
<element>
 <name>RepresentativeID</name> (1.)
 <label>Representative ID</label> (2.)
 <type>INTEGER</type> (3.)
 <path>n:RepresentativeID</path> (4.)
```

```
</element>
```

1. Indicates the name of the element.
2. Indicates the label (in English) that is displayed in IA Web App.
3. Indicates the type of the column, which determines what the user can do in IA Web App.
4. Indicates the xPath to the element.

This XML describes the XML returned by the query. This description will be used during the composition to help the Developer select the column to display in the result page.

Add the namespaces prefix at the root level.

For each column you want to expose, add an <element/> node. The element needs to contain a name, a label, a type and a path. The allowed types are : STRING, INTEGER, LONG, DOUBLE, DATE, DATETIME, ID and CID. You can write the XPath you want. It can be useful to perform some value massage (string-join, duration computation, etc.).

For repeating elements, enclose with a <group/> node. The group needs to contain a name, a label and a path.

**Note:** Do not forget to include the CID to have the ability to download CIs.

```
<?xml version="1.0" encoding="UTF-8"?>
<resultConfigurationHelper xmlns:n="urn:x-emc:ia:demo:schema:tweets:1.0"
 xmlns:ia="urn:x-emc:ia:schema:pdi">
 <element>
 <name>id</name>
 <label>ID</label>
 <type>LONG</type>
 <path>n:id</path>
 </element>
 <element>
 <name>text</name>
 <label>Text</label>
 <type>STRING</type>
 <path>n:text</path>
 </element>
 <element>
 <name>createdAt</name>
 <label>Created at</label>
 <type>DATETIME</type>
 <path>n:createdAt</path>
 </element>
 <element>
 <name>retweet</name>
 <label>Retweet</label>
 <type>STRING</type>
 <path>n:retweet</path>
 </element>
 <element>
 <name>retweetCount</name>
 <label>Retweet count</label>
 <type>INTEGER</type>
 <path>n:retweetCount</path>
 </element>
 <element>
 <name>userId</name>
 <label>User ID</label>
 <type>LONG</type>
 <path>n:user/n:id</path>
 </element>

```

```
</element>
<element>
 <name>userName</name>
 <label>User name</label>
 <type>STRING</type>
 <path>n:user/n:name</path>
</element>
<element>
 <name>userLocation</name>
 <label>Location</label>
 <type>STRING</type>
 <path>n:user/n:location</path>
</element>
<element>
 <name>userScreenName</name>
 <label>Screen name</label>
 <type>STRING</type>
 <path>n:user/n:screenName</path>
</element>
<element>
 <name>Media1</name>
 <label>Media 1</label>
 <type>CID</type>
 <path>n:mediaEntity[1]/n:id/@ia:cid</path>
</element>
<group>
 <name>MediaEntities</name>
 <label>Medias</label>
 <path>n:mediaEntity</path>
<element>
 <name>cid</name>
 <label>CID</label>
 <type>CID</type>
 <path>n:id/@ia:cid</path>
</element>
<element>
 <name>mediaId</name>
 <label>ID</label>
 <type>LONG</type>
 <path>n:id</path>
</element>
<element>
 <name>mediaUrl</name>
 <label>URL</label>
 <type>STRING</type>
 <path>n:url</path>
</element>
<element>
 <name>mediaType</name>
 <label>Type</label>
 <type>STRING</type>
 <path>n:type</path>
</element>
<element>
 <name>mediaStart</name>
 <label>Start</label>
 <type>INTEGER</type>
 <path>n:start</path>
</element>
<element>
 <name>mediaEnd</name>
 <label>End</label>
 <type>INTEGER</type>
 <path>n:end</path>
</element>
```

```
</group>
<group>
 <name>urlEntities</name>
 <label>URLs</label>
 <path>n:urlEntity</path>
 <element>
 <name>urlText</name>
 <label>Text</label>
 <type>STRING</type>
 <path>n:url</path>
 </element>
 <element>
 <name>urlUrl</name>
 <label>URL</label>
 <type>STRING</type>
 <path>n:url</path>
 </element>
 <element>
 <name>urlStart</name>
 <label>Start</label>
 <type>INTEGER</type>
 <path>n:start</path>
 </element>
 <element>
 <name>urlEnd</name>
 <label>End</label>
 <type>INTEGER</type>
 <path>n:end</path>
 </element>
 <element>
 <name>expandedUrl</name>
 <label>Expanded url</label>
 <type>STRING</type>
 <path>n:expandedUrl</path>
 </element>
 <element>
 <name>displayUrl</name>
 <label>Display url</label>
 <type>STRING</type>
 <path>n:displayUrl</path>
 </element>
</group>
<group>
 <name>hashtagEntities</name>
 <label>Hashtags</label>
 <path>n:hashtagEntity</path>
 <element>
 <name>hashtagText</name>
 <label>Text</label>
 <type>STRING</type>
 <path>n:url</path>
 </element>
 <element>
 <name>hashtagStart</name>
 <label>Start</label>
 <type>INTEGER</type>
 <path>n:start</path>
 </element>
 <element>
 <name>hashtagEnd</name>
 <label>End</label>
 <type>INTEGER</type>
 <path>n:end</path>
 </element>
</group>
```

```
</resultConfigurationHelper>
```

To configure the `resources/content/result-configuration-helper.xml` file in the legacy PhoneCalls application:

```
<element>
 <name>sipProductionDate</name>
 <label>Sip Production Date</label>
 <type>DATETIME</type>
 <path>n:sipProductionDate</path>
</element>
```

## Configuring Repeating Elements

Repeating elements are handled with a group element. Group elements cannot be nested. A group consists of:

- name
- label
- path

The following is a breakdown of repeating elements:

```
<document>
 ...
<author> (1.)
 <firstname>John</firstname>
 <lastname>Steinbeck</lastname>
</author>
<author>
 <firstname>August</firstname>
 <lastname>Strindberg</lastname>
</author>
</document>
```

1. Indicates the repeating element.

The following is a breakdown taken from a `resultConfigurationHelper.xml` file:

```
<group>
 <name>Authors</name> ①
 <label>Authors</label> ②
 <path>n:author</path> ③
 <element>
 <name>firstname</name>
 <label>First name</label>
 <type>STRING</type>
 <path>n:firstname</path>
 </element>
 <element>
 <name>lastname</name>
 <label>Last name</label>
 <type>STRING</type>
 <path>n:lastname</path>
 </element>
</group>
```

```
</group>
```

1. Indicates the name of the group.
2. Indicates the label (in English) that is displayed in IA Web App.
3. Indicates the xPath to the repeating element.

## Using the Confirmation Mechanism

A confirmation is a message generated in reaction to an AIP event. Confirmation messages are generated during a package's lifecycle transitions.

The confirmation mechanism acknowledges to the source applications or notifies other business applications, such as a portal, that new objects have been archived. For example, once a SIP has been ingested and committed, a storage confirmation message is generated and can be passed back to the source application to confirm that the data has been correctly archived. The message can be used to trigger the deletion of the data from the source application or to register the content IDs.

A confirmation message can be generated from the SIP descriptor or the PDI metadata files. The PDI metadata is only accessible for two events:

- Storage
- Purge

To perform a query on the PDI metadata, the AIP needs to be online. If the AIP is not online, the system requests a cache in. In this case, the confirmation is delayed. To perform a query on the SIP descriptor, the file needs to be immediately accessible in the XML store. If the file is archived to Glacier, a restoration is requested and the confirmation is delayed.

The following event types can trigger a confirmation:

Type	Description	SIP Query	PDI Query	Priority
Receipt	Indicates that the SIP has been received.	Available	Not available	Custom SIP XQuery > Default SIP XQuery
Storage	Indicates that the AIP ingestion has been ingested and committed.	Available	Available	Custom PDI XQuery > Custom SIP XQuery > Default SIP XQuery
Reject	Indicates that the AIP has been rejected.	Available	Not available	Custom SIP XQuery > Default SIP XQuery
Invalid	Indicates that the AIP has been invalidated.	Available	Not available	Custom SIP XQuery > Default SIP XQuery
Purge	Indicates that the AIP has been disposed.	Available	Available	Custom PDI XQuery > Custom SIP XQuery > Default SIP XQuery

You can configure zero or multiple confirmations for a single event for each AIP. The configuration is based on two main objects:

- Confirmation
- DeliveryChannel

## Working with the Confirmation Object

The following table outlines the fields in the confirmation object:

Field	Description
id	Type: UUID  Label: ID
name	Type: String  Label: name  Name of the confirmation
application	Type: Application  Label: Application  Application of this confirmation
types	Type: List of strings  Label: Confirmation type  Possible values: receipt, storage, purge, reject, invalid
holdings	Type: List of holdings  Label: Holdings  List of the holdings on which the confirmation can be applied to.

Field	Description
sipQuery	<p>Type: String</p> <p>Label: SIP query</p> <p>Local mode (will run the query on the sip.xml file attached). If a SIP XQuery has not been defined, the following default query is used:</p> <pre>declare namespace s = "urn:x-emc:ia:schema:sip:1.0"; declare namespace fun = "urn:x-emc:ia:functions"; declare variable \$aip_id as xs:string external; declare variable \$conf_type as xs:string external; declare variable \$conf_datetime as xs:string external; for \$sip in /s:sip return &lt;confirmation&gt; &lt;aip_id&gt;{\$aip_id}&lt;/aip_id&gt; &lt;conf_type&gt;{\$conf_type}&lt;/conf_type&gt; &lt;conf_datetime&gt;{\$conf_datetime}&lt;/conf_datetime&gt; &lt;sip&gt;{\$sip}&lt;/sip&gt; &lt;/confirmation&gt;</pre> <p>Some external variables are available during the XQuery execution:</p> <ul style="list-style-type: none"> <li>• An <code>\$aip_id</code> event indicates the ID of the AIP.</li> <li>• A <code>\$conf_type</code> event indicates the type of the confirmation.</li> <li>• A <code>\$conf_datetime</code> event indicates the date of the confirmation.</li> </ul>
pdiQuery	<p>Type: String</p> <p>Label: PDI query</p> <p>xDB mode (will run the query on the pdi file in xDB). If a PDI XQuery has not been defined, the SIP XQuery is used.</p> <p>A pdiQuery can be defined in the <code>240-confirmation.xml</code> file:</p> <pre>&lt;confirmation&gt; &lt;...&gt; &lt;pdiQuery&gt;&lt;![CDATA[ declare namespace sip = "urn:x-emc:ia:schema:sip:1.0"; let \$uri := replace(document- uri(.), '\.pdi\$', '.sip') for \$c in doc(\$uri)/sip:sip/ sip:dss/sip:producer return &lt;result&gt;{\$c}&lt;/result&gt; ]]&gt;&lt;/pdiQuery&gt; &lt;/...&gt; &lt;/confirmation&gt;</pre>

Field	Description
deliveryChannel	Type: DeliveryChannel  Label: Delivery channel  The delivery channel to use for this confirmation.
deliveryChannelParameters	Type: String key value pairs  Label: Delivery channel parameters  The list of additional parameters to pass to the delivery channel (key, value). For example, '%conf_type'.

An example of a confirmation in a configuration.yml file:

```
confirmation:
 name: PhoneCalls-confirmation
 deliveryChannelParameters:
 aip_id: '%ia_conf_aip_id%'
 ia_conf_datetime: '%ia_conf_datetime%'
 ia_conf_type: '%ia_conf_type%'
 holdings:
 - PhoneCalls
 types:
 - invalid
 - purge
 - receipt
 - reject
 - storage
```

## Working with the DeliveryChannel Object

This object allows you to control where to write the confirmation message.

The following table outlines the fields in the DeliveryChannel object:

Field	Description
id	Type: UUID  Label: ID
application	Type: Application  Label: Application  Application to which it depends.
name	Type: String  Label: Name  The name of the delivery channel.

Field	Description
parameters	<p>Type: String key value pairs</p> <p>Label: Parameters</p> <p>Default parameters to use. Only fixed parameters can be used (for example, do not specify dynamic values as %aip_id%).</p>
store	<p>Type: Store</p> <p>Label: Store</p> <p>The store to use as root:</p> <ul style="list-style-type: none"> <li>• If the store is a xdbstore, the destination will be xDB.</li> <li>• If the store is a filestore, the destination will be the filesystem.</li> <li>• If another kind of store is specified, an exception will be raised.</li> </ul>
fileName	<p>Type: String</p> <p>Label: File/document name</p> <p>The value can contain variables such as %key1% defined in the parameters map. Random UUID, by default.</p>
subPath	<p>Type: String</p> <p>Label: File/XdbLibrary sub-path</p> <p>Relative path from the FileSystemFolder/XdbLibrary parent defined at the Store level (cannot start with / or contain ./, .. or //). The value can contain variables, such as %key1% defined in the parameters map. Empty by default.</p>
prefix	<p>Type: String</p> <p>Label: File name prefix</p> <p>The value can contain variables such as %key1% defined in the parameters map. Empty by default.</p>
suffix	<p>Type: String</p> <p>Label: File name suffix</p> <p>The value can contain variables such as %key1% defined in the parameters map. Empty by default.</p>
overwrite	<p>Type: boolean</p> <p>Label: Overwrite</p> <p>Remove the file if it already exists. False by default.</p>

Field	Description
compress	Type: boolean Label: Compress Compress the file and add the zip extension. False by default.
metadata	Type: String key value pairs Label: Metadata xDB metadata added on the document. The value can contain variables such as %key1% defined in the parameters map. Empty by default.

The deliverychannel object is defined in the `fileconfiguration.yml` file. For example:

```
deliveryChannel:
 name: default-delivery-channel
 fileName: confirmation
 overwrite: true
 prefix: '%aip_id%-'
 store: default-confirmation-store
 subPath: confirmation/%ia_conf_type%
 suffix: .xml
```

If you want to use dynamic parameters, you can define them at the confirmation object level by adding a new entry in the field `deliveryChannelParameters`:

```
confirmation:
 name: PhoneCalls-confirmation
 deliveryChannelParameters:
 aip_id: '%ia_conf_aip_id%'
 ia_conf_datetime: '%ia_conf_datetime%'
 ia_conf_type: '%ia_conf_type%'
 ...
```

There are two types of dynamic parameters:

- The precalculated one:

Variable	Description
%ia_conf_type%	Indicates the type of the confirmation.
%ia_conf_datetime%	Indicates the date of the confirmation.
%ia_conf_aip_id%	Indicates the ID of the AIP that is confirmed.
%ia_conf_aip_aipid%	Indicates the aipId of the AIP that is confirmed.
%ia_conf_cfg_id%	Indicates the ID of the confirmation configuration.
%ia_conf_cfg_name%	Indicates the name of the confirmation configuration.

- The object properties only for object AIP and confirmation.

You can also dynamically get the properties of two available objects:

- AIP
- Confirmation

Use the following format to create a pattern: %<object\_pattern>. <propertyName>%:

Object Pattern	Object	Example
ia_conf	Confirmation	%ia_conf.name%
ia_aip	AIP	%ia_aip.id%

## Creating an Application Using Declarative Configuration

The sample applications include a [YAML](#)-based declarative configuration files that:

- Make it easier to set up an application.
- Support migration between environments (for example, migrating from the staging to production environment).

The new declarative configuration files can be found in the <INFOARCHIVE\_ROOT>\examples\applications\<APPLICATION\_NAME>\config directory. The configuration.yml file contains metadata describing the configuration objects. Certain content is put in external files that are referenced from the YAML files, to make that content easier to edit.

A YAML file must be complete in the sense that all objects that are being referenced must also be present in the file. For instance, when configuring a space that points to an application, there must be a section in the YAML file for the application.

The [YAML](#)-based declarative configuration files are easier to use, as you simply state what you want, not how to get there (for example, when using the Ant-based files, you would have to enter two tasks: <create> and <update>). The YAML-based files support additional use cases, such as migration.

- The YAML structure is mapped to configuration objects using the required syntax.
- To properly use default values.
- To work with namespaces and queries.
- To simplify the configuration process to make it easier to specify objects.
- To prevent overwrites.

## Detailed Information About the Declarative Configuration Format

The following subsections explain:

- The YAML structure is mapped to configuration objects using the required syntax.
- To properly use default values.
- To work with namespaces and queries.
- To simplify the configuration process to make it easier to specify objects.
- To prevent overwrites.

## Using the Correct Syntax

First, identify the version of the configuration to properly track its evolution:

```
version: 1.0.0
```

The value is a semantic version. If it is omitted, `1.0.0` is assumed.

At the top level of the YAML file, define objects that correspond to configurable resource types. If only a single object is to be configured, use the singular name of the resource:

```
tenant:
 name: INFOARCHIVE
```

Most top-level objects have a name that identifies them.

At lower levels, set values for the object's properties using name/value pairs:

```
application:
 name: PhoneCalls
 category: Customer Support
 description: The application has customer support phone calls history
 type: ACTIVE_ARCHIVING
 archiveType: SIP
 metadataCacheSize: 0
```

Some properties can only take on a value from a predefined set. These values are constants and, therefore, written in all uppercase letters. In the configuration file, optionally, specify these values using all lowercase letters and replacing underscores with spaces:

```
application:
 name: PhoneCalls
 category: Customer Support
 description: The application has customer support phone calls history
 type: active archiving
 archiveType: sip
 metadataCacheSize: 0
```

Some properties take on values that are secrets, such as passwords or encryption keys. It is important to protect this sensitive information. For instance, use the Transport Layer Security (TLS) Protocol when sending a configuration file to the server. You may want to store them encrypted locally and only decrypt them for sending.

The server will not export secrets to ensure that data is not included in exported configurations.

When referencing an object, refer to another object by its name:

```
space:
 name: PhoneCalls-space

spaceRootLibrary:
 name: PhoneCalls-space-root-library
 space: PhoneCalls-space
```

Use a path-like syntax to refer to an object when its name is not unique:

```
applications:
- name: app1
- name: app2

space:
- name: default-space
 application: app1
- name: default-space
 application: app2
```

```
spaceRootLibrary:
 name: space-root-library1
 space: app1/default-space # Space 'default-space' in application 'app1'
```

Some objects must already exist in InfoArchive before processing the configuration file and should not be changed. Include them in the configuration so that they can be referenced, but indicate that they are not to be touched:

```
fileSystemRoot:
 name: defaultFileSystemRoot
 configure: false # Only used for referencing, must already exist in IA
```

The server will use this construct during export for objects that are referenced, but fall outside the scope of the export (holding, application, or tenant).

To describe multiple objects of the same resource type, use the plural form with a sequence:

```
auditEvents:
- name: partial_dispose
 type: aip
- name: apply
 type: retention_policy
- name: remove
 type: retention_policy
- name: apply
 type: hold
- name: remove
 type: hold
```

Or use the name of the object as a YAML key:

```
stores:
 file_store:
 fileSystemFolder: PhoneCalls-folder
 type: filesystem
 status: online
 storeType: regular
 result_store:
 fileSystemFolder: PhoneCalls-result-folder
 type: filesystem
 status: online
 storeType: result
```

The singular form and both plural forms are supported on all levels, not just at the top.

Some objects have content: the value of some of their properties are coherent files in and of themselves, like XQueries or XML Schemas. Include them in the configuration using a content object and a literal block:

```
appExportPipelines:
 PhoneCalls-search-results-xsl-with-content-gzip-pipeline:
 description: gzip envelope for xsl csv export
 outputFormat: csv
 content:
 format: xml
 text: |
 <p:declare-step version="1.0" xmlns:p="http://www.w3.org/ns/xproc"
 xmlns:ia="http://infoarchive.emc.com/xproc" name="main">
 <p:input port="source" sequence="true" />
 <p:input port="parameters" kind="parameter" />
 <p:input port="stylesheet" />
 <p:option name="xslResultFormat" required="true" />
 <ia:filter-search-results exportContent="true" name="filtered">
 <p:input port="source">
```

```
<p:pipe step="main" port="source" />
</p:input>
</ia:filter-search-results>
<ia:xslt name="xslt">
 <p:with-option name="xslResultFormat" select="$xslResultFormat" />
 <p:input port="source">
 <p:pipe step="filtered" port="result" />
 </p:input>
 <p:input port="stylesheet">
 <p:pipe step="main" port="stylesheet" />
 </p:input>
</ia:xslt>
<ia:gzip>
 <p:input port="source">
 <p:pipe step="xslt" port="result" />
 <p:pipe step="xslt" port="resources" />
 <p:pipe step="filtered" port="content" />
 </p:input>
</ia:gzip>
<ia:store-export-result>
 <p:with-option name="format" select="$xslResultFormat" />
</ia:store-export-result>
</p:declare-step>
```

The content object can either be a sequence, for those objects that support multiple content objects, or it can be a single object.

The advantage of in-lining content this way is that the configuration is self-contained. The downsides, however, are that the configuration file can become quite large and that it mixes different media types (YAML, XML, XSLT, XML Schema, XQuery, XProc, HTML), which is not well-supported by editors.

To mitigate these issues, place the content in an external resource:

```
appExportPipelines:
 PhoneCalls-search-results-xsl-with-content-gzip-pipeline:
 description: gzip envelope for xsl csv export
 outputFormat: csv
 content:
 format: xml
 resource: search-results-xsl-with-content-gzip-pipeline.xpl
```

The resource must be resolvable from the configuration's location: in the same ZIP file or part of the same multipart HTTP request when submitted to an IA Server, or in the same directory when processed by a client like the SDK.

## Including Other Configuration Files

It is possible to import other resources from the YAML configuration files so that larger configurations can be split up.

Using the include mechanism, it is possible to include other configuration files, such as

- <INFOARCHIVE\_ROOT>/examples/applications/<APPLICATION\_NAME>/config/custom-presentations/configuration.yml
- <INFOARCHIVE\_ROOT>/examples/applications/<APPLICATION\_NAME>/config/exports/configuration.yml
- <INFOARCHIVE\_ROOT>/examples/applications/<APPLICATION\_NAME>/config/searches/configuration.yml

## Using Default Values

Some properties can be omitted from the configuration, in which case they take on default values. This reduces the amount of boilerplate code and duplication in the configuration.

Omit default references when there is only a single object defined of the appropriate type:

```
space:
 name: PhoneCalls-space

spaceRootLibrary:
 name: PhoneCalls-space-root-library
 # This can be omitted:
 # space: PhoneCalls-space
```

Or when there are multiple objects and only one is marked as the default:

```
stores:
 file_store:
 default: true # This object is the default when no reference is specified
 fileSystemFolder: PhoneCalls-folder
 type: filesystem
 status: online
 storeType: regular
 result_store:
 fileSystemFolder: PhoneCalls-result-folder
 type: filesystem
 status: online
 storeType: result

holding:
 name: PhoneCalls
 # These can be omitted:
 # sipStore: file_store
 # ciStore: file_store
 # xmlStore: file_store
 # renditionStore: file_store
 # managedItemStore: file_store
 # xdbStore: file_store
 # logStore: file_store
```

Many objects have properties that take on the same value in a large percentage of cases. Omit the following properties from the configuration if you want the properties to retain their respective default values:

Object	Property	Default Value
appExportPipeline	envelopeFormat	gzip
	includesContent	true
	inputFormat	ROW_COLUMN
appExportConfiguration	exportType	asynchronous

Object	Property	Default Value
full.text.index	convert.terms.to.lowercase	true
	filter.english.stop.words	false
	include.attributes	false
	index.all.text	true
	optimize.leading.wildcard.search	true
	support.phrases	false
	support.scoring	false
	support.start.end.token.flags	false
holding	ciHashValidationEnabled	true
	keepSipAfterCommitEnabled	false
	logStoreEnabled	true
	pdiXmlHashEnforced	false
	pdiXmlHashValidationEnabled	true
	syncCommitEnabled	true
	xdbMode	PRIVATE
ingest	processors	There are a lot of default values
ingestNode	enumerationCutoffDays	30
	enumerationMaxResultCount	10
	enumerationMinusRunning	true
	logLevel	INFO
path.value.index	buildWithoutLogging	false
	compressed	false
	concurrent	false
	uniqueKeys	true
query	resultRootElement	result
queryQuota	aipQuota	0
	aiuQuota	0
	dipQuota	0

Object	Property	Default Value
receiverNode	logLevel	INFO
	sips	<ul style="list-style-type: none"> <li>- format: sip_zip</li> <li>extractorImpl:</li> <li>com.emc.ia.reception</li> <li>.sip.extractor.impl</li> <li>.ZipSipExtractor</li> <li>- format: eas_sip_zip</li> <li>extractorImpl:</li> <li>com.emc.ia.reception</li> <li>.sip.extractor.impl.Leg</li> </ul>
store	status	ONLINE
	storeType	REGULAR
	type	FILESYSTEM

A property with a default value cannot be set to an empty value by omitting it from the configuration. Instead, set it explicitly to ~ or null:

```
order:
 name: PhoneCalls-order
 priority: 1
 retentionPolicy: ~ # Should have no value, prevent default from being inserted
 duration: 86400
```

## Using Properties Files

It is possible to use properties files to move certain settings outside the YAML files. The samples already make use of these. Within the YAML files, property references can be used in values, such as:

```
application:
 name: ${application.name}
```

The values are then looked up in the properties files. Initially, they are looked up in the <INFOARCHIVE\_ROOT>/examples/applications/<APPLICATION\_NAME>/config/configuration.properties file. If no value for the property can be found there, the property value is looked up in the <INFOARCHIVE\_ROOT>/examples/applications/default.properties file that is stored in the folder). Other property files on other folder levels would also be inspected for properties.

It is possible to enter defaults for properties for the case the property value is not defined in any property file. This is done by entering a colon after the property name, followed by the default value:

```
application:
 name: ${application.name:Default Application Name}
```

The sample applications make use of the syntax \${propertyName:}, which means an empty value (no value) is used when the property cannot be found.

The property files themselves are text values with a standard propertyName=PropertyValue syntax. Refer to other properties within the properties file with the same mechanism described above.

When applications, holdings, and searches are exported from InfoArchive using IA Shell or IA Web App, it will result in a zip file that contains a property file. As the web interface also indicates, depending on the goal of the import, edit the `configuration.properties` within the zip file before performing an import of that exported configuration.

## Working with Namespaces and Queries

Queries form an important part of the configuration. Queries are separated into two parts:

1. We refer to namespaces by their prefix. Each referenced namespace will be translated into a declare namespace line.
2. The second part is the rest of the XQuery. In some cases, this is simply an XPath, which is easier to understand for XQuery novices than the whole thing.

The two parts are combined into a proper XQuery behind the scenes. Specify the parts using the `namespace` and `text` properties:

```
namespaces:
- prefix: sip
 uri: urn:x-emc:ia:schema:sip:1.0

xdbLibraryPolicy:
 name: PhoneCalls-xdb-library-policy
 aipQuota: 10
 aiuQuota: 100
 closeMode: close hint date
 closeHintDateQuery:
 namespace: sip
 text: /sip:sip/sip:dss/sip:production_date/text()
 pKeyQuery:
 namespace: sip
 text: year-from-dateTime(xs:dateTime(/sip:sip/sip:dss/sip:production_date/text()))
 closePeriod: 2
```

Namespaces are referenced by `prefix` rather than by name. A query can specify one using the singular namespace or multiple using the plural namespaces and a sequence.

## Preventing Overwrites

There are use cases where you do not want to blindly overwrite values (passwords, for example), or do not have the permission to update objects. For such use cases, you can add a special `configure` property to the object that indicates if and how the object should be configured:

Value of Configure Property	Description
create or update	When the object does not exist, create it. Otherwise, update it to the given properties.  This is the default and can be omitted.
create	When the object does not exist, create it. Otherwise, do nothing.

Value of Configure Property	Description
use existing	When the object does not exist, raise an error. Otherwise, use it.  In this case, only the <code>name</code> property is used to look up the object, all other properties are ignored.
ignore	The configuration item should not be processed at all.

## Configuring Import/Export Functionality

InfoArchive provides the opportunity to export the configuration of an application, holding or search in either a ZIP or YAML format:

- The exported ZIP file contains:
  - Configuration resources in a YAML format,
  - Data model files,
  - Search definitions,
  - Result list custom presentations,
  - Result list export definitions, and
  - A `configuration.properties` file with system-specific settings (for example, system paths).
- The exported YAML file does not contain dedicated resources or files but, instead, the application's configuration is presented as one large YAML file.

When the export is done, the output contains properties that you may want to update/ review before importing. This allows you to change settings, such as passwords and paths, that vary from system to system. When exporting to a ZIP file, it contains a `configuration.properties` file in the top level of the ZIP file that you can edit. For both ZIP and YAML file exports, the YAML output contains property references that may need to be reviewed, especially when the import is performed on another system from the export.

For security reasons, an exported configuration never contains credentials and may contain system-specific settings that must be updated before you import into another system.

The number of artifacts inside the exported configuration is different, and depends on two factors:

- The export level, which can be:
  - System
  - Tenant
  - Application

- Holdings
- Searches
- The property that is passed to export: `create`, `use existing`, `create or update`, `ignore` allow you to manage the presence or absence of some objects in the result list configuration YAML, and the way the resources are created during the import process. By default, the export is completed with the `create or update` property.

InfoArchive allows the opportunity to import the configuration of an application, holding or search in either a ZIP or YAML format:

The configuration is passed to InfoArchive and, during the import process, resources can be created, updated, or skipped. Resource creation depends on the `configure` property that is specified for a single resource in the YAML configuration. The property can be:

- `create`: If the object does not exist during the import, it is created. Otherwise, nothing is done.
- `create or update`: If the object does not exist during the import, it is created. Otherwise, it is updated.
- `use existing`: If the object does not exist during the import, then an error is reported.

The following list outlines the places where to call the import and export configuration functionality in InfoArchive.

- **REST API Server:** InfoArchive provided REST API that can be consumed by any REST client. Please, refer to REST documentation.
- **IA Shell:** The `import` and `export` commands in IA Shell allow for flexible configuration of the operations. The export can be done for system, tenant, application, holding, and search levels. There is a possibility to use all possible export scopes for configuration. Refer to the *InfoArchive Shell Guide* for more information about these commands.
- **IA Web App:** User can export and import: application, search, and holding. There are the restrictions from the user interface on export for applications and holdings: only `create or update` scope of export is allowed, as missing objects are created and existing objects are updated. As for the search, the export scope is `use existing`, which allows the user to achieve the need to have the search self-contained in an exported file. The only format for import and export is ZIP, as it the most convenient format. If more flexible functionality is required, then use IA Shell or REST API. After the import, a notification message informs the user about the created/skipped resources. It worth preserving this information for a bit of time during configuration debugging on development environment.

**Note:** Before importing/exporting configurations, be careful when you provide `configure` property for the commands and resources. Also be careful when you change the exported functionality prior to importing it back. Manual changes can make the configuration inconsistent and/or wrong.

# Chapter 4

---

## Searches in InfoArchive

### What is Search?

Searching is the primary method that users use to access data that has been ingested by InfoArchive.

Searches work for both SIP- and table-based archives, although the way searches are designed are slightly different:

- For a SIP archive, a search is associated with an archive information collection (AIC) and a query configuration. SIP-based searches require the name of the AIC and QueryConfiguration, which determines the criteria and results for the search.
- For a table archive, a search is associated with a schema or table. Table-based searches require a query (built upon an XQuery template), as well as the schema or table the search will be executed upon. The Developer must understand XQuery and how it is related to search design. The designer of a search needs to ensure:
  - The query is valid.
  - The parameters used are accurate.
  - The correct binding for elements is used in the result set.

Refer to [Appendix B – XQuery Best Practices](#) for more information.

InfoArchive allows for backward compatibility for search creation. All searches created using 4.0 and all sample searches will run. The EXPORT option is available at run time from the main result grid (some or all rows selected).

### How Searches Work in a SIP Archive

InfoArchive uses a two-tiered approach to searches. The following happens when a search is executed:

1. Tier 1: The system locates the packages (AIPs) that might contain results.  
This process is aided by the use of partition keys, which quickly narrow the scope of the executed search. Refer to [Using Partition Keys](#) for more information.
2. Tier 2: The system scans those packages for individual results (AIUs) via the use of indexes. Refer to [Configuring Indexes](#) for more information.

# Who Uses the Search Functionality?

All user roles can execute searches with the exception of:

- Administrator,
- IT Owner, and
- The Business Owner can execute searches.

## Composition and Configuration

The Developer is able to create a new search from scratch.

While a search is being created, it remains in Draft status, which means only the Developer can see the search. Other user roles can only access a search when the status has been updated to Ready.

The Developer is able to:

- Duplicate a search set to create a new search;
- Export the search set to a .zip file and import the search set into a similar application that has the same schema.
- The search developer is also able to configure for a search if it is possible to export the search results or create a collection from the search results.

## Use of Search

The End User role can only execute searches against ingested data from a decommissioned application or active archive.

The Retention Manager is able to apply a hold to all or some rows of the search results. Refer to [Applying a Hold to Search Results](#) for further information.

It is possible to filter search results. Both export and apply hold can act on the filtered set.

Depending how the Developer configured a particular set of search results, the E-Discovery Administrator is able to create a collection from a set of search results. Once a collection has been created, the E-Discovery Administrator is able to add the collection to a legal matter, which acts like a hold. Refer to [Creating a Collection and Applying a Legal Matter](#) for further information.

## Searches and Amazon Glacier

Amazon Glacier is a more inexpensive (albeit slower) cloud location and service. The Developer is able to move infrequently accessed data to Amazon Glacier's archival storage to save money on storage costs.

An application can be set up so that content is moved, first, to the Amazon S3 Simple Storage System before being stored in Glacier.

Once a search is executed, if the search results contains a content ID (CID) and is stored on Glacier, the content will not be available immediately for download or viewing.

The user, however, may request content restoration through InfoArchive search interface during runtime. Once restoration is requested, a restore order item will be placed in queue to run asynchronously and the user will be advised to check the **Background Requests** tab. When restoration is complete, the user is able to view or download the content. Note:

**Note:** When content is copied to S3, the length of time the content will be available depends on the system configuration.

When content is not available, the View and Download links are replaced by a warning sign. Also, there is a **RESTORE CONTENT** link on the top of the page. This link indicates that there is some unavailable content in the result list (main grid) or in an in-line or side panel. The service (IAS) polls for content availability according to the following attribute set in server application.yml file:

```
#Configuration for the order items
order:
 pollingDelayForRestorationOfflineContent: 900 (default 15 min.)
 orderItemRetentionDays: 1
```

 Content will take longer to be restored [RESTORE CONTENT](#)



Call start	Call end	Attachment 1
2001-11-02T21:06:39.104+...	2001-11-02T21:50:07.104+...	
2001-11-14T05:21:15.104+...	2001-11-14T05:45:23.104+...	
2001-11-26T10:17:18.104+...	2001-11-26T10:45:05.104+...	
2002-11-18T16:22:04.104+...	2002-11-18T16:35:42.104+...	
2002-11-16T16:57:42.104+...	2002-11-16T17:26:37.104+...	
2002-11-30T19:38:34.104+...	2002-11-30T20:13:49.104+...	

# Preparation for Searches

## How Search Differs for Table and SIP Archives

Search composition is slightly different depending on whether the Developer is creating a search for a table or SIP archive.

While composing a table search, the Developer uses a **Query Editor** tab to enter XQuery code that will perform the actual query against the data stored in the xDB database.

On the other hand, SIP searches use a wizard-based approach to search composition. For end users, however, searches for table and SIP archives look and feel the same. The Developer can also use manual steps, for example, when creating a form. The Developer can use the Add Form element or use Add Column and Add Field in the Result List or Result Detail tabs.

## Custom Exports

The InfoArchive XProc-based export mechanism enables custom export pipelines to control how search results are exported. Export configuration can be used to enable users to download search results in a different format with or without associated contents (attachments/BLOBs).

A number of common formats are available out of the box and are installed. To install tenant-level export configurations, the Administrator needs to run ANT from the `<INFOARCHIVE_ROOT>/examples/legacy-ant-applications` directory.

At runtime, users are given a choice of exporting search results according to the options selected during composition time. When exporting at the tab level in the detail section, one row of data (selected row) is available for the export operation. From the main search result screen, the user is required to select an export option from the Export menu, which contains options selected by the Developer during search composition.

The download option also depends on the browser settings. For instance, in some browsers, a save as menu may be presented or the download may start when the download button in the background items listing page is selected. The gzip option downloads with a `.gz` extension. In case of included content or multiple files in the archive for other reasons, the content of the `.gz` file will be a tar file and the full file extension will be `.tar.gz`. It is also possible to download as `.tar` or `.zip` files instead, assuming the corresponding export pipelines are enabled.

This mechanism utilizes three configuration components to set up custom export pipelines:

- [export pipeline](#)
- [export transformation](#)
- [export configuration](#)

## Export Pipeline

An export pipeline defines the underlying XProc pipeline as well as additional InfoArchive-specific metadata. The XProc pipeline follows regular XProc syntax while using InfoArchive-specific or custom XProc steps. Its input consists of the search results to export and an optional stylesheet. It can also declare pipeline options and stylesheet parameters. The search results will be provided during runtime by InfoArchive. If declared, the stylesheet as well as options and/or parameters need to be defined in the export configuration mentioned below. An export pipeline is intended to be used by one or more export configurations, which may define different stylesheet and/or options/parameters to control the export result, without having to duplicate the pipeline.

## Transformation

One of the options available for customizing the export pipeline is to perform a stylesheet transformation on the search results. This includes the ability to generate a PDF by transforming to a formatting object (FO) using an XSL-FO stylesheet, and converting the result to PDF using the 'PDF' InfoArchive custom XProc step. The export transformation references a zip file containing both the XSL stylesheet and any optional supporting resources (such as, for example, logos css, javascript, fonts, etc.) required either during the stylesheet transformation or by the end result. All files in a sub-directory called 'resources' are made available to the pipeline as output of the transformation step(s) so that these can be included in the export artifact(s). The other files in the zip file are only available during the transformation and not exposed to the pipeline afterwards.

## Export Configuration

When selecting available exports for an individual search, a reference to the corresponding export configuration is assigned to the search. An export configuration references a particular export pipeline and defines values for any pipeline options and/or stylesheet parameters used by it. In case of a pipeline that requires a transformation, it is the export configuration that maintains the reference to the transformation to use as well as defining the input port to use when binding the stylesheet to the pipeline. Each export configuration represents a particular type of export, dedicated to a particular use case. InfoArchive defines a number of generic out-of-box tenant-level export configurations, but applications may define application-specific export configurations. By defining the use case specific details of particular exports in the export configurations, export pipelines and export transformations can potentially be reused between several export configurations, reducing the amount of duplication of such configuration information.

## Custom Presentation

InfoArchive allows customers to:

- Use a custom HTML template to render a page full of search results. Note that using custom HTML is not the primary way to create or design the result listing page. The result list in a grid, columns of data that presents the results of a user's search.
- Communicate selection, sorting, page navigation, invocation of nested searches, downloads and viewing in a viewer from the rendered view.

## Writing XProc Pipelines to Export Search Results

InfoArchive provides a number of custom XProc steps that can be used to write XProc pipelines to export search results.

All steps use the namespace `http://infoarchive.emc.com/xproc`, which needs to be specified when referring to them in an XProc pipeline.

Name	Input	Output	Description
filter-search-results	0..* XML search results	0..* XML search results 0..* non-XML content	Outputs filtered input ROW_COLUMN search results based on ResultMaster and export settings, as well as optionally content referenced by those filtered search results.
search-results-content-export	0..* XML search results	0..* non-XML content	Exports content referenced by input ROW_COLUMN search results.
raw-search-results-content-export	0..* XML search results	0..* non-XML content	Exports content referenced by input RAW_XML search results.
search-results-csv	0..* XML search results	1 csv file 0..* non-XML content	Converts filtered input ROW_COLUMN search results into single CSV Optionally exports content referenced by search results.

Name	Input	Output	Description
search-results-json	0..* XML search results	1 json file 0..* non-XML content	Converts filtered input ROW_COLUMN search results into single JSON file. Optionally, exports content referenced by search results.
search-results-xml	0..* XML search results	1 XML file 0..* non-XML content	Converts filtered input ROW_COLUMN search results into single XML file. Optionally exports content referenced by search results.
xslt	1 XML (combined search results) 1 XSL stylesheet	1 transformed output 0..* resources	Transforms single XML input consisting of all search results combined using custom XSLT stylesheet.  Exports XSLT result as well as optional resources associated with the transformation.
pdf	1 XML (combined search results) 1 XSL-FO stylesheet	1 pdf 0..* resources	Transforms single XML input consisting of all search results combined using custom XSL-FO stylesheet into PDF.  Exports PDF result as well as optional resources associated with the transformation.
gzip	0..* 'anything'	1 (.tar).gz file	Generates a single gzip file from all provided input. In case of multiple inputs, implicitly generates intermediate tar file.

Name	Input	Output	Description
tar	0..* 'anything'	1 .tar file	Generates a single tar file from all provided input.
zip	0..* 'anything'	1 zip file	Generates a single zip file from all provided input.
store-export-result	1 'anything'		<p>Stores the export result into the 'RESULT' InfoArchive content store associated with the application.</p> <p>This step is necessary for the result to be downloadable from the UI.</p>
file-system-copy	1 'anything'		<p>Copies the export result to the specified (remote) filesystem path (which must be accessible to InfoArchive).</p> <p>This for example allows InfoArchive to export to a network location, but the result cannot be downloaded through the UI.</p>
xquery	0..* XML search results 1 XQUERY	0..* xquery result text 0..* xquery result XML nodes	<p>Executes the provided xquery on each provided input, which can be either all search results combined or each search result individually.</p> <p>All XQuery results are output as a string, XML node XQuery results are also separately output as XML DOM nodes.</p>

## Referencing External Resources

There are use cases where an XProc step needs to reference external resources. Primary examples are the XSLT and PDF steps where, besides the stylesheet used to perform the transformation, such additional external resources could, for example, include logos, fonts, css, etc. XProc supports this by means of secondary inputs to either the pipeline or individual steps. However, such inputs must be explicitly named and specified in the pipeline.

InfoArchive uses transformations that are assigned to a pipeline by means of an export configuration, which also provides values for options and/or parameters declared by the pipeline. A transformation essentially refers to a single zip file archive containing all related external resources that are required for a particular transformation or type of pipeline execution, where only a single root resource is passed to the pipeline. Any additional secondary external resources can be resolved relative to the path of this root resource. InfoArchive ensures that all resources in the zip file associated with the transformation can be resolved, while it is up to each individual step that needs the transformation to actually do so.

For the XSLT and PDF steps/pipelines, there is a single (secondary) input defined, which is called a stylesheet, and is populated with the name/path of the root file of the transformation. This stylesheet, in turn, optionally references other external resources included in the same transformation zip file using a path relative to that of the stylesheet file itself.

For other (custom) steps that need to reference external resources, but do not really have a root resource, the root resource can be set to any of the available external resources whose paths can be used to resolve the other external resources. The step still requires you to declare a single secondary input port on which this root resource will be provided, but it can then resolve any of the necessary external resources against that resource without having to explicitly use that declared root resource.

## Restrictions to the InfoArchive Search Result Export XProc Pipelines

Although XProc itself provides various official out-of-the-box steps, due to the restrictions imposed by the XProc specification, most if not all of these out-of-the-box steps are incompatible with InfoArchive's search result export functionality. The official XProc documentation can be used for reference purposes when writing custom pipelines, but none of the official out-of-the-box XProc steps should be expected to work for InfoArchive unless explicitly stated otherwise. Therefore, it is recommended that you limit InfoArchive pipelines to using only the InfoArchive custom XProc steps described in this section.

## Extending InfoArchive XProc Export Functionality

InfoArchive supports exporting search results by means of executing an XProc pipeline. A number of custom XProc steps are provided out-of-the-box that can be used to load custom XProc export pipelines (configuration) to tweak the corresponding export functionality. It is also possible to deploy additional custom XProc steps (logic) to further extend and customize the out-of-the-box InfoArchive export functionality. Because this mechanism involves deploying custom code into an InfoArchive distribution, there are certain requirements with regards to how the deployed artifacts need to be

structured, as well as restrictions with regards to what can be supported. Within the boundaries of those requirements and restrictions, however, the logic that can be deployed using such an extension is entirely up to the customer.

Once a custom XProc export extension has been deployed, it will not automatically affect the available export options. It is necessary to first configure custom export pipelines that actually use the newly deployed custom XProc steps, and assign those custom pipelines to individual searches, before the extension can actually be used. The process to do so, however, is (once the extension has been successfully deployed) identical to the process of configuring custom pipelines using only out-of-the-box custom XProc steps.

A custom export pipeline that depends on custom XProc steps provided by an extension will only function properly when the relevant extensions have been deployed to InfoArchive. In the event that it fails, an error message is issued that indicates the problem with one of the custom XProc steps. The most likely cause is that the relevant extension has not yet been deployed. It is also possible that there is an actual problem with the custom step itself, caused by either a bug in the corresponding Java code or a missing or conflicting dependency.

An InfoArchive custom XProc export extension is, essentially, a single JAR artifact, optionally, together with any third party dependencies, which could either be embedded within the JAR or included as separate JARs instead. However, there are some requirements related to how to instruct InfoArchive to use the custom XProc steps provided by the extension.

1. The custom XProc step logic for each individual step must be implemented in a Java class that implements the `com.emc.documentum.xml.xproc.pipeline.model.step.AtomicStepBody` interface of Calumet.
  - a. It is strongly recommended to extend the class `com.emc.documentum.xml.xproc.pipeline.model.step.AbstractAtomicStepBody` of Calumet instead of implementing the interface from scratch, however, as this class takes care of some mandatory initialization that would otherwise have to be implemented explicitly, as well.
2. A separate XProc step declaration file is required for each custom XProc step included in the extension.
  - a. This declaration file is a so-called XProc external library, for which the syntax is described in the documentation of Calumet.
  - b. These files are packaged in the same JAR as the corresponding logic (Java classes), but as long as they are on the same classpath, it is sufficient.
3. An InfoArchive-specific custom XProc steps configuration file called `META-INF/{prefix}-custom-xproc-steps.xml` is included in the jar. This XML adheres to the corresponding InfoArchive-specific schema.

The value of the `{prefix}` placeholder identifies the source or purpose of the collection of custom XProc steps it describes.

  - a. At least one such file is required per XProc export extension JAR, though more can be included as well, as these files are used to identify the custom XProc steps provided by the extension. Only steps explicitly declared in such a file will be registered with InfoArchive.
    - i. The use of different values for this prefix allows you to define multiple files within the same JAR, each describing a different subset of custom XProc steps, to allow additional organizational freedom.

- ii. All such files will be processed sequentially. Each step should not be declared in more than one such file to avoid potential conflicting definitions and problems during resolution.
- b. Each step needs to be represented by a dedicated entry declaring both:
  - i. The fully qualified Java class name providing the logic of the step mentioned in point 1.
  - ii. A 'classpath:' URI reference to the step declaration file mentioned in point 2 identifying it by means of an absolute path to this resource on the classpath.

There is no restriction to the number of extensions that provide custom XProc steps, nor the number of custom XProc steps provided by a single extension. The number of custom XProc steps provided in a particular extension and/or how many such extensions are deployed is entirely up to the customer.

InfoArchive utilizes the exact same mechanism to register custom XProc steps, with the only exception that the JAR providing these steps is not a separate extension but is, instead, embedded in InfoArchive.

The following examples, InfoArchive's own out-of-the-box custom xslt and zip steps are used to illustrate how to set up a separate extension for them, with the exception of the corresponding Java sources.

```
/com/emc/ia/search/export/xproc/steps/xslt-step.xpl
<p:declare-step version="1.0" type="ia:xslt" xmlns:p="http://www.w3.org/ns/xproc" xmlns:ia="http://infoarchive.emc.com/xproc">
 <p:input port="source" primary="true"/>
 <p:input port="stylesheet"/>
 <p:input port="parameters" kind="parameter"/>
 <p:output port="result" primary="true"/>
 <p:output port="resources" sequence="true"/>
 <p:option name="xslResultFormat"/>
</p:declare-step>

/com/emc/ia/search/export/xproc/steps/zip-step.xpl
<p:declare-step version="1.0" type="ia:zip" xmlns:p="http://www.w3.org/ns/xproc"
 xmlns:ia="http://infoarchive.emc.com/xproc">
 <p:input port="source" primary="true" sequence="true"/>
 <p:output port="result" primary="true"/>
</p:declare-step>
```

The following is an InfoArchive custom XProc step configuration file declaring custom xslt and zip steps:

```
/META-INF/ia-custom-xproc-steps.xml
<custom-xproc-steps xmlns="http://infoarchive.emc.com/xproc/config">
 <step>
 <class-name>com.emc.ia.search.export.xproc.IAXSLTStep</class-name>
 <declaration>classpath:/com/emc/ia/search/export/xproc/steps/xslt-step.xpl</declaration>
 </step>
 <step>
 <class-name>com.emc.ia.search.export.xproc.ZIPStep</class-name>
 <declaration>classpath:/com/emc/ia/search/export/xproc/steps/zip-step.xpl</declaration>
 </step>
</custom-xproc-steps>
```

Directory listing for a jar containing all of the above mentioned/referenced files:

```
/com/emc/ia/search/export/xproc/IAXSLTStep.class
```

```
/com/emc/ia/search/export/xproc/ZIPStep.class
/com/emc/ia/search/export/xproc/steps/xslt-step.xpl
/com/emc/ia/search/export/xproc/steps/zip-step.xpl
/META-INF/ia-custom-xproc-steps.xml
```

The InfoArchive distribution includes one sample XProc export extension called `xproc-ftp`. This extension is set up to be built by means of gradle using the IA Server extension gradle plugin, and serves as a sample to check out how to build custom extensions from scratch. It includes a sample IA Shell configuration script to configure an export pipeline and corresponding export configuration to use the custom step provided by this extension. Running this IA Shell configuration script using the IA Shell will configure InfoArchive accordingly, after which this export pipeline is available to be assigned to any of the available searches to try it out.

## Search Composition for the Developer

### Overview

This section is dedicated to the Developer role, and explains how to work with searches and XQuery modules.

The following table outlines which form fields allow the search developer to configure default values:

Form Field	Control Type	Does Field Allow Default Value Configuration?
------------	--------------	-----------------------------------------------

Input	Text	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> No</li> </ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> Yes</p> <p><b>Regex:</b> Yes</p> <p><b>Min/Max Characters:</b> Yes</p> <p><b>Multiple Values:</b> Yes</p> <p><b>Default Value:</b> Yes</p>
Number	Number	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> No</li> </ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> Yes</p> <p><b>Regex:</b> Yes</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> Yes</p>
Number Range		<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> No</li> </ul>

Form Field	Control Type	Does Field Allow Default Value Configuration?
Checkbox	Check Box (Boolean Input)	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> No</li> </ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> Yes</p>
	Check Box Group	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> No</li> </ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> Yes</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
Radio Group	Radio Group	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> Yes</li> </ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> Yes</p>
Select	Single Drop -Down	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> Yes</li> </ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> Yes</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	Single List	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> Yes</li> </ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> Yes</p>
	Multiple Drop-Down	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> No</li> </ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> Yes</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	Multiple List	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> No</li> </ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> Yes</p>
Date	Date	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> No</li> </ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> Yes</p> <p><b>Default Value:</b> Yes</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	Date Range	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> No</li> </ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> Yes</p>
	Date Time	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> Yes</li> <li>• <b>Composite:</b> Yes</li> <li>• <b>Not a Search Criterion:</b> No</li> </ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> Yes</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	Date Time Range	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"><li>• <b>Single:</b> Yes</li><li>• <b>Composite:</b> Yes</li><li>• <b>Not a Search Criterion:</b> No</li></ul> <p><b>Data Binding:</b> Yes</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> Yes</p> <p><b>Hidden:</b> Yes</p> <p><b>Tool Tip:</b> Yes</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> Yes</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
Text	Text	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> No</li> <li>• <b>Composite:</b> No</li> <li>• <b>Not a Search Criterion:</b> No</li> </ul> <p><b>Data Binding:</b> No</p> <p><b>Label:</b> No</p> <p><b>Required:</b> No</p> <p><b>Hidden:</b> No</p> <p><b>Tool Tip:</b> No</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> No</p>
Container	Container	<p><b>Binding Type:</b></p> <ul style="list-style-type: none"> <li>• <b>Single:</b> No</li> <li>• <b>Composite:</b> No</li> <li>• <b>Not a Search Criterion:</b> No</li> </ul> <p><b>Data Binding:</b> No</p> <p><b>Label:</b> Yes</p> <p><b>Required:</b> No</p> <p><b>Hidden:</b> No</p> <p><b>Tool Tip:</b> No</p> <p><b>Prompt Text:</b> No</p> <p><b>Regex:</b> No</p> <p><b>Min/Max Characters:</b> No</p> <p><b>Multiple Values:</b> No</p> <p><b>Default Value:</b> No</p>

The following table outlines which form fields allow configuration flexibility to the search developer:

Form Field	Control Type	Does Field Allow Default Value Configuration?
Input	Text	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> Yes</li> <li>• <b>External Service URL:</b> Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> No</li> <li>• <b>Target Select:</b> No</li> </ul> <p><b>Values from XQuery:</b> No</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> No</li> <li>• <b>UTC:</b> No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> Yes</li> <li>• <b>Regex Pattern Not Match:</b> Yes</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>
	Number	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> No</li> <li>• <b>External Service URL:</b> No</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> No</li> <li>• <b>Target Select:</b> No</li> </ul> <p><b>Values from XQuery:</b> No</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> No</li> <li>• <b>UTC:</b> No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> Yes</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
Input (continued)	Number Range	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> No</li> <li>• <b>External Service URL:</b> No</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> No</li> <li>• <b>Target Select:</b> No</li> </ul> <p><b>Values from XQuery:</b> No</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> No</li> <li>• <b>UTC:</b> No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> Yes</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
Checkbox	Check Box (Boolean Input)	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> Yes</li> <li>• <b>External Service URL:</b> Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> No</li> <li>• <b>Target Select:</b> No</li> </ul> <p><b>Values from XQuery:</b> No</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> No</li> <li>• <b>UTC:</b> No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> No</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	Check Box Group	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> Yes</li> <li>• <b>External Service URL:</b> Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> No</li> <li>• <b>Target Select:</b> No</li> </ul> <p><b>Values from XQuery:</b> No</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> No</li> <li>• <b>UTC:</b> No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> No</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
Radio Group	Radio Group	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> Yes</li> <li>• <b>External Service URL:</b> Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> Yes</li> <li>• <b>Target Select:</b> No</li> </ul> <p><b>Values from XQuery:</b> No</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> No</li> <li>• <b>UTC:</b> No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> No</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
Select	Single Drop -Down	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> Yes</li> <li>• <b>External Service URL:</b> Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> Yes</li> <li>• <b>Target Select:</b> Yes</li> </ul> <p><b>Values from XQuery:</b> Yes</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> No</li> <li>• <b>UTC:</b> No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> No</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	Single List	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> Yes</li> <li>• <b>External Service URL:</b> Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> Yes</li> <li>• <b>Target Select:</b> Yes</li> </ul> <p><b>Values from XQuery:</b> Yes</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> No</li> <li>• <b>UTC:</b> No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> No</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	Multiple Select Drop-Down	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> Yes</li> <li>• <b>External Service URL:</b> Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> Yes</li> <li>• <b>Target Select:</b> Yes</li> </ul> <p><b>Values from XQuery:</b> Yes</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> No</li> <li>• <b>UTC:</b> No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> No</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	Multiple Select List	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> Yes</li> <li>• <b>External Service URL:</b> Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> Yes</li> <li>• <b>Target Select:</b> Yes</li> </ul> <p><b>Values from XQuery:</b> Yes</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> No</li> <li>• <b>UTC:</b> No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> No</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
Date	Date	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> Yes</li> <li>• <b>External Service URL:</b> Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> No</li> <li>• <b>Target Select:</b> No</li> </ul> <p><b>Values from XQuery:</b> No</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> Yes</li> <li>• <b>UTC:</b> Yes</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> No</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	Date Range	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> No</li> <li>• <b>External Service URL:</b> No</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> No</li> <li>• <b>Target Select:</b> No</li> </ul> <p><b>Values from XQuery:</b> No</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> Yes</li> <li>• <b>UTC:</b> Yes</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> No</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	Date Time	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> Yes</li> <li>• <b>External Service URL:</b> Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> No</li> <li>• <b>Target Select:</b> No</li> </ul> <p><b>Values from XQuery:</b> No</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> Yes</li> <li>• <b>UTC:</b> Yes</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> Yes</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> No</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	Date Time Range	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"><li>• <b>XQuery:</b> No</li><li>• <b>External Service URL:</b> No</li></ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"><li>• <b>Source Select:</b> No</li><li>• <b>Target Select:</b> No</li></ul> <p><b>Values from XQuery:</b> No</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"><li>• <b>Local:</b> Yes</li><li>• <b>UTC:</b> Yes</li></ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"><li>• <b>Required:</b> Yes</li><li>• <b>Out of Range:</b> No</li><li>• <b>Regex Pattern Not Match:</b> No</li></ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>

Form Field	Control Type	Does Field Allow Default Value Configuration?
Text	Text	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> No</li> <li>• <b>External Service URL:</b> No</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> No</li> <li>• <b>Target Select:</b> No</li> </ul> <p><b>Values from XQuery:</b> No</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> No</li> <li>• <b>UTC:</b> No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> No</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> No</li> </ul> <p><b>Text:</b> Yes</p> <p><b>Background/Border Color:</b> No</p> <p><b>Conditional Show:</b> No</p>
Container	Container	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• <b>XQuery:</b> No</li> <li>• <b>External Service URL:</b> No</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• <b>Source Select:</b> No</li> <li>• <b>Target Select:</b> No</li> </ul> <p><b>Values from XQuery:</b> No</p> <p><b>Time Zone:</b> No</p> <ul style="list-style-type: none"> <li>• <b>Local:</b> No</li> <li>• <b>UTC:</b> No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• <b>Required:</b> No</li> <li>• <b>Out of Range:</b> No</li> <li>• <b>Regex Pattern Not Match:</b> No</li> </ul> <p><b>Text:</b> No</p> <p><b>Background/Border Color:</b> Yes</p>

## Search

This section illustrates how to:

- Access searches stored in an application
- Create searches, including how to:
  - Export, import and duplicate a search
  - Create search sets and compose a search form

## Search Listing

Searches are stored in applications. To view an application's available searches, click **SELECT** for the desired application from the Application listing page

The **Search Forms** tab lists the application's available searches.

Each search may contain one or more search sets. The Developer can create different search sets, with each set having slightly different behavior in the actual search form and/or query and/or result. The Developer uses permissions make to make each search set available to different groups.

All searches appear in the **Search Forms** tab, regardless of whether the user can access the search or not.

When a search is in Draft mode, it is only visible to users with the Developer role

To locate a specific search, enter the name in the **Find a Search** field and press **Enter** on your keyboard.

Use the **Status** filter to filter the searches based on search status: Ready, Draft or All.

Some of the actions the Developer is able to undertake from the **Record Search** tab include:

- [Exporting the search](#) to a .zip file
- [Importing a search](#)
- [Creating a duplicate](#) of a search.

## Exporting a Search

The Developer is able to export a search:

1. Access the desired application.
2. For the search being exported, click the context menu and select **Export to ZIP file**.  
The contents of the search are exported into a .zip file.

**Note:** When importing or exporting nested searches, you only need to import/export the primary search. The rest will be automatically included.

## Importing a Search

Searches can be imported for a given application provided that no substantial changes were made to the application (for example, no updates were made to the SIP application's result configuration helper or to the table schema).

The search importation functionality is especially helpful because it allows you to import an existing search from one of the sample applications provided with InfoArchive and configure the imported search to suit your needs.

The following example illustrates how to import an existing search from the Ballball sample application into one of your own applications.

1. From the **Application** tab, **SELECT** the application the search is being imported into.
2. Click **+** and select **Import from ZIP file**.
3. Navigate to the location where the **.zip** file is located, select the search and click **Open**. Once selected, click **Search**.

The search name now appears in the list of searches.

**Note:** After importing a search, the search will be in Draft mode.

## Creating a Duplicate Search

The Developer is able to create a duplicate of an existing search and refine it to create an entirely new search.

1. Click the context menu for the search being duplicated and select **Create Duplicate**.
2. Enter the following information:
  - a. A unique **Search Name** for the new search form.  
The following special characters are not supported: '#', '<', '>', '\', '/', '!', '='.
  - b. A brief **Description** of the new search form.
  - c. Click **OK**.

The new search form appears in the list of search forms on the **Search Forms** tab.

3. To further refine the new search, click the context menu.

## Editing a Search

It is important to note that once any changes are made to a search in the 'Ready' mode will cause the search form to revert to 'Draft' mode.

1. Select the application in which the search being edited is stored.
2. Click the context menu for the search being edited and select **Edit**.
3. Edit the search, as desired.
4. Click **Save**.

## Creating a Search

This section illustrates how to view and create search sets. For table archives, there are examples of how to write XQuery, as well as use the database schema and table reference.

This section also documents how to compose a search form, including:

- How to add, move and resize fields; and
- Add containers to create conditional fields.

### Search Sets

Within a search, there may be one or more search sets.

Search sets provide a way to have different search forms and results for each of the different user roles. By default, search sets are available to any user that can execute a search. The Developer, however, can restrict a search set to specific groups when creating a search.

Search sets can be duplicated and renamed, and they can be deleted. A search has at least one search set. If a search includes only one search set, that set cannot be deleted.

### Viewing Search Sets

To view a search set, **SELECT** the desired application from the **Applications** tab. The available searches appear in a panel on the left side of the screen. Select a search.

In runtime, if there are multiple search sets contained in the search and the user role allows access to multiple sets, a drop-down list appears at the top of the search screen that allows the user to toggle between the search sets. The drop-down list is not displayed if a search only contains one search set.

While the Developer can view all search sets, other users can only view the search sets if:

- During the search composition phase, the Developer allows the specific group or multiple groups to access the search, or makes a search not restricted.
- The search status is Ready. If the search status is not set to Ready, only the Developer can access the search set.

### Creating a Search Set

Only users in the Developer role have the ability to create new searches.

For a table archive, determine which tables and columns will be searched and displayed in the results. Searching table-based archives requires creating an XQuery. Knowing which tables and columns exist will help in determining the query later on. This step assumes that you are familiar with the table schema and the different columns inside of the tables.

**Tip:** Once inside the table-based application, access the **Tables** tab to view a list of the tables. The **Tables** tab also provides information about retention, holds, number of records, etc. A panel on the right indicates the column name and the data types of those columns. The sample panel in the Query Editor also allows you to view a list of tables.

Whether you are creating a search set for a table or SIP archive, complete the following steps:

1. Select the application and select + and select **Create New**.
2. Enter the following information:

Field	Description
<b>Search Name</b>	Enter a name for the search that is unique within the application. The following special characters are not supported: '#', '<', '>', '\', '/', '!', '='.
<b>Description</b>	Enter a description for the search.
<b>Archival Collection</b>	<p>If you are creating a search set for a table archive:</p> <ul style="list-style-type: none"> <li>a. Select an xDB database that was created for the application.</li> <li>b. Select a Schema the search form will access when executed.</li> <li>c. Optionally, if your XQuery will only query a single table, select a table the search form will access when executed.</li> </ul> <p>If selected, the Designer is presented with a list of tables. Browse through the list or conduct a search to locate a specific table and click <b>Select</b>.</p> <p><b>Tip:</b> Specify the table if the search will be used to apply a hold to the results.</p> <p>If you are creating a search set for a SIP archive, this field allows you to select the AIC object for the search to be applied against. The AIC object can refer multiple holdings in which the search is performed.</p>
<b>Configuration</b>	If you are creating a search set for a SIP archive, select the query configuration that will be searched. The query configuration contains information to build the search from (for example, the list of queryable fields, the partition key, the indexed fields, the field format, the query quota, etc.) and the result page (the list of result fields).

Field	Description
Type	<p>Indicate whether you are creating a:</p> <ul style="list-style-type: none"> <li>Primary search, which can be accessed directly by users.</li> <li>Nested search, which is a search to be linked from other searches to retrieve additional information on a result.</li> </ul> <p>Most searches that you create will be primary searches.</p>
Search Form	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li><b>Composed:</b> You will use the interface to manually add fields to the search form. Fields can be further configured for the search form. If selected for a SIP search, once you click <b>NEXT</b>, you must complete the <b>Criteria</b> step of the search wizard. Go to step 3.</li> <li><b>Custom:</b> You will use the XForms Editor to create the search form. If selected for a SIP search, you must complete the <b>Results</b> step of the search wizard. Go to step 4. If selected for a table search, click <b>CREATE</b> to start composing the search form with the XForms Editor. Refer to <a href="#">Composing the Search Form Using the XForms Editor</a> for further information.</li> </ul>

If you are creating a search set for a table archive, click **CREATE**. You are now able to [compose the search form](#).

If you are creating a search set for a SIP archive, complete the following steps.

3. In the Criteria step:
  - a. In the Show in Form column, indicate which fields should be included in the search form.
  - b. In the Required? column, indicate which fields the user will have to complete prior to running the search.

**Tip:** It is not mandatory to make any of the criteria required, but not specifying any criteria will result in all of the AIUs being returned.

- c. Click **Next**.

The data displayed in the Criteria step was defined in an **.xml** file when the application was created and stored in the query configuration object. The data includes:

- Any fields returned by the query configuration object.
- The Partition Key column indicates the field designated as the partition key. While not obligatory, a search form that uses a partition key works more efficiently.

The Criteria page may be blank if information was not set up in the query configuration object.

4. In the **Results** step of the wizard:
  - a. Select the columns that will appear in the search results in the **Include In Results** column. .

- b. Indicate which column will comprise the default sorting of the search results by selecting a column name in the **Default Sort** column.

- c. Click FINISH.

Like the search criteria, the data displayed in the Results step was defined in an .xml file when the application was created and stored in the query configuration object.

How you proceed to compose the search form depends on the option you selected in the **Search Form** field:

- If **Composed** was selected, refer to [Composing the Search Form Using the User Interface](#)
- If **Custom** was selected, refer to [Composing the Search Form Using the XForms Editor](#).

## Composing the Search Form Using the XForms Editor

When creating a new search, the search developer has the option of using the user interface to manually add and configure fields to the form or using the XForm Editor to create the form.

If you are composing a search form for a table archive, you will need the XForm data to match the XQuery external parameter names, which allows the custom XForm to execute the search. XQuery external parameters reference table column names. For reference purposes, a side panel allows you to select a table to view the table columns while you compose the search form.

Conversely, if you are composing a search form for a SIP archive, a side panel contains the relevant fields, partition keys and field types for the relevant AIC the search form is being composed for.

While composing the form, click **Preview** at any time to see how the search form will appear in runtime.

InfoArchive's XForm Editor supports the XForms 1.1 standard with the following limitations:

- Form submission can only be the search criteria submitted to the IA Server and only use application and XML serialization. The following is an example of serialization:

```
<xforms:submission id="submit" method="post" serialization="application/xml"/>
```

- XForm Editor data needs to follow a specific structure:
  - For a table application, use the following structure:

```
<data>
 <searchCriterionName>
 any element structure ...
 </searchCriterionName>
 <searchCriterionName>
 any element structure ...
 </searchCriterionName>
 ...
</data>
```

- For a SIP application, use the following structure:

```
<data>
 <criterion>
 <name>...</name>
 <operator>...</operator>
 <value></value>
 <value></value>
 ...
 </criterion>
```

```
<criterion>
 <name>...</name>
 <operator>...</operator>
 <value></value>
 ...
</criterion>
...
</data>
```

- Even if the following XForm control elements are defined, they will not be rendered:
  - The `range` Element
  - The `upload` Element
- The following optional XForms features are not implemented in the Formula Engine:
  - The `p3ptype` attribute
  - The SHA-384 and SHA-512 hash algorithms
  - The `inputmode` attribute

## Composing the Search Form Using the User Interface

The search developer can use the **Search Form** tab to design a basic search form. Multiple form elements can be used from the Form toolbar.

Click **SAVE** at any time to save your changes. If you make a change to a saved form and attempt to navigate away without saving, IA Web App will prompt you to save your changes. Alternately, if you do not wish to save any of the changes you have made to the form, click **DISCARD CHANGES**, which will revert the form to how it was before you changed it.

### Adding Fields

The canvas you create forms on is divided into 12 columns. Once the Developer adds a form element to the canvas, whether it is a input/text field or radio group, the element occupies three columns.

When you click **Add Form Field**, a pallet is displayed that contains the various elements that can be added to the search form. These elements represent potential search criteria. Click an element to add it to the search form.

When elements are next to each other, the row height will be as tall as the tallest element on the row.

Elements are placed from left to right, top to bottom.

The group or container element can be used for organizing multiple form elements.

The text element is used for giving textual hints to the user of the form. The content is not submitted. The text element can be used as horizontal spacer, as illustrated in the screen shots below. The text element can also display HTML, including logos and images. The text element shown in the following two screen shots are examples of how the element can be used as spacer to center a paragraph, as well displaying content as HTML. HTML styling can be applied as long as

```
<html><body> ...</body></html>
```

are used, the content is displayed as HTML document.

The first is an example of the text element used during search composition:

The screenshot shows the InfoArchive search composition interface. At the top, there's a toolbar with buttons for 'Add Form Field', 'Select from Schema', and 'Preview'. Below the toolbar, there are three separate text boxes, each with its own set of edit controls (gear, arrows, close). The middle text box contains the following content:

**This is a sample Text**

- Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nam vero.
- Laboriosam quaerat sapiente minima nam minus similique illum architecto et!
- Incidunt vitae quae facere ducimus nostrum aliquid dolorum veritatis dicta!
- Tenetur laborum quod cum excepturi recusandae porro sint quis soluta!

The bottom text box contains the following content:

• Lorem ipsum dolor sit amet, consectetur adipisicing elit. Qui dicta minus molestiae vel beatae natus eveniet ratione temporibus aperiam harum alias officiis assumenda officia quibusdam deleniti eos cupiditate dolore doloribus!

Ad dolore dignissimos asperiores dicta facere optio quod commodi nam tempore recusandae. Rerum sed nulla eum vero expedita ex delectus voluptates rem at neque quos facere sequi unde optio aliquam!

Tenetur quod quidem in voluptatem corporis dolorum dicta sit pariatur porro quaerat autem ipsam odit quam beatae tempora quibusdam illum! Modi velit odio nam nulla unde amet odit pariatur at!

Consequatur rerum amet fuga expedita sunt et tempora saepe? Iusto nihil explicabo preferendis quos provident delectus ducimus necessitatibus reiciendis optio tempora unde earum doloremque commodi laudantium ad nulla vel odio?

The following demonstrates how the text field would appear to the user of the search during runtime:

The screenshot shows the InfoArchive search runtime interface. At the top, there's a header with 'All Text' on the left and a 'Search' button on the right. Below the header, there's a text area containing the following content:

**This is a sample Text**

- Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nam vero.
- Laboriosam quaerat sapiente minima nam minus similique illum architecto et!
- Incidunt vitae quae facere ducimus nostrum aliquid dolorum veritatis dicta!
- Tenetur laborum quod cum excepturi recusandae porro sint quis soluta!

• Lorem ipsum dolor sit amet, consectetur adipisicing elit. Qui dicta minus molestiae vel beatae natus eveniet ratione temporibus aperiam harum alias officiis assumenda officia quibusdam deleniti eos cupiditate dolore doloribus!

Ad dolore dignissimos asperiores dicta facere optio quod commodi nam tempore recusandae. Rerum sed nulla eum vero expedita ex delectus voluptates rem at neque quos facere sequi unde optio aliquam!

Tenetur quod quidem in voluptatem corporis dolorum dicta sit pariatur porro quaerat autem ipsam odit quam beatae tempora quibusdam illum! Modi velit odio nam nulla unde amet odit pariatur at!

Consequatur rerum amet fuga expedita sunt et tempora saepe? Iusto nihil explicabo preferendis quos provident delectus ducimus necessitatibus reiciendis optio tempora unde earum doloremque commodi laudantium ad nulla vel odio?

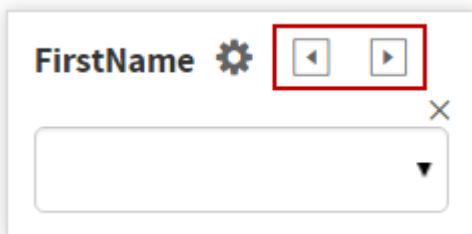
The following form controls are supported but are not listed in the control toolbar:

- Number control: Use the INPUT field type drop-down to choose number control
- Number range control: Use the INPUT field type drop-down to choose number range
- Single list: Use the SELECT element from the toolbar. Then, use control type and selection type drop-down selectors in the EDIT element dialog for a single select/multi-select drop-down.
- Multi-list
- Multi-drop-down

## Moving and Resizing Fields

To rearrange the elements of a search form, drag and drop an element to a new part of the form.

Each element contains arrows that allow you to resize the element. Click the arrows to make an element larger or smaller. The resizing arrows are highlighted in the red box in the following screen shot:



Once you start adding fields, click **Preview** to see how the search form will appear during runtime.

To remove an element from the search form, click 'X' in the top right corner of the element being removed.

## Adding Containers

The container element is the only element that allows you to add conditional fields to a search form. Within the container you can add one or more other fields. During runtime, the search form will display the container based on the criteria specified. Multiple conditions can be added to a search form.

The following example demonstrates how to create a search form that contains conditional elements. This example assumes that you have completed the steps outlined in [Creating a Search Set](#). In this case, you have created a SIP-based search set.

**Note:** The following example does not demonstrate how to create the sub-criteria section of the form.

1. In the **Search Form** tab, click the **Select** element.

What the search user selects in this field will determine the conditional fields that appear in the search form. First, you must refine this initial field.

2. Click to further refine the field.

Only enter the following information. Otherwise, leave the field blank:

Field	Description
Control Type	Select the <b>Drop-down</b> option.
Selection Type	Select the <b>Single</b> option.
Binding Type	Select the <b>Single</b> option.
Data Binding	Select the <b>data5</b> option.
Field Label	Enter the name for the field shown on the form (for this example, it is <b>Direction</b> ).

Field	Description
Values	Indicate the source of the values in the list. For this example, select <b>Specified Manually</b> .  Enter the following values in the Values field:  BD,Bank of Debitor BC,Bank of Creditor
Default Value	For this example, set the default value to <b>BC</b> .

Click **OK**.

- Add the following elements to the search form:

- Container
- Input

Notice how the Input element (marked as **Text Field**) is outside of the container. Click the Input element and drag it into the Container that you added.

- In the Container, click  to further refine the element.

Only enter the following information. Otherwise, leave the field blank:

Field	Description
Field Label	Enter <b>If Direction == "Bank of Creditor"</b> .
Background Color	Set the background color to <b>#BFDF5</b> .

In the Conditionally Show section, click  and set up the condition:

- Select a form element from the drop-down list (for this example, select **direction**).
- Select an operator from the drop-down list (for this example, select **Exact Match**).
- Specify a value (for this example, enter **BC**). This is the same value that was entered when you specified the values for the Direction field in step 2.
- Click **OK**.

In the Text Field that you dragged into the container, click  to further refine the element.

Only enter the following information. Otherwise, leave the field blank:

Field	Description
Type	For this example, select <b>Text</b> .
Binding Type	For this example, select <b>Single</b> .
Data Binding	For this example, select <b>minos</b> .
Field Label	For this example, enter <b>Transport</b> .

Click **OK**.

5. Add the following elements to the search form:

- Container
- Three Input elements

Drag the input elements into the second container that was added. When adding multiple elements to a container, they can be placed to the left or right of an element that is already in the container. The vertical blue line indicates where the element can go in the container.

6. In the Container, click  to further refine the element.

Only enter the following information. Otherwise, leave the field blank:

Field	Description
Field Label	Enter <b>If Direction == "Bank of Debitor".</b>
Background Color	Set the background color to #E9F4FC.

In the Conditionally Show section, click  and set up the condition:

- a. Select a form element from the drop-down list (for this example, select **direction**).
- b. Select an operator from the drop-down list (for this example, select **Exact Match**).
- c. Specify a value (for this example, enter **BD**). This is the same value that was entered when you specified the values for the Direction field in step 2.
- d. Click **OK**.

In the first Text Field that you dragged into the container, click  to further refine the element. Only enter the following information. Otherwise, leave the field blank:

Field	Description
Type	For this example, select <b>Text</b> .
Binding Type	For this example, select <b>Single</b> .
Data Binding	For this example, select <b>vacation</b> .
Field Label	For this example, enter <b>Transport Batch Reference</b> .

Click **OK**.

In the second Text Field that you dragged into the container, click  to further refine the element.

Only enter the following information. Otherwise, leave the field blank:

Field	Description
Type	For this example, select <b>Text</b> .
Binding Type	For this example, select <b>Single</b> .

Field	Description
Data Binding	For this example, select <b>pain-ID</b> .
Field Label	For this example, enter <b>Transport Document Number</b> .

Click **OK**.

In the third Text Field that you dragged into the container, click  to further refine the element. Only enter the following information. Otherwise, leave the field blank:

Field	Description
Type	For this example, select <b>Text</b> .
Binding Type	For this example, select <b>Single</b> .
Data Binding	For this example, select <b>pain-order</b> .
Field Label	For this example, enter <b>Transport Order Number</b> .

Click **OK**.

- On the **Search Form** tab, click **SAVE**.

## Customizing a Container

The following values can be customized in a container:

Field	Description
UI Control	A read-only field that is set to Container for the search form criteria.
Field Label	Enter the name for the field shown on the form.
Background Color	Select a background color for the container.
Border Color	Select a border color for the container.
Conditionally Show	<p>Allows you to configure how conditional elements will function in a search form.</p> <ol style="list-style-type: none"> <li>Select a field from the drop-down list.</li> <li>Select an operator from the drop-down list (for example, Exact Match or Does Not Match).</li> <li>Specify a value.</li> </ol> <p>Multiple conditions can be added to a container</p>

## Configuring Multiple Default Values for Checkboxes

The search developer is able to add multiple default values to checkboxes on a search form.

The following procedure uses the PhoneCalls sample application's FirstNameCheckbox search.

1. Click  to edit the FirstNameCheckbox search.  
In the **First Name** field, a series of checkboxes are displayed in the search form, each box includes a different name.
2. Click  to edit the **First Name** field.

The Checkboxes field contains the following information:

Mia, Mia  
Morgan, Morgan  
Maria,Maria  
Matthew,Matthew  
Molly,Molly  
Jack, Jack

The first instance of a name represents the raw value that is linked to the ingested data. The second instance of a name represents the value that will appear on the search form during run time.

3. In the **Default Value** field, enter the names that will be, by default, checked. When applying multiple defaults, use comma as a delimiter. For example:

Mia,Molly,Morgan

4. Click **OK**.

When the search user accesses the search in run time, the names added to the **Default Value** field will contain checkmarks.

## Adding a Default Value to a Text Field

The search developer is able to add a default value to an Input Field on a search form.

An input field can have the **Control Type** field set to one of the following:

- **Text**
- **Number**
- **Number Range**

The following example uses the PhoneCalls sample application's FirstName\_Operator search to add the default value of John to the search form.

1. Click  to edit the FirstName\_Operator search.
2. Click  to edit the settings for the Input Field that has no label associated with it.
3. To enter a single default value:
  - In the **Default Value** field, enter a first name that you want to set as the default value (for this example, we are entering John).

To enter multiple default values:

- a. Check the **Allow multiple values** box.

- b. Enter the values in the **Default Value** field. Use commas as a delimiter for multiple values.  
For example, enter Joe, Jane as default values.
4. Click **OK**.

## Adding a Default Value to a Number Field

An input field with the **Control Type** field set to Number does not have the ability to allow multiple values or multiple default values.

The following example uses the Certificates sample application's Certificate By Date Range search to edit the **Certificate Number** field to allow a default value.

1. Click  to edit the Certificate By Date Range search.
2. Click  to edit the settings for the Certificate Number field.
3. Enter the values in the **Default Value** field.
4. Click **OK**.

## Adding Default Values to a Number Range Field

When adding default values to an input field with the **Control Type** field set to Number Range, you must enter:

- **Default From Value**
- **Default To Value**

## Adding Default Date Range Values to a Search Form

The search developer is able to add default values to a range of dates on a search form.

The following example illustrates how to update the Debut Date Range Search in the Baseball sample application.

1. In the Search Form tab of the Debut Date Range Search search set. click  to edit the **Debut** field.
2. Enter a date for the **Default From Value**.
3. Enter a date for the **Default To Value**.
4. Click **OK**.

When the search user accesses the Debut Date Range Search, the dates that you entered will be presented in the date range fields. The user is still able to navigate to a date in the past or future from the specified default dates.

## Adding Default Number Values to a Search Form

The search developer is able to add default values to a range of numbers on a search form. The 3\_Record Id Search in the Patent sample application contains an example of a search form that uses a default number range.

To enter default values for a number range:

1. When creating the search, the **Control Type** must be set to **Number Range**.
2. Enter a number for the **Default From Value** field.
3. Enter a number for the **Default To Value** field.

When the search user accesses the search, the default value numbers that you entered will be presented in the numerical fields. The user is still able to enter a number for the specified default values.

## Customizing Number Ranges

An Input element can be designated as an input field in which the user in runtime can enter:

- Text,
- A number, or
- A number range

If it is a Number field, the user of the form cannot enter a character in that field.

If it is a Number Range field, the user of the form:

- Cannot enter a character in the field; and
- Must abide by the additional rules that were applied to the number range, as specified by the Developer when the field was configured (for example, the **From** field cannot be greater than the **To** field).

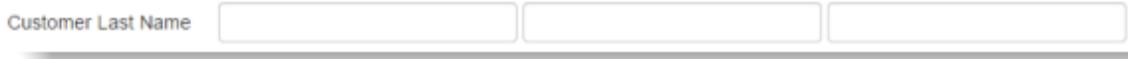
## Searching for a Value in Multiple Fields

The Developer is able to configure a search so a value entered as search criteria can be applied to multiple fields. For example, the name John could be a first or last name.

1. Click  for an Input field element.  
The Control Type must be set to **Single Select** or **Multi Select**.
2. In the Binding Type drop-down list, select **Composite**.
3. For the Composite Bindings:
  - a. Select the binding from the drop-down list.
  - b. Enter a display label.

For instance, the Developer may select the CustomerFirstName binding from the drop-down list. Therefore, First Name would be entered as the display label. The Developer clicks the + button to add

another binding. In the second binding, the Developer selects CustomerLastName. Therefore, Last Name would be entered as the display label for the second binding.



An operator can be applied to an extended field as well. Then the operator will be applied to each of the composite bindings.

**Note:** In selecting the relevant operator, the Developer must consider the data types and relevant operators before selecting the operator to apply to the chosen composite binding. For example, if one of the composite bindings is encrypted then the only suitable operator will be EQUAL or NOT EQUAL operators.

In the following screen shot, all first names or last names containing “Ja” will be matched and returned:

A screenshot of a search interface. On the left, a dropdown menu titled "Operator for Employee Name" is open, showing various search operators: Contains (selected), Exact Match, Does Not Match, Begins with (case sensitive), Begins with, Ends with, Contains (highlighted in blue), and Fulltext expression. On the right, there is a search input field containing "Ja" and a dropdown menu below it labeled "Search in First Name, Last Name".

## Configuring a Table-Based Search to Allow Wildcard Searches

The search Developer is able to configure an Input element in a search form to allow wildcard searches for a table archive. While there is more than one way to add this functionality, the following scenario illustrates one method whereby a text field (Last Name) is created and configured to search for an exact match or run a wildcard search.

1. Add the following elements to the search form:
  - a. Input
  - b. Select
2. For the Select element:
  - a. Configure element to be a drop-down – Single Select field.
  - b. Name this field **Operator**.
  - c. Set the values for the drop-down list to be:  
wildcard, Wildcard  
equal, Exact Match

This allows the search user to select either a wildcard match or an exact match.

- d. Set the Data Binding for the Operator drop-down to **operator**.  
 It is important that the value set matches the XQuery code (see the XQuery code in step 4 below).
3. For the Input element:
- Name the Input element Last Name.
  - Set the Data Binding to **lastName**.
4. Write the XQuery to include the above external variables. If the operation is set to wildcard, Then the XQuery code replaces '\*' with '.\*'. Otherwise, the following input is taken as is:

```

declare namespace ia = "urn:x-emc:ia:schema:fn";
declare namespace table="urn:x-emc:ia:schema:table";

declare variable $page external;
declare variable $size external;

declare variable $lastName external;
declare variable $operator external;

declare function local:getResultsPage($rows, $page, $size) {
 let $offset := $page * $size
 let $total := count($rows)
 return <results total="{ $total }"> {
 for $row in subsequence($rows, $offset + 1, $size)
 return $row
 } </results>
};

declare function local:getWhereClauseWildcard($param, $name, $opt) {
 if ($opt = "wildcard")
 then concat($name, " contains text ''", fn:replace($param/text(), "[*]\"", ".*"), "' using wildcards")
 else concat($name, " = '", $param, "'")
};

let $query :=
 concat("for $elem in /BASEBALL/MASTER/ROW where ",
 local:getWhereClauseWildcard($lastName, "$elem/NAMELAST",
 $operator),
 " return $elem")

let $rows :=
 for $elem in xhive:evaluate($query)
 return
 <row>
 <column name='lastName'>{ $elem/NAMELAST/text() }</column>
 <column name='firstName'>{ $elem/NAMEFIRST/text() }</column>
 <column name='birthYear'>{ $elem/BIRTHYEAR/text() }</column>
 <column name='birthMonth'>{ $elem/BIRTHMONTH/text() }</column>
 <column name='birthDay'>{ $elem/BIRTHDAY/text() }</column>
 <column name='weight'>{ $elem/WEIGHT/text() }</column>
 <column name='height'>{ $elem/HEIGHT/text() }</column>
 <column name='debut'>{ $elem/DEBUT/text() }</column>
 </row>
 return local:getResultsPage($rows, $page, $size)

```

During runtime, when the user enter 'Sm\*' into a search field and selects **Wild Card** from the Operator drop-down list, the search results will include names such as Smith.

## Customizing a Radio Group

A radio group allows the search user to select a particular radio button on the search form. Use the following values to configure a radio group:

Field	Description
<b>UI Control</b>	A read-only field that is set to Radio Group for the search form criteria.
<b>Binding Type</b>	<p>When you select a composite binding type, it allows the user to enter search criteria in multiple fields. For example, the user may want to search for Joe in First Name, Last Name or Middle Name fields.</p> <p>One of the following values can be selected:</p> <ul style="list-style-type: none"> <li>• <b>Single:</b> The selected list or drop-down list is bound to a single element. Select this value if the search user will search for data in a single field.</li> <li>• <b>Composite:</b> The selected list or drop-down list is bound to multiple elements. Select this value if the search user will search for data in multiple fields.</li> <li>• <b>Not a Search Criterion:</b> Select if you want to exclude the control value from the search criteria. For example, you only want to use this control value to hide or show a container, then you can choose this option to exclude this control value from the search criteria.</li> </ul>
<b>Data Binding</b>	<p>For table searches, enter the name of the variable in the xDB query.</p> <p>For SIP searches, define the value as a criterion. The name value has to match in order to use the criterion for the SIP search.</p> <p>For more information, see <a href="#">Example of Data Binding for a SIP Archive</a>.</p>

Field	Description
<b>Field Label</b>	Enter the name for the field shown on the form.
<b>Composite Bindings</b>	If <b>Composite</b> is selected as a <b>Binding Type</b> , select binding from the drop-down list. Essentially, you are selecting another field to bind the currently selected field to. Enter a disposition label that will appear on the search form. Refer to <a href="#">Allow a Value in a Search to be Included in Multiple Fields</a> for more information.
<b>Required</b>	Indicate whether an input value is required.
<b>Hidden</b>	Indicate whether this field should be hidden from the end user on the search form. This allows the designer to hard code values into a query, which the end user cannot change.
<b>Tooltip Text</b>	Enter concise, helpful information about the field that appears in a small “hover box” when the user hovers the cursor over the control.
<b>Radio Group</b>	Indicate the values offered in the radio group. For example, if the user is expected to select a particular city, the radio group would be configured, as illustrated:  value1, Chicago value2, Los Angeles value3, New York
<b>Default Value</b>	Enter one of the value text entries from the data specified in the Values field. For a single default value, only enter one value. For SELECT multiple selection, values should be separated by SPACE.
<b>Data Resolution</b>	Allows InfoArchive to expand a single search value according to a configured value map. Then, all expanded values are used as search criteria to execute search. Refer to <a href="#">Configuring Data Resolution for Supporting Search Value Expansion</a> for further information.

## Customizing the Select Element

When the Select element is added to a search form, you are essentially adding a list or drop-down list. During runtime, the search user would then be able to select one or multiple values in the list, depending on the setting of the Selection Type field. The following values can be customized in a List element:

Field	Description
<b>UI Control</b>	A read-only field that is set to <code>Input</code> for the search form criteria.
<b>Control Type</b>	Indicate if the search user will be presented with a list or a drop-down list.

Field	Description
<b>Selection Type</b>	Indicate whether the user is able to select one or multiple values from the list.
<b>Binding Type</b>	<p>When you select a composite binding type, it allows the user to enter search criteria in multiple fields. For example, the user may want to search for Joe in First Name, Last Name or Middle Name fields.</p> <p>One of the following values can be selected:</p> <ul style="list-style-type: none"><li>• <b>Single:</b> The selected list or drop-down list is bound to a single element. Select this value if the search user will search for data in a single field.</li><li>• <b>Composite:</b> The selected list or drop-down list is bound to multiple elements. Select this value if the search user will search for data in multiple fields.</li><li>• <b>Not a Search Criterion:</b> Select if you want to exclude the control value from the search criteria. For example, you only want to use this control value to hide or show a container, then you can choose this option to exclude this control value from the search criteria. This option is only available if the <b>Selection Type</b> is set to <b>Single</b>.</li></ul>

Field	Description
<b>Data Binding</b>	<p>For table searches, enter the name of the variable in the xDB query.</p> <p>For SIP searches, define the value as a criterion. The name value has to match in order to use the criterion for the SIP search.</p> <p>For more information, see <a href="#">Example of Data Binding for a SIP Archive</a>.</p>
<b>Composite Bindings</b>	If <b>Composite</b> is selected as a <b>Binding Type</b> , select binding from the drop-down list. Essentially, you are selecting another field to bind the currently selected field to. Enter a disposition label that will appear on the search form. Refer to <a href="#">Allow a Value in a Search to be Included in Multiple Fields</a> for more information.
<b>Operator Selector</b>	Check if the drop-down form field is to be bound to an operator. For more information, see Values in this table.
<b>Field Label</b>	Enter the name for the field shown on the form.
<b>Required</b>	Indicate whether an input value is required.
<b>Hidden</b>	Indicate whether this field should be hidden from the end user on the search form. This allows the designer to hard code values into a query, which the end user cannot change.
<b>Tooltip Text</b>	Enter concise, helpful information about the field that appears in a small "hover box" when the user hovers the cursor over the control.
<b>Values</b>	<p>Indicate the source of the values in the list:</p> <ul style="list-style-type: none"> <li>• <b>Specified Manually:</b> Enter the values that the search form will display.</li> <li>• <b>Derive from another list selection:</b> <ul style="list-style-type: none"> <li>— Select an existing list to populate the values.</li> <li>— In the <b>Values</b> tab, create the mapping of the values from the existing list to the new list.</li> </ul> </li> <li>• <b>Derive from XQuery execution:</b> <ul style="list-style-type: none"> <li>— Select an option in the <b>Value List</b> field. A value list indicates the data source XML in which the XQuery will run. For more information, see <a href="#">Creating a Value List</a>.</li> <li>— Enter the XQuery data. The XQuery transforms the value lists document you have loaded into items for the drop-down list. The XQuery must produce an XML document with the following structure:</li> </ul> <pre> &lt;data&gt;   &lt;item&gt;     &lt;label&gt;Text label shown in web interface of item 1&lt;/label&gt;     &lt;value&gt;value_of_item_1_in_data&lt;/value&gt;   &lt;/item&gt;   &lt;!-- etc. --&gt; &lt;/data&gt;</pre> </li> </ul>

## Example of Data Binding for a SIP Archive

The following example illustrates how data binding is defined in a query:

```
declare variable $lastName external := ""
```

The following example illustrates the AIC definition under resources for the PhoneCalls application:

```
<criterias>
 <name>CallStartDate</name>
 <label>Call Start Date</label>
 <type>DATETIME</type>
 <pKeyMinAttr>pkeys.dateTime01</pKeyMinAttr>
 <pKeyMaxAttr>pkeys.dateTime02</pKeyMaxAttr>
</criterias>
<criterias>
 <name>CallEndDate</name>
 <label>Call End Date</label>
 <type>DATETIME</type>
</criterias>
<criterias>
 <name>CustomerID</name>
 <label>Customer ID</label>
 <type>STRING</type>
 <pKeyValueAttr>pkeys.values01</pKeyValueAttr>
</criterias>
<criterias>
 <name>CustomerLastName</name>
 <label>Customer Last Name</label>
 <type>HASHED</type>
</criterias>
<criterias>
 <name>CustomerFirstName</name>
 <label>Customer First Name</label>
 <type>STRING</type>
</criterias>
<criterias>
 <name>RepresentativeID</name>
 <label>Representative ID</label>
 <type>HASHED</type>
</criterias>
<criterias>
 <name>CallFromPhoneNumber</name>
 <label>From PhoneNumber</label>
 <type>STRING</type>
</criterias>
<criterias>
 <name>CallToPhoneNumber</name>
 <label>To PhoneNumber</label>
 <type>STRING</type>
</criterias>
```

## Creating a Value List

InfoArchive allows the customer to create a value list, an XML document that has a structure determined by the Developer. It is possible to create multiple value list documents in one application.

Each value list contains:

- A name,
- Description, and
- The XML document that holds the content.

The value list is used to create drop down boxes in search forms with entries based on a document query rather than a fixed list.

Value list documents are not maintained through IA Web App but are loaded from IAShell and its ANT file framework.

In the application directory structure, it is possible to create a directory with the name:

```
value-lists
```

In that value-list's directory, create a new value list by creating two files:

- `valueListName.xml` with the value list document content.
- `valueListName.properties`, a property file with a single `valueListDescription` property. The content could be:

```
valueListDescription=Description of the list
```

The base name (the file name without extension) of the two files should match. It is possible to create multiples of these file pairs, to create multiple value lists in an application.

When a full application that contains a value-lists directory is fully created, the value lists are loaded.

InfoArchive includes multiple examples of value lists:

- Order\_Management/value-lists/container-types
- Baseball/value-lists/birth-years-selection
- Trades/value-lists/trader-names
- Audit/value-lists/event-types

When a value list of an application is changed, load them again by running the following command from the application folder:

```
ant create-value-lists
```

## Using Value Lists

There are different ways to specify what values appear in drop-down boxes. With the value lists functionality, it is possible to:

- Store the values to be shown outside the search form.
- Reuse the same value lists data in multiple forms.
- Specify an XQuery to select a subset of the data in the value lists.

Value lists are XML documents that are stored at the application level. It is possible to have multiple value list documents for one application. For drop-down boxes in search forms, using a value list is one way to select the values to be shown in the drop-down box. When this method is selected, a value list is selected by name, and an XQuery is specified to be run on the value list document to show the values.

In most cases, there will be one value list per drop-down box.

## Deployment

Some new ant targets were introduced to be able to load the value list documents. Within an application folder, it is possible to create a sub-folder named:

```
value-lists
```

Within that folder, create pairs of files with the following setup:

```
nameOfValueList.xml
nameOfValueList.properties
```

These two files belong together and comprise one value list. The .xml document contains the XML content of the value list, and can be any well-formed XML. The properties file only contains one property:

```
valueListDescription=Description of value list
```

If the property file is omitted, the description is left empty.

It is possible to have multiple pairs of these two files in the value-lists folder.

When loading an application with the ant command, value lists present in the value lists directory are automatically loaded. It is also possible to load them explicitly by running the command:

```
ant create-value-lists
```

This can also be used to reload the value lists that were previously loaded. For example, after adding new content to the XML documents, rerun the previous command and the XML content will be replaced in InfoArchive.

Other relevant commands include the following:

Command	Description
ant value-list-names	Shows the names of the value lists of the application stored on IA Server.
ant delete-value-list -Dname =valueListName	Deletes a value list from IA Server.
ant update-value-list -Dname =valueListName	Updates only one specific value list in IA Server.

It is also possible to use the REST interface to create or update value lists.

## Configuration

After a value list has been created within an application, it can be used by a user with the Developer role within drop-down boxes of search forms.

In the form editor, it is possible to specify 'Derive from XQuery execution' in the configuration of a drop-down box, and then specify:

- the value list document to use; and
- the XQuery to be run on that document before presenting the list in the drop-down box.

If the search form is then used, the values for the drop-down list are then determined by executing a query against a form:

## Parts Specification

The screenshot shows a search interface with two main sections:

- Manufacturer & Brand:** This section contains two dropdown menus. The first dropdown is labeled "Manufacturer:" and has the value "Manufacturer#1". The second dropdown is labeled "Brand:" and is currently empty.
- Container Type:** This section contains two dropdown menus. The first dropdown is labeled "Container Size:" and has the value "JUMBO". The second dropdown is labeled "Container Type:" and is open, displaying a list of options: BAG, BOX, CAN, CASE, DRM, JAR, PACK, and PKG. The option "BAG" is highlighted with a blue border.

## XQueries for Value Lists

In the example image above, a sample XQuery is provided. The resulting document must have the following format:

```
<data>
 <item>
 <label>Arkansas</label>
 <value>AK</value>
 </item>
 <item>
 <label>California</label>
 <value>CA</value>
 </item>
 ...
</data>
```

In the example above, a transformation on the document is done. In the most straightforward example, the value list document itself could have the structure given. In that case, the XQuery can simply be:

```
/*
```

This instructs the system to select the entire document.

It is also possible to use an order by clause in the XQuery and this will affect the order in the drop-down box.

## Samples

There is a variety of samples of value lists included in the sample applications:

Sample Application	Value List
Order_Management	container-types
Trades	trader-names
Audit	event-types
Baseball	birth-years-selection US_States

Several of the sample applications also use these value lists in their drop-down boxes. The screen shot above is taken from the "Parts specification query" of the Order\_Management sample application.

## Troubleshooting Value Lists

It is currently not possible to see the value lists within the IA Web App **Administration** interface. You have to use the ant value-list-names command. Using the xDB Admin client, the value lists can be seen, they are in the system database (mainDatabase) in the ValueList library for their metadata, that refers to documents in the library valueListDocuments with the actual XML content.

## Configuring Data Resolution for Supporting Search Value Expansion

Essentially, data resolution configuration allows InfoArchive to expand a single search value according to a configured value map. Then, all expanded values are used as search criteria to execute search. For example, if a user enters multiple values, such as 'a@ot.com, iapm@ot.com', and the value 'iapm@ot.com' can be expanded to 'b@ot.com, c@ot.com', then the entire expanded value 'a@ot.com, b@ot.com, c@ot.com' will be used as search criteria to execute the search.

There are two ways to specify where the data gets resolution:

- [Use XQuery](#) to get data resolution from an internal system. For internal data resolution, value lists are used for specifying a value map, and an XQuery is specified to be run on the value map document to get the resolved data.
- [Use External Service URL](#) to get data resolution from an external system. The external system holds the value map, InfoArchive can communicate with the external system via http request/response to get the resolved data.

## Configuring Data Resolution to Use XQuery

First, a value list must be set up. Refer to [Creating a Value List](#) for more information.

The following is an example of a value map for a stock category map in the Trade sample application. It outlines, for instance, how the Consumers stock category is mapped to 'AAPL' and 'NKE' and the Media stock category is mapped to 'NFLX', etc.

```
<StockCategoryMap>
 <Category>
 <name>Consumers</name>
 <value>AAPL</value>
 <value>NKE</value>
 </Category>
 <Category>
 <name>Media</name>
 <value>NFLX</value>
 </Category>
 <Category>
 <name>Technology</name>
 <value>CSCO</value>
 <value>IBM</value>
 <value>MSFT</value>
 </Category>
 ...
</StockCategoryMap>
```

## Configuring Data Resolution for Form Control

Complete the following steps to configure data resolution for form control:

1. In the form editor's form control properties edit dialog, choose **Use XQuery** for the Data Resolution drop-down.
2. Specify a value list as the XQuery source.
3. Enter the XQuery to be run on specified value list document.

The screenshot shows a user interface for configuring XQuery data resolution. At the top, there are two dropdown menus: 'Data Resolution' set to 'Use XQuery' and 'XQuery Source' set to 'StockCategory'. Below these is a code editor containing the following XQuery script:

```

1 declare variable $Ticker external;
2 let $result :=
3 for $x in /StockCategoryMap/Category[name = $Ticker]/value
4 return <value>{$x/string()}</value>
5 return if (empty($result)) then <value>{$Ticker/string()}</value> else $result

```

For XQuery of data resolution only supports external variable name \$input or the same name as the data binding. In the screen shot above, it is '\$Ticker'.

## Executing Search During Runtime

At run time, if the user inputs a trade category name such as 'Technology' in the input field, which is bound to the Ticker search criterion, and executes search, the search will return results that ticker equates with one of the values 'CSCO', 'IBM' and 'MSFT'.

Because 'Technology' is mapped to 'CSCO', 'IBM' and 'MSFT' values, data resolution expands the 'Technology' to 'CSCO', 'IBM', 'MSFT' and applies the expanded values to the Ticker search criterion to execute the search.

## Configuring Data Resolution to Use an External Service

The customer implements an external system that has value map definition and supports HTTP request/response to get the expanded value for a specific input value.

Using following command to start the sample external system:

```
java -jar StockSectors.jar --server.port=2017
```

The URL <http://localhost:2017> can then be accessed and can get the following response, which are the value map keys:

- Consumers
- Media
- Technology
- ...

If you do a POST <http://localhost:2017> with the request body 'Technology', the response is:

CSCO, EMC, IBM, MSFT

If users want to change the value map in the sample external system, they can define the value map in a properties file and pass the properties file via the `stocksectors-datafile` system property in the start command, such as '`-Dstocksectors-datafile=<properties file path>`'.

```
Consumers=AAPL,NKE
Media=NFLX
Technology=CSCO,EMC,IBM,MSFT
```

```

Auto\ Manufacturer=F
Social\ Media=FB
Industrials=GE
Airlines=JBLU
Semiconductor=INTC,MU
Pharmaceutical=PFE
Oil\ and\ Gas=WLL

```

For example, if users defined the value map properties file at the path 'C:\stocksectors-datafile.properties', then they can start up the external system using following command:

```

java -Dstocksectors-datafile=C:\stocksectors-datafile.properties
-jar StockSectors.jar --server.port=2017

```

Then users can get their customized value map from the sample external system.

## Configuring Data Resolution for Form Control

Complete the following steps to configure data resolution for form control:

1. In the form editor's form control properties edit dialog, choose **Use External Service URL** for the Data Resolution drop-down.
2. Specify the external service URL.
3. Specify the data delimiter that is used for parsing the external service resolved data.



## Executing the Search at Run Time

At run time, ensure the external service has been started. If the user enters 'Technology' in the search field, which is bound to the Ticker search criterion, and executes the search, the search will return results that ticker equates with one of the values 'CSCO', 'IBM' and 'MSFT'.

## Main Search XQuery Adjustment for a Table Application

If the end user submits a single value to execute a search, the single value could be expanded into multiple values by data resolution. For table applications, it needs to adjust the main search XQuery by using following multipleValues function to handle both single and multiple values in where clause:

```

declare function local:multipleValues($values as node()) as xs:string
{
if (empty($values) or $values = "") then ""
else if (count($values/*) eq 0) then concat("", $values , "")
else concat("(", string-join(for $x in $values/value return $x, ", "), ")")
};

```

---

```
let $whereClause := local:addClause("", $birthCity, concat("$elem/BIRTHCITY = ",
local:multipleValues($birthCity)))
```

## Unsupported Form Controls

The following form controls do not support data resolution:

- Date range
- Number range
- Operator selector
- Text
- Group

## Samples

There are a variety of samples of data resolution included in the sample applications:

Sample Application	Sample Search	Optional Command for Using Customized Value Map File in Sample External Service
Baseball	Search By Birth State	-Dstatecity-datafile =<value map file path>
Trade	Trade Category Search	-Dstocksectors-datafile =<value map file path>

## XQuery Modules

Refer to [Appendix B – XQuery Best Practices](#) for an in-depth review of how to use XQuery in conjunction with InfoArchive.

XQuery modules allow the Developer to build utility functions once and reuse them in multiple searches. This avoids redundant code and enables the writing of modular XQuery that can be maintained efficiently.

The XQuery modules can be managed via an ANT task. The sample xquery modules are provided in the bundled Tickets application, which is located in the <INFOARCHIVE\_ROOT>\examples\legacy-ant-applications\Tickets\xquery-modules folder. The table below illustrates different ANT task details for typical CRUD operations for XQuery modules.

There is no way to view, create, delete or update XQuery modules in IA Web App. For this reason, a number of ANT tasks were introduced.

The following ANT tasks can be run from <INFOARCHIVE\_ROOT>/examples/legacy-ant-applications/<application>directory</application>.

ANT Target	Description	Option
xquery-module-names	List the names of the XQuery modules.  To see the content of a specific XQuery module, Ant target <code>export-xquery-module</code> should be used with option name.	
create-xquery-modules	Creates all XQuery modules, located under the <application>/xquery-modules directory.	
create-xquery-module	Creates an XQuery module from a file.	option = location – Required. The location of the file.
update-xquery-module	Updates an XQuery module from a file.	option = location – Required. The location of the file.
delete-xquery-module	Deletes an XQuery module.	option = name – Required. The name of the XQuery module  option = context – Optional. Context can be tenant or application.
export-xquery-module	Exports the XQuery module object, including all properties to an XML file.	option = name – Required. The name of the XQuery module.  option = context – Optional. Context can be tenant or application.  option = location – Required. The location of the output file.

As XQuery modules can be stored on both tenant and application level, an XQuery module with the same name can exist on both tenant and application level. For this reason, ANT targets 'export-xquery-module' and 'delete-xquery-module' have the following strategy when no context option is set:

- If an XQuery module is found on the application level, the operation is applied to this module. If no XQuery module is found on application level, the operation is applied on the XQuery module on tenant level, if found.

Execute the create-xquery-modules ANT task to create all xquery-modules in the xquery-modules folder. For example, to create all XQuery modules in the <INFOARCHIVE\_ROOT>\examples\legacy-ant-applications\Tickets\xquery-modules folder:

```
ant create-xquery-modules -Dlocation=xquery-modules
```

Use the following high-level steps to upload the XQuery module:

1. Execute the create-xquery-module ANT task to create one single xquery-module. For example:

```
ant create-xquery-module -Dlocation=xquery-modules\ iau-utility-tenant.xml
```

2. Execute the update-xquery-module ANT task to update one single existing xquery-module. For example:

```
ant update-xquery-module -Dlocation=xquery-modules\ iau-utility-tenant.xml
```

3. Refer the XQuery module in the XQuery. To do this, a module import declaration is required at the top of the XQuery:

```
import module namespace iau = 'urn:x-emc:ia:util:fn';
```

In the XQuery, functions of the module can be used by using the module prefix:

```
let $whereClause := iau:addClause(...)
```

To see which XQuery modules are loaded into InfoArchive, obtain the names of the loaded XQuery modules by using ANT target xquery-module-names.

To inspect the content of one single loaded XQuery module, use ANT target export-xquery-module.

## Composing the Result List

There is a logical order to design a search but you are free to compose in any order.

After composing the search form, the Developer can create a result list by adding and configuring columns in the Result List.

### Adding Columns

When composing the result list for a table archive, you can add columns and bind them to XQuery element names.

When composing the result list for a SIP archive, you have the options of:

- Manually adding columns to the search results.
- Selecting columns from a schema to add to the form by clicking **Select from Schema**.

### Customizing the Column Type

When you configure a column in the **Result List** tab, the column can be configured to:

- Serve as a link that navigates to a related set of records that is typically retrieved through another search already configured.
- Allow the search user to download information, such as a .pdf file.
- Allow the search user to view information in either the user's native browser or the OpenText Brava! viewer.

## Configuring a Column of Search Results

A normal column is when:

- The column will not be configured with nested search results; and
- The user will not be able to download or view information in a browser or the OpenText Brava! viewer

To configure a normal column:



1. Click to configure the desired column.
2. Enter the following information:

Field	Description
<b>Column Label</b>	Enter the label of the column, which will appear as the header in the result details.
<b>Column Type</b>	For a table archive, select <b>XQuery Reference</b> .  For a SIP archive, select <b>Schema Column Name</b> .
<b>Column Name</b>	For a table search, enter a name for the column defined in the row. Select the xDB Binding.  For a SIP search, the column name is the name of the field. This is the name (case sensitive) that is specified in the criterias section in XML and is also defined in the 220-query.xml file:  <code>&lt;operands&gt; &lt;name&gt;CallStartDate&lt;/name&gt; ... &lt;/operands&gt;</code>
<b>Enable Filter</b>	a. 1. Select if you want to allow the user to filter the column's results.  b. 2. Select the desired data type operators.  If the filter is enabled for at least one of the main fields, during search run time, a drop-down menu is enabled for each field that has the filter.
<b>Sensitive Information</b>	Click to ensure that, when the user executes a background search, that the results are encrypted.  For a SIP search, if the system can detect that the field is encrypted, this field is set and cannot be changed.  For a table search, if the xDB binding is set to a field that is known to be encrypted, the field cannot be changed.
<b>Masked</b>	For a table archive, indicate if the column is masked in the XQuery.
<b>Include in Export</b>	If selected, allows the user of the search to export the search results of the selected column.

Field	Description
<b>Sort Order</b>	Indicate whether: <ul style="list-style-type: none"> <li>• Sort will be disabled for the column.</li> <li>• Sort will be enabled for the column.</li> <li>• The column is to be displayed as a default sort.</li> </ul> The sort order column is hidden if the sensitive information is set.
<b>Data Type</b>	Select the data type for the column.
<b>Hide Column</b>	Indicate if you want the result column to be hidden from the user.
<b>xDB Binding</b>	Binds a search to a specific table. The list of values for the xDB binding also indicates if the field is encrypted.

3. Click OK.

## Adding Filters to a Column of Search Results

The search developer can allow a filter to be used on a single column or multiple columns in the results of a search.

Once a filter is enabled for a column, during search run time, a drop-down menu is enabled for the column. The search user is able to filter the column's results based on the filter operators the developer configures during search composition.

To add a filter to a column of search results:

1. On the **RESULT LIST** tab, click  for the column that you are adding a filter to.
2. Click the **Enable Filter** box.  
The filter operators that are displayed depend on the type of data that will appear in the column. These values may include:
  - Number Operators: For example, to filter a column containing Customer ID numbers, the user can select one of the following filter operators:
    - Equals To
    - Less Than
    - Less Than or Equal To
    - Greater Than
    - Greater Than or Equal To
  - String Operators: For example, to filter a column containing first names, the user can select one of the following filter operators:
    - Begins with
    - Begins with (case sensitive)
    - Contains
    - Ends with

- Exact Match
- Not Equal To
- Date Operators: For example, to filter a column containing the dates employees joined a company, the user can select one of the following filter operators:
  - After
  - Before
  - Between
  - Not On
  - On
  - On or After
  - On or Before

**Note:** If the field is encrypted, the filter operators will be updated to reflect the encryption.

3. Select the filter operators that a search user will be able to apply to the result column.
4. Click **OK**.
5. Repeat the above steps for any other columns the are to include a filter.
6. Because you have updated the search set, it is now considered a draft and be inaccessible to search users. Reset the search **Status** to **Ready**.
7. Click **SAVE**.

## Adding an External Link to Search Results

The search developer is able to add an external link to:

- A column of search results, or
- A side or inline panel that contains search results.

Once configured, the search user will be presented with a link to an external host in the search results, whether within a particular column or a side/inline panel.

The following procedures assume that you are adding an external link to the results of an existing search set. If you are adding a link to a new search set, complete the fields on the **Properties** tab prior to completing the **Linked Column** tab.

To add an external link to a column of search results:

1. On the **RESULT LIST** tab of the search set being updated, click  for the column that you are adding the external link to.
2. On the **Linked Column** tab, for the **Link column as** field, select **External URL**.
3. Complete the following fields.

Field	Description
<b>URL Protocol</b>	<p>Select whether the URL protocol identifier will be <b>Static</b> or <b>Dynamic</b>:</p> <ul style="list-style-type: none"> <li>• If <b>Static</b> is selected, select either <b>http</b> or <b>https</b>.</li> <li>• If <b>Dynamic</b> is selected: <ul style="list-style-type: none"> <li>— For a table archive, enter either HTTP or HTTPS in the field.</li> <li>— For a SIP archive, select the column from the list that contains the applicable URL protocol identifier.</li> </ul> </li> </ul>
<b>Host*</b>	<p>Select whether the host will be <b>Static</b> or <b>Dynamic</b>:</p> <ul style="list-style-type: none"> <li>• If <b>Static</b> is selected, enter the name of the host in the field provided.</li> <li>If <b>Dynamic</b> is selected: <ul style="list-style-type: none"> <li>— For a table archive, enter the name of the column that contains the applicable host.</li> <li>— For a SIP archive, select the column that contains the applicable host.</li> </ul> </li> </ul>
<b>Port</b>	<p>Select one of the following to indicate the port number to which to connect:</p> <ul style="list-style-type: none"> <li>• <b>None</b>: There is no port number.</li> <li>• <b>Static</b>: Enter the port number to which to connect.</li> <li>• <b>Dynamic</b>: <ul style="list-style-type: none"> <li>— For a table archive, enter the name of the column that contains the applicable port number.</li> <li>— For a SIP archive, select the column that contains the applicable port number.</li> </ul> </li> </ul>

Field	Description
<b>Path</b>	Select one of the following to indicate the path: <ul style="list-style-type: none"> <li>• <b>None:</b> There is no path.</li> <li>• <b>Static:</b> Enter the path to which to connect.</li> <li>• <b>Dynamic:</b> <ul style="list-style-type: none"> <li>— For a table archive, enter the name of the column that contains the applicable path.</li> <li>— For a SIP archive, select the column that contains the applicable path.</li> </ul> </li> </ul>
<b>URL Target</b>	Indicate if you want the target to open in a <b>New Browser</b> tab or a <b>New Window</b> .

4. In the External Parameters Mapping section:

- Click 

- Enter the **External URL Parameter Name**.

The external URL has a following structure: [protocol]://[host]:[port]/[path]?[query string]. Provide query string, which is a repeating element consisting of zero or more query parameters. Each parameter needs to have name and value. In this case, "External URL Parameter Name" is a query parameter name, and "Element Name" is a value for that query parameter.

- For the **Element Name** field, provide values for the query parameters:
  - For a table archive, manually enter the column name.
  - For a SIP archive, select the name of the column.

5. Click **OK**.

## Creating a Nested Search

To configure a column for a nested search:

- Click  to configure the desired column.
- Enter the following information:

Field	Description
<b>Column Label</b>	Enter the label of the column, which will appear as the header in the result details.
<b>Column Type</b>	Select <b>Linked Column (Nested Search)</b> .
<b>Column Name</b>	<p>Enter one of the following:</p> <ul style="list-style-type: none"> <li>• For a table search, enter a name for the column defined in the row.</li> <li>• For a SIP search, select the column name from the list. The column name is the name of the field. This is the name (case sensitive) that is specified in the criterias section in XML and is also defined in the <code>220-query.xml</code> file:</li> </ul> <pre>&lt;operands&gt; &lt;name&gt;CallStartDate&lt;/name&gt; ... &lt;/operands&gt;</pre>

In the Nested Search Mapping section:



- Click . You can map as many result columns to search fields, as required.
- For a table archive, enter a Result Column Binding Name and a Search Field Binding Name.  
For a SIP archive, select a Result Column Binding Name and a Search Field Binding Name.
- Click OK.

Alternately, you can also:



- Click to configure the desired column.
- Access the **Linked Column** tab.
- In the **Linked column** as radio group, select **Nested Search**.
- Select a column name from the value from the **Nested Search** list. See the information for Column Names in the table above.

In the Nested Search Mapping section:



- Click . You can map as many result columns to search fields, as required.
- For a table archive, enter a Result Column Binding Name and a Search Field Binding Name.  
For a SIP archive, select a Result Column Binding Name and a Search Field Binding Name.
- Click OK.

## Configuring a Result Column to Allow Downloadable Content

You can configure a column so that it allows the user to download content in runtime.



- Click to configure the desired column.
- Enter the following information:

Field	Description
<b>Column Label</b>	Enter the label of the column, which will appear as the header in the result details.
<b>Column Type</b>	Select <b>Downloadable Content</b> .
<b>Column Name</b>	<p>Enter one of the following:</p> <ul style="list-style-type: none"> <li>For a table search, enter a name for the column defined in the row.</li> <li>For a SIP search, select the column name from the list. The column name is the name of the field. This is the name (case sensitive) that is specified in the criterias section in XML and is also defined in the 220-query.xml file:</li> </ul> <pre>&lt;operands&gt; &lt;name&gt;CallStartDate&lt;/name&gt; ... &lt;/operands&gt;</pre>
<b>Content Link Display</b>	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li><b>Custom Label:</b> Enter a customized label for each table cell for the column that contains the downloadable content. The label that you enter will be displayed to the search user in run time.</li> <li><b>From a column:</b> The value entered depends on the type of application: <ul style="list-style-type: none"> <li>For a SIP application, select the column</li> <li>For a table application, enter a column</li> </ul> <p>For example, if a Customer ID is picked when the Content link display is <b>From a column</b>, during run time, the user sees the value that is returned for customer ID such 000453.</p></li> <li><b>Download Icon:</b> Select to display the download icon in the search results during runtime.</li> </ul>
<b>Enable Filter</b>	<ol style="list-style-type: none"> <li>1. Select if you want to allow the user to filter the column's results.</li> <li>2. Select the desired data type operators.</li> </ol> <p>If the filter is enabled for at least one of the main fields, during search run time, a drop-down menu is enabled for each field that has the filter.</p>
<b>Include in Export</b>	If selected. allows the user of the search to export the search results of the selected column.

3. Click **OK**.

## View

You can configure a column so that it allows the user to view content in runtime.

1. Click  to configure the desired column.

2. Enter the following information:

Field	Description
Column Label	Enter the label of the column, which will appear as the header in the result details.
Column Type	Select <b>Downloadable Content</b> .
Include in Export	If selected, allows the user of the search to export the search results of the selected column.
Column Name	<p>Enter one of the following:</p> <ul style="list-style-type: none"><li>• For a table search, enter a name for the column defined in the row.</li><li>• For a SIP search, select the column name from the list. The column name is the name of the field. This is the name (case sensitive) that is specified in the criterias section in XML and is also defined in the 220-query.xml file:</li></ul> <pre>&lt;operands&gt; &lt;name&gt;CallStartDate&lt;/name&gt; ... &lt;/operands&gt;</pre>

Field	Description
Label>Show Icon	<p><b>Label:</b> Enter a customized label for each table cell for the View Content column.</p> <p><b>Show Icon:</b> Select this box if you would rather have an icon displayed as the View Content header.</p>
Preview	By default, this is checked, which means that the user will be allowed to view a preview of the item. Deselect this box to disable the popup preview.
Viewer Type	Indicate the browser that will be used to view the information, either <a href="#">Browser Native</a> or the OpenText Brava! viewer.
Mime Type	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>a. <b>Static:</b> Indicate the format of the content being viewed.           <ul style="list-style-type: none"> <li>• If <b>Viewer Type</b> is set to <b>Browser Native</b>, enter the MIME type.</li> <li>• If <b>Viewer Type</b> is set to OpenText Brava! viewer, select the MIME type from the drop-down list.</li> </ul> </li> <li>b. <b>From a column:</b> Assign the MIME type from the value in the result set.           <ul style="list-style-type: none"> <li>• For SIP archiving, select the column name from the drop-down list.</li> <li>• For table archiving, enter the column name from the return section of the XQuery.</li> </ul> </li> <li>c. Alternatively, if the content type is not consistent over the entire data, the mime type can be specified as a value of another column. The same can be configured by selecting "From a column" and selecting/typing in the same of the column. The mime type of each content would then be picked by from the corresponding entry in the selected column.</li> </ul>
Print/Download	Select if you want the user to be able to print or download the item being viewed.

3. Click **OK**.

## OpenText Brava! Viewer

The OpenText Brava! Viewer supports the following content types: pdf, bmp, tiff, png, jpeg, and gif. Ensure that the appropriate Mime Type is selected for the content.

The following tables outline whether or not the OpenText Brava! viewer supports specific file types for the different browsers.

### Microsoft Edge 38+

Format	Browser Native	Brava! Viewer	Preview
PDF	Yes	Yes	Yes
TIFF	Yes	Yes	Yes
JPEG	Yes	Yes	Yes
GIF	Yes	Yes	Yes
PDF-A	Yes	Yes	Yes
BMP	Yes	Yes	Yes
PNG	Yes	Yes	Yes

**Internet Explorer Version 11+**

Format	Browser Native	Brava! Viewer	Preview
PDF	Yes	Yes	Yes
TIFF	Yes	Yes	Yes
JPEG	Yes	Yes	Yes
GIF	Yes	Yes	Yes
PDF-A	Yes	Yes	Yes
BMP	Yes	Yes	Yes
PNG	Yes	Yes	Yes

**Firefox 57+**

Format	Browser Native	Brava! Viewer	Preview
PDF	Yes	Yes	Yes
TIFF	No	Yes	No
JPEG	Yes	Yes	Yes
GIF	Yes	Yes	Yes
PDF-A	Yes	Yes	Yes
BMP	Yes	Yes	Yes
PNG	Yes	Yes	Yes

**Chrome Version 63+**

Format	Browser Native	Brava! Viewer	Preview
PDF	Yes	Yes	Yes
TIFF	No	Yes	No
JPEG	Yes	Yes	Yes
GIF	Yes	Yes	Yes
PDF-A	Yes	Yes	Yes

Format	Browser Native	Brava! Viewer	Preview
BMP	Yes	Yes	Yes
PNG	Yes	Yes	Yes

## Native Browser

The Native Browser supports a broader set of content types. Since each browser has a different set of formats it supports out-of-the-box, check with the browser documentation for the formats it supports. The native viewer is launched by the browser depending on the Mime Type provided. If the content plug-in for that mime type is installed, it would be launched and content rendered. Most modern browsers allow additional plug-ins to be installed for rendering different types of content. Refer to the documentation of your browser to see how to install these plug-ins.

Our testing shows the following formats are supported by the listed browsers:

Browser	PDF	PDF/A	TIF	JPEG	GIF	BMP	PNG
Firefox	No	No	No	Yes	Yes	Yes	Yes
Internet Explorer 11	Yes	No	No	No	No	No	No
Chrome	Yes	No	No	Yes	Yes	Yes	Yes
Edge	Yes	No	No	No	No	No	No

## Configuring Exports

The Developer is able to allow search results to be exported in a specific format. This action can be performed when a search is being created or edited. Export functionality is displayed on the main search result grid or the on side or in-line panels at the tab level. Export functionality can be enabled or disabled per tab.

At runtime, users are given a choice of exporting search results according to the options selected during composition time. When exporting at the tab level in the detail section, one row of data (selected row) is available for the export operation. From the main search result screen, the user is required to select an export option from the Export menu, which contains options selected by the Developer during search composition.

The download option also depends on the browser settings. For instance, in some browsers, a save as menu may be presented or the download may start when the download button in the background items listing page is selected. The gzip option downloads with a .gz extension. In case of included content or multiple files in the archive for other reasons, the content of the .gz file will be a tar file and the full file extension will be .tar.gz. It is also possible to download as .tar or .zip files instead, assuming the corresponding export pipelines are enabled.

To add the ability to export search results:



1. Click the button and select Export. The **Add Action** is available in the **Result List** tab for the main result grid, as well as the side and in-line panels of the **Result Detail** tab.

By default, it is enabled on the **Result List** tab. On the **Result** tab, however, it needs to be enabled.

2. Click the button to further configure the export process.
3. Select the pipeline to be permitted for the export process. A pipeline represents a series of processing steps for the selected search result. The following options are available:  
InfoArchive provides a number of out-of-box ‘tenant’ level pipelines. Furthermore, some sample applications may include custom application-specific pipelines. Customers can also define their own custom pipelines.
4. Click **OK**.
5. Click **Save**.

### Adding the Export Action to the Result List

In the main grid, when user clicks on the **Result List** tab, the available actions toolbar will appear and a default configuration (for example, .csv, .gzip format with no content) is selected. The Developer may choose to change this setting. The Developer can disable the export feature for the main result listing page or individual tabs. By default, it is off for tabs in the detail section.

**Date\_Operator** Draft

Set 1

SEARCH FORM	<b>RESULT LIST</b>	RESULT DETAIL	PERMISSIONS
-------------	--------------------	---------------	-------------

**+ Add Column**   **+ Add Action ▾**   **Select from Schema**

*Available Actions*

Click next to EXPORT to launch the Export configuration selection dialog (the Developer can disable the export functionality for the main grid and/or individual tab using the top switch). The tenant level is available to all applications:

**EDIT EXPORT**

**Pipelines**

Select the pipeline to be allowed as custom export methods.

Pipeline Scope: All

Name	Application Scope	File Format	Export Content	Compression	Export Location
PhoneCalls-search-results-pdf-with-content-gzip	✓ ⓘ	PDF	✓	GZIP	Default
PhoneCalls-search-results-xsl-with-content-gzip	✓ ⓘ	CSV	✓	GZIP	Default
gzip_csv	ⓘ	CSV		GZ	Default
gzip_csv_with_content	ⓘ	CSV	✓	TGZ	Default
tar_csv_with_content	ⓘ	CSV	✓	TAR	Default
zip_csv_with_content	ⓘ	CSV	✓	ZIP	Default

**Asynchronous Export**

An export file results in a Background Export order item, while keeping the application responsive during the export.

CANCEL OK

When EXPORT is disabled:

**EDIT EXPORT**

**Pipelines**

User cannot export any data. The Export button will not be displayed.

CANCEL OK

The add export action is also available at the tab level in the Result Detail in the in-line and side panels. By default it is enabled in the main grid. In other places, however, it has to be enabled.

## Including Repeating Elements in Exported Data

By default, repeating attributes are not marked to be included in an exported file. The search Developer can identify the columns of interests that are to be included and viewed in the exported output.

The screenshot shows a search results page for a customer named Neil Mcneil. The top section displays basic customer information: Customer ID (10000), First Name (Neil), Last Name (Mcneil), Email (Neil.Mcneil EMAIL), Credit Card N... (partial), Member Since (2014-09-03T21:21:43-07:00), and a Download link. Below this, a 'Transactions' section is expanded, showing two rows of transaction details: Amazon (Description, Date 2014-09-03T21:21:43-07:00, Amount 300) and BMW Motors (Description, Date 2015-09-03T21:21:43-07:00, Amount 30000). Each transaction row has a 'View' icon.

The exported CSV shows a column per repeating attribute table (Group). In the above screen shot, it is named Transactions. The first row is the exported properties followed by two rows of repeating values:

```
Transactions
"Description", "DateOfTransaction", "AmountOfTransaction"
"Amazon", "2014-09-03T21:21:43-07:00", "300"
"BMW Motors", "2015-09-03T21:21:43-07:00", "30000"
```

<code>id</code>	<code>Email</code>	<code>FirstName</code>	<code>Transaction1</code>	<code>MemberSince</code>	<code>LastName</code>	<code>Transactions</code>
f8f1cca2-31fd-4024-9ee7-12285b7d642	Neil.Mcneil EMAIL	Neil	f8f1cca2-31fd-4024-9ee7-	2014-09-03T21:21:43-07:00	Mcneil	"Description", "DateOfTransaction", "AmountOfTransaction" "Amazon", "2014-09-03T21:21:43-07:00", "300" "BMW Motors", "2015-09-03T21:21:43-07:00", "30000"

## Enabling Collection Functionality

Before the collections functionality can be used, the search designer has to allow the E-Discovery Administrator to access the functionality.

1. When designing a search form, on the Result List tab, click **Add Action > Create Collection**. This allows the E-Discovery Administrator to access the collections functionality.
2. Click **Save**.

## Custom View

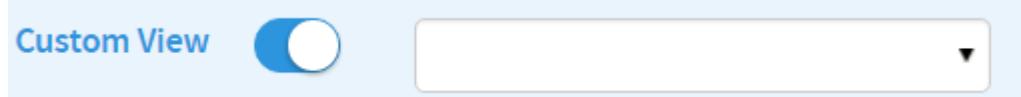
The Developer is able to apply a custom layout to a search result set. The custom layout has to be created and loaded to the system before it can be used. Refer to [Custom Presentation](#) for further information.

To apply a customized layout to a result set:

1. Click and **Edit** for the search being refined.

2. On the **Result List** tab, select **Custom View**.

Once selected, a list is displayed that allows you to select the desired custom view to apply to the search results:



3. Select the desired layout.

When the search is now executed, the set of search results will be displayed according to the selected layout.

## Composing Result Details

The **Result Detail** tab allows you to enter information that will appear in either a side or in-line panel of the search results. Side and in-line panels contain information that is displayed when the user selects a specific row in the search results. In-line panels appear when the user click the triangle next to a specific row. In the **Result Detail** tab, you are able to add tabs. Repeating fields can only appear in the in-line panel.

For a table archive, you add fields to the selected result detail panel with the intention of binding them to XQuery element names.

### Adding Details to a Side Panel

The following shows the **Result Detail tab > Side Panel** selected without any fields or tabs added:

A screenshot of a web interface for managing search results. At the top, there's a header with "Name Search Draft" and icons for "DISCARD CHANGES" and "SAVE". Below the header, a section titled "Set 1" is shown. Under "Set 1", there are four tabs: "SEARCH FORM", "RESULT LIST", "RESULT DETAIL" (which is highlighted in blue), and "PERMISSIONS". Below these tabs, there are two buttons: "Side Panel" (which is highlighted in blue) and "Inline Panel". A note below the tabs says: "The following fields will appear in detail panel when a row in result is selected. For large number of fields, you can organize them into tabs." At the bottom of the screenshot, there's a message: "Begin adding fields to compose your result detail". At the very bottom of the interface, there are three buttons: "+ Add Field", "+ Add Action ▾", and "Select from Schema".

To add a field, click + **Add Field**.

When working with a SIP archive, you are able to add fields by selecting them from the schema:

1. In the **Result Detail** tab for the SIP search, click **Select from Schema**.
2. Select the fields you want to include in the panel and click **OK**.

To remove a field, click the X in the row for the field being removed (indicated by the red arrow in the following screen shot):

The screenshot shows the 'Name Search' interface with the 'RESULT DETAIL' tab selected. Below it, an 'Inline Panel' is open, listing four fields: 'Customer ID', 'Last name', 'First name', and 'ID'. Each field has a small gear icon and an 'X' icon to its right. A red arrow points to the 'X' icon next to the 'First name' field.

## Adding an In-line Panel

The following shows the “**Result Detail tab > Inline Panel**” selected without any fields or tabs added:

To add a field, click **+ Add Field**.

When working with a SIP archive, you are able to add fields by selecting them from the schema:

1. In the **Result Detail** tab for the SIP search, click **Select from Schema**.
2. Select the fields you want to include in the panel and click **OK**.

## Adding Tabs

If you plan on adding a lot of fields to a side or in-line panel, consider organizing them in tabs.

To add a tab, click **+ Add Tab**. The following screen shot illustrates how the tab appears once two tabs have been added. The currently selected tab is underlined:

The screenshot shows the 'Name Search' interface with the 'RESULT DETAIL' tab selected. Below it, an 'Inline Panel' is open, showing two tabs: 'Tab 0' and 'Tab 1'. 'Tab 1' is currently selected and highlighted with a red box. Below the tabs, there is a message: 'The following fields will appear in detail panel when a row in result is expanded. For large number of fields, you can organize them into tabs.' At the bottom of the panel are buttons for '+ Add Field', '+ Add Action', and 'Select from Schema'.

To remove a tab, click the X beside the desired tab.

To change the name of a tab:

1. Click  for the desired tab.
2. Enter a **Tab Name** and click **SAVE**.

### **Customizing Panel Fields for a Table Archive**

After adding fields to a side or inline panel, you are able to further customize them.

1. Click  for the desired field.
2. Complete the required fields. The fields that are displayed will depend on the value you set for the Field Type field:

Field	Description
<b>Detail Field Label</b>	The name entered here will appear in the panel when a user executes a search and views the results in runtime.
<b>Field Type</b>	<p>Indicates the type of field that will be presented in the panel.</p> <p>If you select <b>XQuery Reference</b>, complete the following fields:</p> <ul style="list-style-type: none"> <li>• <b>Detail Field Name:</b> Enter a name that will appear in the panel's search results.</li> <li>• <b>Sensitive Information:</b> Click to ensure that, when the user executes a background search, that the results are encrypted.</li> <li>• <b>Include in Export:</b> If selected, allows the user of the search to export the search results of the selected column.</li> <li>• <b>Hide Column:</b> Indicate whether you want the result column to be hidden from the user.</li> </ul> <p>If you select <b>Downloadable Content</b>, complete the following fields:</p> <ul style="list-style-type: none"> <li>• <b>Detail Field Name:</b> Enter a name that will appear in the panel's search results.</li> <li>• <b>Content Link Display:</b> If you select: <ul style="list-style-type: none"> <li>— <b>Custom Label</b>, enter the label text that will appear in the panel's search results.</li> <li>— <b>From a column</b>, enter the name of the column that will appear in panel's search results.</li> <li>— <b>Download Icon</b>, the default download icon will appear in the panel's search results.</li> </ul> </li> <li>• <b>Include in Export:</b> If selected, allows the user to export the search results of the selected column.</li> </ul> <p>If you select <b>View Content</b>, complete the following fields:</p> <ul style="list-style-type: none"> <li>• <b>Detail Field Name:</b> Enter a name that will appear in the panel's search results.</li> <li>• <b>Content Link Display:</b> If you select, <ul style="list-style-type: none"> <li>— <b>Custom Label</b>, enter the label text that will appear in the panel's search results.</li> <li>— <b>From a column</b>, enter the name of the column that will appear in panel's search results.</li> <li>— <b>View Icon</b>, the default view icon will appear in the panel's search results.</li> </ul> </li> <li>• <b>Viewer Type:</b> Select the browser in which the user will view the content. If this field is set to: <ul style="list-style-type: none"> <li>— <b>Browser Native Viewer:</b> The native browser supports a broader set of content types. Since each browser has a different set of formats it supports out-of-the-box, check<sup>333</sup> the particular format with the corresponding viewer. The native viewer is launched by the browser depending on</li> </ul> </li> </ul>

3. Click **OK**.
4. Click **SAVE**.

## **Customizing Panel Fields for a SIP Archive**

After adding fields to a side or inline panel, you are able to further customize them.

1. Click  for the desired field.
2. Complete the required fields. The fields that are displayed will depend on the value you set for the Field Type field:

Field	Description
<b>Detail Field Label</b>	The name entered here will appear in the panel when a user executes a search and views the results in runtime.
<b>Field Type</b>	<p>Indicates the type of field that will be presented in the panel.</p> <p>If you select <b>Schema Column Name</b>, complete the following fields:</p> <ul style="list-style-type: none"> <li>• <b>Detail Field Name:</b> Select the name that will appear in the panel's search results.</li> <li>• <b>Sensitive Information:</b> Click to ensure that, when the user executes a background search, that the results are encrypted.</li> <li>• <b>Include in Export:</b> If selected, allows the user of the search to export the search results of the selected column.</li> <li>• <b>Hide Column:</b> Indicate whether you want the result column to be hidden from the user.</li> </ul> <p>If you select <b>Downloadable Content</b>, complete the following fields:</p> <ul style="list-style-type: none"> <li>• <b>Detail Field Name:</b> Select the name that will appear in the panel's search results.</li> <li>• <b>Content Link Display:</b> If you select: <ul style="list-style-type: none"> <li>— <b>Custom Label</b>, enter the label text that will appear in the panel's search results.</li> <li>— <b>From a column</b>, select the name of the column that will appear in panel's search results.</li> <li>— <b>Download Icon</b>, the default download icon will appear in the panel's search results.</li> </ul> </li> <li>• <b>Include in Export:</b> If selected, allows the user to export the search results of the selected column.</li> </ul> <p>If you select <b>View Content</b>, complete the following fields:</p> <ul style="list-style-type: none"> <li>• <b>Detail Field Name:</b> Select the name that will appear in the panel's search results.</li> <li>• <b>Content Link Display:</b> If you select, <ul style="list-style-type: none"> <li>— <b>Custom Label</b>, enter the label text that will appear in the panel's search results.</li> <li>— <b>From a column</b>, select the name of the column that will appear in panel's search results.</li> <li>— <b>View Icon</b>, the default view icon will appear in the panel's search results.</li> </ul> </li> <li>• <b>Viewer Type:</b> Select the browser in which the user will view the content. If this field is set to: <ul style="list-style-type: none"> <li>— <b>Browser Native Viewer:</b> The native browser supports a broader set of content types. Since each browser has a different set of formats it supports out-of-the-box, check the particular format with the corresponding viewer. The</li> </ul> </li> </ul>

3. Click **OK**.
4. Click **SAVE**.

## Reordering Fields

Reorder the rows by dragging and dropping a particular field into a new position in the **Result Detail** tab. The cursor will change to indicate that a row can be moved.

## Managing Permissions

The **Permissions** tab allows you to restrict search sets to specific groups. By default, all search sets are available to any user that can execute a search. Once a search set is restricted to a group, that group cannot be assigned to another set.

The Developer can access all search sets by selecting a set from a drop-down in **Search Detail** tab and running the search.

The **Permission** tab includes a **Find a group** field that enables the the Developer to locate a specific group. The tab also includes a filter to show either **Show all groups** or **Show only groups with access**. If a set is restricted to a group, then it will be excluded and another set can be reassigned to the same restricted group. A person or role belonging to multiple groups, however, will have access to multiple sets.

To restrict access to a group, select one or more groups and click **SAVE**.

## Saving a Search Set

You are advised to continuously save any changes made to a search set.

If you access a search set and make a change (for example, add or remove a field), and then you try to navigate away from the set without saving, the system will prompt you to:

- Discard your changes,
- Save your changes, or
- Cancel the exit out of the search set. This allows you to make further changes to the search set.

## Renaming a Search Set

The Developer is able to change the name of a search set:

1. Click  and select **Set Name**.
2. Enter a new name for the search set and click **SAVE**.

## Deleting a Search

The Developer is able to delete the search even if other users have done background searches. In previous versions, the developer had to wait for the Clean job to remove the background tasks, which was a day after the search completed. Users will not be able to view background search results if the search is deleted.

1. Click the context menu for the search being deleted and select **Delete Search**.
2. When prompted to confirm the deletion, click **DELETE**.

## Deleting a Search Set

The Developer is able to delete a search set:

1. Click  and select **Edit**.
2. For the set being deleted, click  and select **Delete Set**.
3. When prompted to confirm the deletion, click **DELETE**.

## Duplicating a Search Set

The Developer is able to create a search set from an existing search set.

1. Access the desired application.
2. For the desired search in which you want to create the new search set, click  and select **Edit**.  
The search sets are displayed on a panel on the right side.
3. Click  and select **Create Duplicate**.  
The new search set appears in the panel.

## Search Form Composition Tips

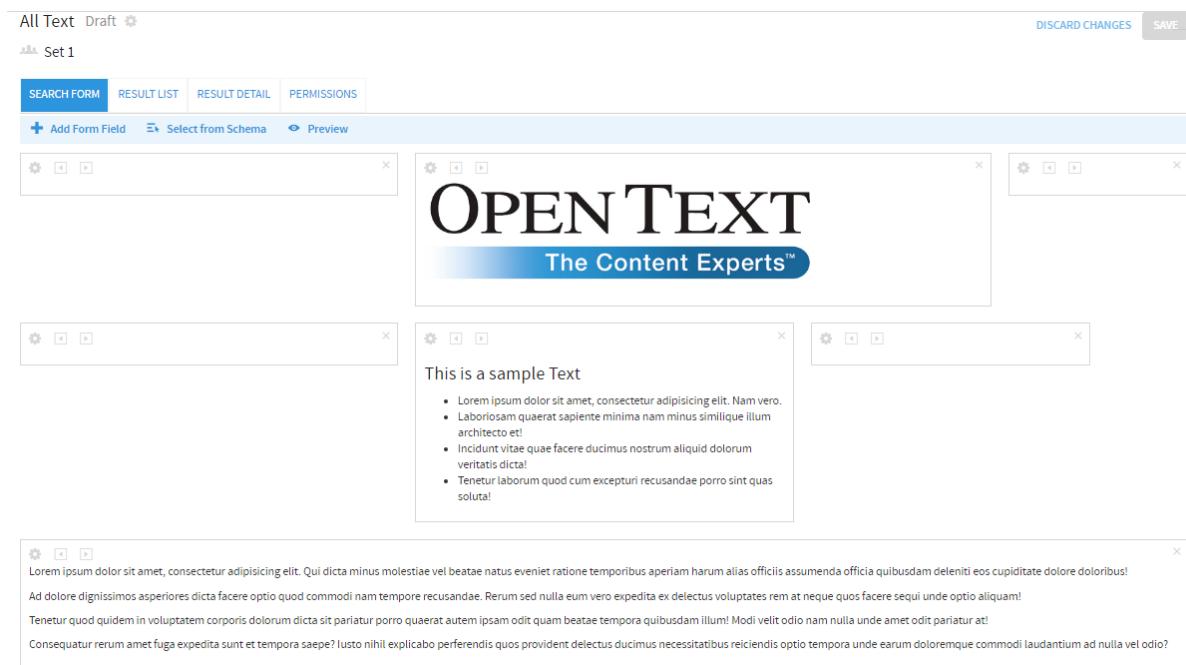
Hidden field elements are used to carry a default value, which will be submitted when search is performed but the hidden element and its value will not be shown on the form. Hidden control values will be submitted when the search is executed.

If using images or logos, the files have to be pre-loaded in the following location:

<INFOARCHIVE\_ROOT>\examples\legacy-ant-tenants\infoarchive\customization\branding\images. The image that appears in the screen shot below was tagged accordingly:

```
<html><body></body></html>
```

## Searches in InfoArchive



Values of the controls in conditional hidden group will not be submitted.

In general, follow these XML element name rules:

- Element names must start with a letter or underscore.
- Element names cannot start with the letters xml (or XML, or Xml, etc.).
- Element names can contain letters, digits, hyphens, underscores and periods.
- Element names cannot contain spaces.

In general, avoid using XForms built-in datatype names:

dateTime	NCName
time	IDIDREF
date	IDREFS
gYearMonth	NMTOKEN
gYear	NMTOKENS
gMonthDay	integer
gDay	nonPositiveInteger
gMonth	negativeInteger
string	long
boolean	int
base64Binary	short
hexBinary	byte
float	nonNegativeInteger
decimal	unsignedLongunsignedInt
double	unsignedShort
anyURI	unsignedByte
QName	positiveInteger
normalizedString	listItem
token	listItems
language	dayTimeDuration
Name	

## Updating a Search Form Status to Ready

When a search template is created, the status remains in Draft mode until it is updated by the Developer. When a search template is in Draft mode, the End User will not be able to access it until the template's status is set to Ready.

1. Select the application in which the search is stored.
2. Click the context menu for the desired search and select Set to Ready.

Furthermore, when a search is edited, the Status of the template returns to Draft. You are, however, able to update the Status of a search template being edited. Changing any data, including the query or columns, resets the status to Draft, which allows the Developer to test the changes prior to setting the status to Ready.

1. The current Status of the search template is displayed beside the name of the template. Once you have finished editing the template, click .
2. On the **Edit Search** page, update the Status field to **Ready**.
3. Click **OK**.
4. Click **Save**.

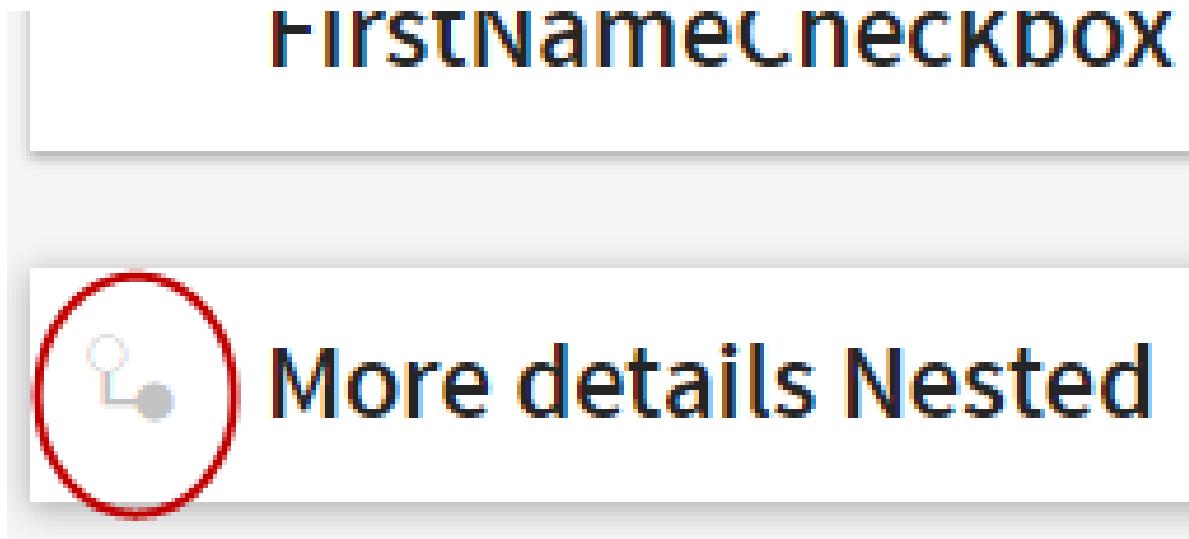
## Nested Searches

When searches are nested, values are passed from primary search to the nested search to run.

For a SIP archive search, the Developer determines if a particular search is a nested search in the search Type drop-down list. For a table archive, however, the search has to be configured.

There is no limit to how many levels appear in a nested search.

The Developer is able to tell if a particular search is a nested search from the following icon:



During runtime, if a value is missing when user clicks in the nested search, it cannot be executed and user will be presented with the search form of the nested search so that the missing value can be entered.

**Note:** Only the Developer will see how the nested search appears in the screen shot above. Other users will only know about a nested search when they access it via the primary search.

## Configuring a Dependent Select Control Powered by XQuery

When creating your own dependent drop-down list, ensure that the following criteria is met:

- For a drop-down list, add the **Select** form field to the search canvas.
- Configure the drop-down list with the following settings:

Field	Description
Control Type	Set to either <b>Drop-down</b> or <b>List</b> .
Selection Type	For both the source and the target select, Selection Type can be Single or Multiple.
Binding Type	Set to <b>Single</b> .

Field	Description
<b>Data Binding</b>	For table searches, enter the name of the variable in the xDB query.  For SIP searches, select the value as a criterion. The name value has to match in order to use the criterion for the SIP search.  For more information, see <a href="#">Example of Data Binding for a SIP Archive</a> .
<b>Field Label</b>	Enter the name for the field shown on the form.
<b>Required</b>	Indicate whether an input value is required.
<b>Hidden</b>	Indicate whether this field should be hidden from the end user on the search form. This allows the designer to hard code values into a query, which the end user cannot change.
<b>Tooltip Text</b>	Enter concise, helpful information about the field that appears in a small “hover box” when the user hovers the cursor over the control.
<b>Values</b>	As a target Select control powered by XQuery, it must use the <b>Derive from another list selection + XQuery</b> option.  A detailed explanation of how to complete this field follows this table.
<b>Default Value</b>	Enter one of the value text entries from the data specified in the Values field. For a single default value, only enter one value. For SELECT multiple selection, values should be separated by SPACE.
<b>Data Resolution</b>	Refer to <a href="#">Configuring Data Resolution for Supporting Search Value Expansion</a> for further information.

When using target select control, you must also configure the source select control. The source select control can be a single drop-down list or a single list. If using either the drop-down or single list, set the Values field to **Specified Manually** or **Derive from XQuery execution**. A radio or checkbox group can also be used as the source select control.

The following illustrates how to complete the **Values** field. All sample searches in the Audit application provide examples of how to configure dependent drop-down lists that are powered by XQuery. There are two fields in the sample searches for the Audit application:

- **Event Type**, which acts as the source select on the search form; and
- **Event Name**, which acts as the target select on the search form.

The fields are used below to further illustrate how to configure the dependent drop-down list powered by XQuery.

In the **Values** field, select one of the following:

- **Specified Manually:** For the source select control, enter the values that the search form will display.
- **Derive from XQuery execution:** For the source select control, select an option in the **Value List** field. A value list indicates the data source XML in which the XQuery will run. For more information, see [Creating a Value List](#).
  - Enter the XQuery data. The XQuery transforms the value lists document you have loaded into items for the drop-down list. The XQuery must produce an XML document with the following structure:

```
<data>
 <item>
 <label>source select item label shown in UI</label>
 <value>source select item value</value>
 </item>
 <!-- etc. -->
</data>
```

In the Audit application, the Application Audit sample search uses this setting for the **Event Type** field.

For the **XQuery Source** field, the search developer selected event-types.

The following is entered in the **XQuery** field:

```
<data> { for $item in /alleventtypes/event_type order by $item/name return
 <item><label>{$item/display_label/text()}</label><value>{$item/name/text()}</value>
 </item> } </data>
```

- **Derive from another list selection + XQuery:** In the Audit application, the Application Audit sample search uses this setting for the **Event Name** field.

For the **List Name** field, the search developer selected **Event Type**: (eventType) for the source select control.

For the **XQuery Source** field, the search developer selected event-types. This is for selecting the XQuery source XML, which is the Value List object stored in IA Server.

The following is entered in the **XQuery** field. The XQuery returned XML structure must be follows:

```
<data>
 <map>
 <key>match the value from source select</key>
 <values>
 <item>
 <label>target select item label shown in UI</label>
 <value>target select item value</value>
 </item>
 <!-- etc. -->
 </values>
 </map>
 <!-- etc. -->
</data>
```

The following is an example:

```
<data> { for $map in /alleventtypes/event_type return <map> <key>{$map/name/text()}
 </key> <values> { for $item in $map/events/event order by $item/name \
 return <item><label>{$item/display_label/text()}</label><value>{$item/name/text()}</value>
 </item> } </values> </map> } </data>
```

# Adding Date-Time Fields During Search Composition

This section illustrates how to add a date-time field to a search form and results:

The information presented in this section assumes that you have stored date-time information in an InfoArchive repository.

The following procedure uses the Trades sample application to illustrate how to add a date-time field to a search form and the search results. While the following procedure demonstrates how to update a SIP archive, the steps to update a table archive are the same:

1. When creating a new search, enter the preliminary information and click **NEXT**. Refer to [Creating a Search Set](#) for further information.
2. When selecting fields to display in the search form, select `TradeDate` and click **NEXT**.
3. When selecting column to include in the search results, select `TradeDate` and click **FINISH**.
4. In the **Search Form**, click  to edit the Trade Date field.
5. Set the **Control Type** field to either:
  - **DateTime**: The search user will have to enter a single date and time.
  - **DateTime Range**: The search user will have to enter a range of dates and select the times for each date.
6. Indicate whether or not you want to use your **Local Timezone** for the field.

If selected, the date-time filed input value will be treated as the local time. For example, if a user in Los Angeles entered the date-time '2010-01-01 12:00:00 AM', this date-time value will be treated as Los Angeles time using the Los Angeles time zone (2010-01-01 12:00:00 AM (-8:00)). If a user in Shanghai enters the time '2010-01-01 12:00:00 AM', then this date-time value will be treated as Shanghai time with Shanghai time zone (2010-01-01 12:00:00 AM (+8:00)).

If the **Local Timezone** box is not selected, the Coordinated Universal Time (UTC) is applied to the field. If the user stored the date-time value without a time zone in the InfoArchive repository, they can un-select this box to search the date-time values that do not include a time zone in the repository.

7. Complete the following fields, as desired:
  - **Binding Type**
  - **Data Binding**
  - **Field Label**
  - **Required**
  - **Hidden**
  - **Tooltip Text**

8. The next step depends on the value you selected for the **Control Type** field:
  - If **DateTime** was selected, for the **Default Value** fields:
    1. Enter a default date and then manually enter a default time or use the timepicker to enter the time.
    2. If desired, configure the **Data Resolution**. Refer to [Configuring Data Resolution for Supporting Search Value Expansion](#) for further information.
    3. Click **OK**.
  - If **DateTime Range** was selected:
    4. For the **Default From Value** field, enter a default date and then manually enter a default time or use the timepicker to enter the time.
    5. For the **Default To Value** field, enter a default date and then manually enter a default time or use the timepicker to enter the time.
    6. Click **OK**.
9. If desired, in the **Result List** tab, click to edit the TradeDate column.  
Update the fields, as desired and click **OK**. Refer to [Configuring a Column of Search Results](#) for further information.

Be sure to save all changes made to the search form.

Now, when the search user clicks the date-time field, the user can select the date and then manually enter the time or use the timepicker to enter the time. The information will also be displayed in the search results.

Because of the introduction of the new **Local Timezone** check box, InfoArchive changed the date value in the client submitted data from the format 'YYYY-MM-DD' to 'YYYY-MM-DD(+|-)hh:mm'. This impacts old searches that had:

- The date field in search form
- The date value parsing xQuery function in the main search xQuery for a table archive application.

If users encounter this issue, change the date value parsing xQuery function to the following:

```
declare function local:getDateTime($date as xs:string) as xs:string {
 if (contains($date, 'T')) then $date else concat(substring($date,1,10),
 'T00:00:00', substring($date,11)) };
```

## Searches and the Cache-In/Cache-Out Feature

### Synchronous Search

If at least one AIP that is needed for a synchronous search (for example, fall within the partition keys used as criteria for this search) is not cached in, the synchronous search is not available.

The user is prompted asking if a background search should run instead:

**SEARCH IN BACKGROUND**

This search can't be run synchronously because some results are cached out and can only run in background. Do you want the search to run in background?

Name\*

After submission, check Background Requests tab in top navigation for results.

**CANCEL SEARCH** **START BACKGROUND SEARCH**

In the above screen shot, the search wants to retrieve all the calls that have been archived, but one AIP is cached-out.

The user can still perform a synchronous search if criteria is used that does not target cached-out AIPs.

Search Forms	Collections	Retention Sets	Hold Sets	Purge Lists	Application Info	Packages	Holdings
Applications	PhoneCalls	In Test	Phone Calls By Date Range	Results			
Call Start Date: 2008-01-08 (+08:00) - 2016-04-10 (+08:00) <input type="button" value="Modify"/>							
Select all 16							
Displaying 1 - 10 of 16 <span style="float: right;">Page <input type="text" value="1"/> of 2</span>							
Customer ID	First Name	Last Name	Call Start	Call To Phone Number	Attachment	Representative ID	
000017	Elizabeth	Allen	2008-11-18 9:22:29.1...	232950412	<a href="#">Download</a>	017	<input type="checkbox"/>
000355	Natalie	Richards	2008-11-21 2:51:40.1...	1061963805	<a href="#">Download</a>	015	<input type="checkbox"/>
000564	Louis	Foster	2008-11-30 11:54:11....	1114860461	<a href="#">Download</a>	001	<input type="checkbox"/>
000556	Jesus	Baker	2009-11-02 7:17:45.1...	1772454789	<a href="#">Download</a>	004	<input type="checkbox"/>
000013	Morgan	Stevens	2009-11-03 12:18:33....	1185292351	<a href="#">Download</a>	023	<input type="checkbox"/>
000297	Leah	Butler	2009-11-27 1:53:01.1...	1601707039	<a href="#">Download</a>	009	<input type="checkbox"/>
000391	Avery	Thompson	2010-01-01 9:35:46.0...	1255846933	<a href="#">Download</a>	020	<input type="checkbox"/>
000528	Nathan	Robinson	2010-01-02 10:39:35....	1114860461	<a href="#">Download</a>	001	<input type="checkbox"/>
000457	Josiah	Wright	2010-01-10 5:54:14.0...	1716726739	<a href="#">Download</a>	035	<input type="checkbox"/>
000147	Maria	Parker	2010-01-14 8:23:06.0...	1114860461	<a href="#">Download</a>	001	<input type="checkbox"/>

In the above screen shot, the Call Start date criteria has been changed so only calls passed after 2008 are taken into account. Call Start Date is a partition key.

The only AIP currently cached-out holds only calls passed before 2008. Therefore, the results are available synchronously.

## Background Search

Whether or not all of the AIPs are available, the **background search process remains the same** to the user:

- The user launches a background search, either directly or because the synchronous one is unavailable.
- The background search will take into account all of the needed AIPs. If some of them are cached-out, it will cache them in.
- The results of the background search are exactly the same.

In the screen shot above, the background search launched in the previous section is completed even if it was targeting a cached-out AIP.

	Customer ID	First Name	Last Name	Call Start	Call To Phone...	Attachment	Representati...
▶	000467	Lucas	White	2001-11-02T21:06:39...	351244391	<a href="#">Download</a>	007
▶	000345	Tristan	Clark	2001-11-14T05:21:15...	1531838353	<a href="#">Download</a>	013
▶	000549	Alexa	Cooper	2001-11-26T10:17:18...	1730802154	<a href="#">Download</a>	017
▶	000388	Jack	Smith	2002-11-16T16:57:42...	33303518	<a href="#">Download</a>	003
▶	000068	Mia	Turner	2002-11-18T16:22:04...	1008136841	<a href="#">Download</a>	016
▶	000479	Christian	Singh	2002-11-10T19:38:34...	1453609275	<a href="#">Download</a>	022
▶	000564	Henry	Foster	2003-11-15T20:25:05...	1255846933	<a href="#">Download</a>	020
▶	000022	Allison	Green	2003-11-16T21:20:31...	1637939699	<a href="#">Download</a>	033
▶	000435	Alexander	Patel	2003-11-19T02:25:32...	1255846933	<a href="#">Download</a>	020
▶	000458	Matthew	Fisher	2004-11-08T17:17:11...	412917315	<a href="#">Download</a>	018

◀ 1 2 3 4 ►

When clicking View, the result contains all the data.

If a user executes a background search, but the search has been deleted, a message is issued when the user tries to view the search results.

Once a background request has been initiated, it is not possible to cancel it.

## Limiting Access to an Application or Search

Users should have enough access to information and functionality for them to perform their individual functions. Therefore, you may want to limit the access of a search in the Audit application, for example, because audits may provide information about applications that end users are not able to access.

There are two methods of limiting access to a search in an application:

1. The Developer or Administrator can apply group restrictions via the Administration > Permissions tab; or
2. The Developer can apply restrictions to a particular search.

The first method restricts access to the entire application whereas the second method simply hides an individual search.

The following procedures use the Audit application. Apply these same steps to restrict access to a search in any application.

For the first method:

1. In the Administration > Permissions tab, select **By Application, Audit** and **Show all groups** from the drop-down lists.
2. In the **Restrict Access To** column, select the groups that will be able to access the Audit application and its available searches.
3. Click **SAVE**.

Only the groups you selected in the previous step will be able to access the Audit application.

For the second method:

1. Select the **Audit** application.
2. Edit the search that you are applying restrictions to.
3. On the **Permissions** tab of the search, in the **Restrict Access To** column, select the groups that will be able to access the selected search.
4. Click **SAVE**.

Only the groups you selected in the previous step will be able to access the search.

## Searching Content in AWS Glacier

When searched content is stored in an offline store, such as AWS Glacier, the search result will look different. The content cannot be downloaded immediately until it is restored, so the download link specified for the search results will not be available and you will see a corresponding warning sign.

The screenshot shows a search results table with columns: Last name, First name, and Attachment 1. A row for 'Anderson, Lucy' has an orange warning icon in the Attachment 1 column. A tooltip over the icon provides details: 'Content will take longer to be restored.' and 'If you want these content you can initiate a request below.' It also mentions 'A background request will be made to bring back search results along with content.' At the bottom right of the tooltip is a blue button labeled 'GET RESULTS WITH CONTENT'.

Last name	First name	Attachment 1
Anderson	Lucy	

**CLICK FOR OPTIONS**

Content will take longer to be restored.  
If you want these content you can initiate a request below.  
A background request will be made to bring back search results along with content.

**GET RESULTS WITH CONTENT**

Click **Click for options** to view details about restoration of the content. Next, click **Get results with content** to launch a background search for the content to be restored.

In the **Background Requests** tab you will then see the new background search with a status of **In Progress**.

Name	Status	
BS_Restore_Content	In Progress 2 Aips have offlin...	 <span style="color: red;">X</span>

The search will be completed when the content is restored from Glacier to Amazon S3. The time for the restoration is configured in the corresponding S3 with Glacier store in the Administrator section (refer to [Configuring AWS S3 with Amazon Glacier](#) for more information).

**Note:** The restoration process will impact the entire AIP, even if you launch the restoration background search for a single record. So, after the restoration, all content of the restored AIP will be available for download.

# Search Troubleshooting

Issue	Resolution
You run a search but it fails. You delete and re-ingest the application data, and you still cannot run the search.	Refresh the page and try running the search again. If it still does not work, navigate away and then back to the page.
Table search is slow.	<p>It is necessary to understand search is performed in two steps:</p> <ol style="list-style-type: none"> <li data-bbox="835 566 1393 656">1. Accessing the actual DOM element and retrieving the values with XQuery from xDB.</li> <li data-bbox="835 692 1393 783">2. Appending the result node to the resulting DOM document where we cache our search results.</li> </ol> <p>Complete the following to improve table application search performance:</p> <ul style="list-style-type: none"> <li data-bbox="835 910 1393 937">• Make sure indexes are used;</li> <li data-bbox="835 973 1393 1085">• Try to reduce the amount of search result set (the bigger result set, the slower the search speed because of number 2 step above).</li> <li data-bbox="835 1121 1393 1248">• Try to reduce the amount of result columns (the more columns in the result list, the slower the search will be because of the number 2 step in search).</li> <li data-bbox="835 1284 1393 1311">• Review load balancing topology.</li> <li data-bbox="835 1347 1393 1374">• Review external IT factors.</li> </ul>

Issue	Resolution
SIP search is slow	<p>To improve SIP application search performance, try the following:</p> <ul style="list-style-type: none"> <li>• Ensure that the indexes and partitioning keys for the search fields are defined</li> <li>• Review the number of AIUs in an AIP. The application and holding configuration depends on number of AIUs per AIP. A low number (1 to 10,000) of AIUs per AIP can slow down the search if the PRIVATE ingestion mode was selected. Consider using the AGGREGATE mode if this is critical for your data.</li> <li>• Ensure you use the optimal ingestion mode (private, aggregate or pooled).</li> <li>• Check the partitioning key strategy. InfoArchive search is a two-tiered search. The first tier selects a subset of AIPs based on the defined partitioning keys. The second tier executes XQuery on the selected AIPs. It is optimal if the first tier selects around 200 to 300 AIP items in a synchronous search so that the second tier is executed quickly. In case there are more items that are supposed to be returned, consider an asynchronous (background) search.</li> </ul> <p>In other words, when defining the partitioning keys, it worth having estimated the amount of AIPs returned by the first tier for the given search criteria.</p> <ul style="list-style-type: none"> <li>• Ensure that structured data encryption is not used without need.</li> <li>• Avoid using full-text search unless it is required, as it is slow by definition.</li> <li>• Enable DEBUG log level for all components (services level and UI level) and repeat the ingestion or the search. Investigate the timing. The logs contain timing metrics. Based on the timing, you can pinpoint the part of application configuration that must be improved.</li> <li>• Take XQuery generated by InfoArchive for the search and try to execute it with xDB Admin directly. See if indexes are taken into account. Use the xDB profiling tool.</li> <li>• The AIPs are offline so it takes longer to bring them online.</li> </ul>

# Chapter 5

## Administration

### Generating SIPs

The SIP .zip is the container that is submitted to InfoArchive for archiving. The SIP .zip is a ZIP file that contains:

- At least one descriptor eas\_sip.xml (SIP); and
- One XML file that contains the records eas\_pdi.xml (PDI).

If the metadata is associated to some content (CI), the content must be put in the SIP ZIP container at the root level.

### Connectors

InfoArchive's simple XML-based ingestion template and integration framework offers a simple, open integration interface.

Table	Data Record	File	Compound
Any ETL tool*	Any ETL tool*	Any ETL tool*	Any ETL tool*
	Oracle, BD2, SQL	InfoArchive Documentum Connector	InfoArchive Connectors
	Asset Suite for InfoArchive Oracle, SQL	InfoArchive SharePoint Connector	Partner Connectors
		Kazeon File shares	
		Crawford Print Stream, reports	
		FME Migration Centre Filenet, Notes, SharePoint, Opentext, Alfresco, File shares	

\* Includes Talend, Powercenter, Datastage (Infosphere Information Server), Pentaho, AB Initio, Clover, Data Integrator and BO Data Integrator.

## Application and Platform Examples

InfoArchive preserves application data. Data is typically extracted via the application API so the underlying database and hardware platform is not a limiting factor.

Application	Platforms	Databases
Lotus Notes	Mainframe	Oracle
SharePoint	AS400	BD2
Documentum	Windows	SQL
PeopleSoft	Unix	XML databases
Baan	Solaris	ADABAS
BASE T24	VMS	
ASG Mobius	LINUX	
ERPs		
Financial Applications		
HR Systems (multiple)		
Core Banking Applications		
Customer Statement Applications		
Healthcare Applications		
Life Sciences Applications		

## Managing Packages

### Using the Packages Tab

The **Packages** tab allows the user to perform the following actions:

- View a list of the available AIPs in an application. Use the four filters to filter by:
  - Holding
  - AIP phase
  - Reception date
  - Errors
- The Retention Manager can apply a retention policy or hold to an AIP;
- Reject or invalidate an AIP; and

- Cache-in/cache-out
- Request closing of pooled library

Each package is displayed in a table that contains the following information:

Column	Description
Type	<p>Indicates whether a particular package is:</p> <ul style="list-style-type: none"> <li>• An open aggregate type</li> <li>• A closed aggregate type</li> <li>• A standard AIP type</li> </ul> <p>Refer to <a href="#">How to Read the Type Column in the Packages Tab</a> for more information that explains the icons used in this column.</p>
Name	The name of the AIP. Click the link to view the details of the package. If the package is an aggregate of multiple AIPs, click the link to view the AIPs that comprise the aggregate.
	<p>A menu that allows a user to perform the following actions on a selected AIP:</p> <ul style="list-style-type: none"> <li>• Apply retention</li> <li>• Apply hold</li> <li>• Reject package</li> <li>• Invalidate package</li> <li>• Request the closing of the pooled library</li> <li>• Cache-in/Cache-out</li> </ul> <p>The actions displayed in the menu depend on your user role as well as the State of the package. For more information, see <a href="#">Applying Actions to a Package</a>.</p>

Column	Description
Phase	<p>Displays the current phase of the AIP:</p> <ul style="list-style-type: none"> <li>• Reception</li> <li>• Waiting Ingestion</li> <li>• Ingestion</li> <li>• Waiting Commit</li> <li>• Completed</li> <li>• Reject</li> <li>• Invalid</li> <li>• Purge</li> <li>• Aggregate</li> </ul>
Holding	Indicates the name of the holding the SIP was ingested into.
Reception Start Date	Indicates the reception date of the SIP.
Online	Indicates if SIP is cached in or out
Retention	Indicates whether a retention policy has been applied to the AIP.
Hold	Indicates whether a hold has been applied to the AIP.
Records	Indicates the number of records in the AIP.

View additional information by clicking on a package name. The **Information** tab contains the custom properties of the selected AIP, package content (`sip.xml`, logs files, `ci.container` file, etc.).

If the store of the content has content offline capacity, such as Glacier, then some content will not be available. A restore button will be displayed in order to restore the content. This may take time, depending on the setting of the store.

The Retention Manager also has the ability to select one, multiple or all packages in an application. A box appears in the column on the left side of the screen for each package:

- Click Select all # to select all of the packages listed in an application.
- Click the empty box in the column header to select all of the packages that appear in the current screen.
- Select a specific package's box or multiple boxes.

Retention policies or holds can be removed from the selected packages.

## How to Read the Type Column in the Packages Tab

The Type column contains one of the following icons that define the package:

Icon	Description
	This icon is displayed if the package is an open aggregate type.
	This icon is displayed if the package is a closed aggregate type.
	This icon is displayed if the package is a standard AIP type.

## Retention and ECS and Centera Storages

Using ECS and Centera storages, you are not allowed to apply an earlier date to an AIP where a date is already applied.

## Applying Actions to a Package

The actions that can be applied to an AIP depend on the State of the AIP:

State	Available Actions
Reception	The following actions can be performed on the AIP: <ul style="list-style-type: none"> <li>• Invalidate</li> <li>• Apply Retention</li> <li>• Apply Hold</li> </ul>
Waiting Ingestion	The following actions can be performed on the AIP: <ul style="list-style-type: none"> <li>• Reject</li> <li>• Invalidate</li> <li>• Apply Retention</li> <li>• Apply Hold</li> </ul>
Ingestion	The following actions can be performed on the AIP: <ul style="list-style-type: none"> <li>• Invalidate</li> <li>• Apply Retention</li> <li>• Apply Hold</li> </ul>

State	Available Actions
Waiting Commit	<p>The following actions can be performed on the AIP:</p> <ul style="list-style-type: none"> <li>• Reject</li> <li>• Invalidate</li> <li>• Apply Retention</li> <li>• Apply Hold</li> </ul>
Completed	<p>The following actions can be performed on the AIP:</p> <ul style="list-style-type: none"> <li>• Reject</li> <li>• Invalidate</li> <li>• Apply Retention</li> <li>• Apply Hold</li> </ul>
Purge	<p>The following actions can be performed on the AIP:</p> <ul style="list-style-type: none"> <li>• Apply Retention</li> <li>• Apply Hold</li> </ul>
Reject	<p>The following actions can be performed on the AIP:</p> <ul style="list-style-type: none"> <li>• Apply Retention</li> <li>• Apply Hold</li> </ul>
Invalid	<p>The following actions can be performed on the AIP:</p> <ul style="list-style-type: none"> <li>• Apply Retention</li> <li>• Apply Hold</li> </ul>

## Rejecting or Invalidating an AIP

When you reject or invalidate an AIP, the records that are contained in the AIP will not be returned as a search result. A rejection and invalidation can be applied to an AIP at any time, but the implications are different if these actions are performed before or after the AIP is committed:

- Reject an AIP when you want to invalidate all AIPs that belong to the same collection. When you reject an AIP, you cannot resubmit an AIP with the same DSS as long there is one or more rejected AIPs in the repository.
- Invalidate an AIP if the wrong SIP was submitted and you want to resubmit the correct SIP with the same identifier.

When you invalidate or reject an AIP, the AIP is immediately removed from a search's scope. When the Invalidation job is executed (manually or automatically), the invalidated/rejected AIPs are processed. If the AIP is part of a DSS with more than one AIP, invalidating or rejecting an AIP may impact the other AIPs:

- If you reject an AIP, other AIPs from the same DSS will be automatically rejected during the job execution.
- If you invalidate an AIP, other AIPs from the same DSS will be automatically demoted to the 'waiting to commit' phase during the job execution.

If the AIP has not been committed, the AIP is immediately destroyed. If the AIP has been committed and retention was applied (often through defining a retention class on the holding), then the AIP needs to be disposed regularly.

When the Confirmation job runs, it confirms that some events on packages has occurred (Receive, Storage, Purge, Invalidation).

If you had rejected a package, and the Disposition job had marked some packages for purge, running the Confirmation job confirms everything outstanding (and that the next run of the Disposition job will cause the packages to be deleted).

You are able to reject an AIP if the AIP was part of a DSS with more than one other AIP. The AIP must also be in one of the following phases:

- Waiting Ingestion
- Waiting Commit
- Completed

You are able to invalidate an AIP if the returnCode is 'OK'. The AIP must also be in one of the following phases:

- Waiting Ingestion
- Waiting Commit
- Completed

To reject or invalidate an AIP:

1. Select **Packages** tab.
2. Select the AIP that is being rejected or invalidated.
3. Call the context menu by clicking the  and selecting either:
  - Reject Package or
  - Invalidate Package
4. Select the Reason why the AIP is being rejected or invalidated.
5. If desired, enter any pertinent information in the Comment field.
6. Click **Reject** or **Invalidate**, depending on your selection in step 3.

## Applying a Retention Policy to an AIP

1. Select the AIP the retention policy is being applied to.



2. Call context menu by clicking and selecting **Apply retention**.
3. Select the retention policy you want applied to the application and click **Next**.
4. Review the retention policy details to verify that it is the correct policy to apply to the application. The fields that are displayed depend on the Aging Strategy of the retention policy. For more information, see [Creating a Retention Policy](#).
5. Enter the following information and click **Next**:
  - **Retention Set Name**: Enter a unique name for the policy.
  - Enter a **Description** for the policy.
6. Review the information you have entered. When satisfied that the information is correct, click **Finish**.

The AIP now indicates that there is a retention policy applied to it.

## Applying a Hold to an AIP

1. Select the AIP the hold is being applied to.



2. Call context menu by clicking and selecting **Apply hold**.
3. Select the hold you want applied to the application and click **Next**.
4. Enter the following information and click **Next**:
  - **Hold Set Name**: Enter a unique name for the hold.
  - Enter a **Description** for the hold.
5. Review the information you have entered. When satisfied that the information is correct, click **Finish**.

This is an asynchronous operation. The AIP will not indicate that there is a hold policy applied to it until the order completes.

## Dashboard

The Dashboard allows Administrators and Retention Managers to easily view information related to applications, such as the number of retention policies, holds and purges. The Administrator can also use the Dashboard to view how much storage is in use and how much is remaining.

If the user does not have the right to access the Dashboard, a list of applications that the user has access to is provided.

## Storage Metrics

This section outlines how the Dashboard reflects storage updates.

### Calculation for Pricing

For XML metadata of both SIP and table files, the volume size for pricing is calculated by counting all characters of an element and attribute content. Characters used for XML formatting purposes are not counted. For example, in the following XML, only the characters in bold are used for pricing: Each character will be counted as 1 byte, even though a UTF-8 character can consist of multiple bytes. This character count is executed during ingestion and before encryption.

```
<BASEBALL>
 <ALLSTARFULL>
 <ROW>
 <PLAYERID>aaronha01</PLAYERID>
 <YEARID>1955</YEARID>
 <GAMENUM>0</GAMENUM>
 <GAMEID>NLS195507120</GAMEID>
 <TEAMID>ML1</TEAMID>
 <LGID>NL</LGID>
 <GP>1</GP>
 </ROW>
```

For SIP archiving:

- The characters of PDI XML files are calculated.
- The size of the additional content files contained in the `sip.zip` file are calculated.

For table archiving:

- The characters of table XML files are calculated.
- The size of the attached content files are calculated.

The size of the original content file is measured before ingestion.

## Storage Footprint Calculation

In the Dashboard, the storage footprint is displayed per application.

- For SIP archiving, the XML metadata size is the sum of the xDB library segment file size of every single AIP.
- For table archiving, the XML data metadata size is the sum of the xDB library segment file size of every single table schema.
- The index size is the sum of the index size of all AIP/Schema xDB libraries.
- The content size is the sum of all associated content of a table/AIP (reversibility formats, CI container, RI XML, SIP XML, xDB Library back up, etc.).

## Licensed Volume

The Administrator can review the licensed volume information for all applications. The Dashboard also provides a breakdown of the licensed volume information for each application.

Licensed volume refers to how customers are charged for the product:

- For a table archive, customers are charged per byte for the structured data that has been archived.
- For a SIP archive, customers are charged based on the amount of unstructured content that has been archived (for example, MP3 or GIF files).

The Dashboard for the Administrator reflects the pricing breakdown for all applications:

- The first pie chart reflects the ratio of content versus data is displayed in megabytes.
- The second pie chart reflects the active archive versus decommissioned archive sizes in megabytes.

The Dashboard allows the Administrator to breakdown the licensed volume information on an application by application basis in a series of bar charts.

The Administrator can order the bar charts according to:

- The total size each application's archived content and data, from largest to smallest.
- The names of each application are displayed alphabetically.

The Administrator can toggle this information between all applications or review bar graphs for all decommissioned applications or active archives.

## Questions & Answers – Pricing

After ingestion how is volume calculation kept in check?	All information is computed during the ingestion and saved into the system repository to be used later.
How does encrypted information impact licensing volume calculations?	Encryption does not affect licensing volume considerations, but it can increase the storage footprint.
How does AIU or record disposition impact the volume calculation?	When the AIUs or records are disposed of, the licensing metrics are updated to reflect the disposition. The storage footprint is not impacted.
How is cache in/cache out considered in the licensing volume calculation?	Cache-out/cache-in does not have an impact on the licensing.
Is the storage of auditing information, the system data repository and the managed item data repository included in the calculation of the storage footprint?	No, the Dashboard does not expose these repositories.

Does the system keep track of the size of all other unstructured information (for example, raw XML, renditions, xDB back up)?	Yes, the other unstructured information is included in calculating the content size. It is reflected in the storage footprint.
What jobs are required to ensure that the correct data is displayed on the Dashboard?	The administrator needs to manually execute or schedule the Refresh Metrics job to view the most recent information.

## Logging

InfoArchive log files will be created in the `logs` directory and rollover into gzip archives in the `archives` subdirectory once a log file reaches 10 megabytes in size. There is also another subdirectory called `context-sifting` that is used by the InfoArchive logging functionality related to job and order item logs:

- Startup logging, as well as most errors, will be printed on console.
- Errors will be logged to the `errors.log` file.
- Context (request, job, order item, scheduled task, etc.) information is logged in the `context-mapping.log` file.

This information includes an ID, type correlation and user, as well as optional context-specific fields (for example, the full request URI for requests or job details for jobs, etc.).

Only a single message per context will be logged here.

Each log message in any of the other log files typically includes the context-id, which can be used to lookup the additional context information in this file. The same context-id can be used to correlate multiple log messages when analyzing the logs.

- Various other log files are maintained to distribute logging based on log message origin or context, primarily to ease troubleshooting.

The following is an example of information in a log file:

```
[Jan 3, 2018 4:48:17 PM]: INFO
```

```
Starting job: name=Remove Policy, handler=RemovePolicyJob, properties={type=, retentionPolicyName=Policy A}, application=PhoneCallsGranular, scheduled by sue@iacustomer.com, scheduled at 2018-01-03T21:48:17.804Z, attempt=1
```

## Using a Preexisting Open Source Logging Solution

In a multi-server deployment, logs need to be stored within a unified location to easily:

- Track and manage logs across several servers; and
- Allow for querying to better understand issues.

When a job is started, the following information may be available in the first log entry:

Field	Always Present	Description
Name	Yes	Indicates the name of the job.
Handler	Yes	Indicates the handler for the job. For jobs that were cloned, this field helps identify what the job does.
Properties	No	Indicates the properties for the job. Job parameters are shown in curly brackets. For example: type =aip, retentionPolicy=Policy A.
Scheduled By	No	Indicates the user that ran the job. This field is not shown for scheduled jobs.
Application	No	Indicates the application the job instance was scoped to.  If a job definition is scoped for multiple applications, when the job runs, each job instance will be scoped to an application. If the job was scoped to run for two applications, two job instances for each application are created.
Attempt	Yes	Indicates the attempt for running the job. Value is always one unless the job has been configured to auto retry upon failure. Does not increment when doing a manual retry.

The following example illustrates how to integrate a preexisting logging solution, in this case, Graylog2, with a multi-server logging deployment.

1. Create a multi-server InfoArchive deployment. Ensure that each IA Server machine has a different host name.
2. To consume the InfoArchive logs, install and configure syslog-*ng* by running the following command:

```
sudo apt-get install syslog-ng
```

- a. Add the log files to the etc > syslog-*ng* > syslog-*ng*.conf file.
- b. Start syslog-*ng* by executing:

---

syslog-**ng**

3. Install Graylog2.
  - a. Download the latest image from the Graylog website and deploy it within a virtual machine.
4. Configure Graylog to consume IA Server logs via syslog-**ng**:
  - a. Add the following lines to the etc > syslog-**ng** > syslog-**ng.conf** file:
 

```
Define graylog TCP syslog destination.
destination d_graylog {
 syslog("graylog location" port(5140));
};

Tell syslog-ng to send data from InfoArchive sources to the
newly defined syslog destination.
log { source(s_ia_log); destination(d_graylog); };
log { source(s_ia_error); destination(d_graylog); };
```
  - b. Update the Graylog server:
    - i. Log into the Graylog web application.
    - ii. Configure input using the process described in the Graylog documentation. Make sure to use the Syslog TCP input and bind the address to 0.0.0.0.
5. Start up IA Server servers. The logs should appear in the configured input after a few minutes.

## Generating Job Instance and Order Item Logs

Whenever a job or order item is executed:

- Activity is logged in the regular log files; and
- Activity is also captured in a separate log file dedicated to that specific job instance or order item.

In the application.yml file, the logging.level. property specifies the global log level threshold. This log level controls the type of messages that are written to the various log files.

The log messages in the dedicated log files, however, are not affected by the global log level threshold, and will instead always be written to the dedicated log files, regardless of log level.

Logs are archived into the RESULT content store defined by the associated application. While order items are generally application-specific, jobs run system-wide and are, therefore, not associated with an application defining the content store to persist such logs.

The System IA App is introduced to define the necessary content store to persist job logs. You install the System IA App as part of installing the core IA Apps. For more information, see the *InfoArchive Installation Guide*.

If the System application is not installed, the dedicated logs will be stored in the <INFOARCHIVE\_ROOT> > logs > context-sifting directory. In this case, the logs are not accessible through either the IA Server REST API or the IA Web App.

To verify installation success, inspect an individual job run that ran after installing the core IA Apps. Such a job instance will have a **DOWNLOAD LOGS** link, while previous job runs would not. If no

such link is displayed, look for the job logs as described for when the System IA App is not installed. If the logs are generated there after installing the core IA Apps, the System IA App is most likely misconfigured and could not be used to archive the logs.

## Downloading Job Instance and Order Item Logs

The ability to download job instance and order item logs is restricted to users who have permission to download content and are allowed to view the job or order item the logs are associated with.

To download the dedicated log file for a specific job instance:

1. On the **Administration > Jobs** tab, click the link in the **Job Name** column to access the desired job's log files.  
You can determine which job instance's log files you want to access by reviewing the **Scheduled Date** column.
2. Click the **Status** link for an individual order item or job instance.
3. Click **DOWNLOAD LOGS**.

To download the dedicated log file for a specific order item instance:

1. On the **Background Requests** tab, click the link in the **Status** column to access the desired order item's log files.
2. For order item logs, click **DOWNLOAD DIAGNOSTIC LOGS**.  
For job instance logs, click **DOWNLOAD LOGS**.

## Configuring Jobs

### Job Scoping

Jobs can be system-scoped or scoped to one or more applications. Depending on the job, it may support one or the other or both.

### Application Scoping

If a job definition is application-scoped, when the job is executed or scheduled, a job instance is created for each application that was specified.

If the job was scheduled, the job will normally automatically create new instances for the next run (for each scoped application), depending on the status of the job, the job may not be scheduled.

The Admin client no longer supports the deletion of scheduled job instances. Instead, stop the schedule, as this will remove all of the scheduled job instances for each of the scoped applications.

If an application is deleted or disposed, for any job definitions that were scoped to the application, the job definition will no longer be scoped to that application. If this was last application-scoped to the job, the job cannot be run.

## System Scoping

If a job is system-scoped, only one job instance will be created when the job is started or scheduled.

## Working with Jobs: List of Available Jobs

The section provides a summary of the jobs that are included with InfoArchive. While a job is running, do not click the 'i' (information) button. Otherwise, you will receive an error message.

Job Name	Schedule	Supported Scopes	Description
Apply Retention Policy To Records	Manual		<p>Applies a specific retention policy to records matching a search criteria.</p> <p>Needs to be scoped to an application. The job also requires an XML file to be created on IA Server to indicate the search criteria. Records that already have the policy will not have the policy applied again.</p>
Apply Hold Rule to Records	Manual	Application only	<p>Provides the ability to run rules previously uploaded for applying a hold. The job parameters allow the ability to specify a search criteria that limits which records are evaluated.</p> <p>Provides the ability to define rules to apply hold to records.</p> <p>The parameters are very similar for this job as the apply retention using rule job When creating the rule, use the type <b>APPLY_HOLD</b> instead of <b>APPLY_RETENTION</b></p>
Apply Retention Rule to Records	Manual	Application only	Provides the ability to use previously uploaded rules for applying retention. Job parameters allow the ability to specify a search criteria that limits which records are evaluated.

Job Name	Schedule	Supported Scopes	Description
<a href="#">Archive Audits</a>	Daily	System only	<p>Archives audits so that they can be searched via the Audit application.</p> <p>Requires the audit application to be installed. After this job runs, the audits are purged so REST calls to fetch the audits will not return the archived audits.</p>
CacheOut	Manual	All	<p>Cache out libraries from Applications who has reach their metadata max size</p> <p>Provides the ability to reduce the metadata footprint in xDB by removing unused AIP libraries. Important to have this job scheduled if the limit data in cache option is enabled on one or more SIP applications. The recommendation is to schedule this job every 5-15 minutes.</p> <p>The default interval value for the CacheOut job has increased from 5 to 15 minutes. This increase on the interval is only for an upgrade if the interval was not changed from the default value. The interval value is the number of minutes that must pass before the job can be repeated. On an upgrade, the schedule will only be updated if the value has not changed from the 5 minute default value.</p>
<a href="#">Clean</a>	Every 15 minutes	All	<p>Frees up resources, such as orders, search results and AIPs.</p> <p>It is important to have this job scheduled.</p>
<a href="#">Clean up Purge Candidate Lists and Applications</a>	Weekly	System only	Deletes purge candidate lists that have been cancelled or disposed. Also removes disposed applications after the Clean job has run.

Job Name	Schedule	Supported Scopes	Description
<a href="#"><u>Close</u></a>	Manual	All	<p>Closes eligible xDB libraries and aggregates.</p> <p>Important to have this job scheduled only if the SIP ingestion modes AGGREGATE or POOLED LIBRARY are used. The recommendation is to schedule this job every 15 minutes.</p> <p>This job is meant for SIP applications only.</p>
Commit	Manual	All	<p>Commits packages in waiting commit.</p> <p>Important to have this job scheduled only if SIP of the same DSS (seqno &gt; 1) are ingested. The recommendation is to schedule this job every 15 minutes.</p>
Confirmation	Every 15 minutes	All	<p>Confirms that some events on packages occurred (Receive, Storage, Purge).</p> <p>Failure to do this means that the AIPs remain on the system, although searches against partially disposed AIPs will not return search results.</p>
Dispose Purge Candidate List	Weekly	Application	<p>Disposes items in approved purge candidate lists.</p> <p>It is important to have this job scheduled. Once an application is active, this job is the only way to remove content from archive.</p> <p><b>Note:</b> This job calls an internal job that is not accessible via rest that initiates a backup.</p>
<a href="#"><u>Generate Purge Candidate List</u></a>	Weekly	System only	<p>Generates a purge candidate list for items that are eligible for disposition.</p> <p>It is important to ensure that the Dispose Purge Candidate List job runs on a similar schedule. If the Generate Purge Candidate List runs before the Dispose Purge Candidate List job, purge lists that are under review will be marked cancelled.</p>

Job Name	Schedule	Supported Scopes	Description
Invalidation	Every 15 minutes	All	<p>Invalidates packages marked as invalid.</p> <p>Supports both system and application scoping.</p> <p>Do not scope this job to a table application.</p>
Post Ingestion Processing	Manual		Performs post-processing of AIP after ingestion.
Process Retention Events	Manual	Application only	Processes triggered events to update the qualification date for policy applications and the estimated disposition date for retained sets.
<a href="#">Refresh Metrics</a>	Daily	System only	<p>Updates the metrics for the dashboards and updates statistics for the Reports application.</p> <p>The calculation is based on the total number of records in the system. If any table applications are installed, including the Reporting application, every row in every table across all schemas and databases is considered a record. Because the sum is calculated based on the total number of records, the percentage of records under retention is zero, as a large number of rows in tables are usually not under retention.</p> <p>Must be scoped to the system. Does not support application scoping.</p> <p>There are three scenarios in which the information will not be available from the server and the UI will show an empty list of applications:</p> <ul style="list-style-type: none"> <li>• If the customer upgrades the system but does not refresh the metrics.</li> <li>• If the customer did not install any application and refreshed the metrics.</li> <li>• The metrics were never refreshed.</li> </ul>

Job Name	Schedule	Supported Scopes	Description
<a href="#">Remove Policy</a>	Manual	System or Application	<p>Removes the named retention policy from items. Can be limited to a type of retention policy.</p> <p>Only one retention policy can be specified, case-sensitive.</p>
<a href="#">Requalification</a>	Manual	System or Application	<p>If retention policies are changed and the changes are retroactive, this job updates the retention information for all managed objects.</p> <p>Some storage systems may not support updating qualification dates.</p>
Table Indexing	Automatic (cannot be scheduled or run manually)	System only	<p>Performs table indexing.</p> <p>Provides visibility on indexing efforts whenever table indexing operations are requested.</p>
<a href="#">Trigger Event Policy</a>	Manual	Application only	<p>Provides an ability to trigger events using an XML file for records using event or mixed retention policies.</p> <p>Requires an XML file to be placed on IA Server to indicate the event, event context, and date that the event happened (or will happen).</p>
Trigger Event Rule	Manual	Application only	Provides the ability to use rules previously uploaded for trigger event rules to start aging for records under an event or mixed retention policy. Job parameters allow the ability to specify a search criteria that limits which records are evaluated.

## Archive Audits Job

When an audit object is created in the system, you cannot execute a search against it. The objects are collected in a temporary storage until they are archived. The Archive Audit job must be executed to allow audit objects to be searched.

Audits are organized in SIP packages by day (for example, only audit for the same day will be put in a SIP). This allows you to apply a retention policy to the archived audits and allow for proper disposition of audits.

The following parameters can be configured:

Parameter	Description
<i>MaxSipAuditEntries</i>	This is the maximum number of audit entries that will be in a SIP package. The default is 50,000 entries.
<i>Start Date</i>	Specify a date to start selecting audits from. Leaving this date empty will default to today's date. The format is YYYY-MM-DD.
<i>End Date</i>	Specify a date to end selecting audits. Leaving this field empty will default to today's date. The format is YYYY-MM-DD.

The start and end date are used to allow for archiving audits that are not current.

The normal operation is to leave the date empty and allow the job to archive all audits for the day. The job will also go back 30 days searching for audits to archive. It will stop once it finds that there are no audits to archive for a day.

This job should not be application scoped.

## Clean Job

The Clean job frees up resources, such as orders, search results and AIPs. It is important to have this job scheduled.

As of the 16EP4 release:

- The default interval value for the Clean job has increased from 5 to 15 minutes. This increase on the interval is only for an upgrade if the interval was not changed from the default value. The interval value is the number of minutes that must pass before the job can be repeated. On an upgrade, the schedule will only be updated if the value has not changed from the 5 minute default value.
- The parameter (*emptyEventPhase*) was added to be able to clean events that are no longer referenced by any managed items. Refer to the table below for more information about the *allPhases* parameter.

The Clean job removes events from the system if no managed items are referring to the event.

You may not want to enable the cleaning of events if you plan on setting event dates before retention is applied to records.

The following parameters can be configured:

Name	Description
<i>allPhases</i>	Type: Boolean  Default Value: True  Value to use for all phases. If the value is set to true, the other parameters are ignored, and the Clean job evaluates all phases of work To disable certain phases, set this parameter to false, and then set the phase to be skipped to false.
<i>aipPhase</i>	Type: Boolean  Default Value: True  If true or <i>allPhases</i> = true, delete all PRUNE AIPs + rejected and invalidated AIP without commit date.
<i>contentPhase</i>	Type: Boolean  Default Value: True  If true or <i>allPhases</i> = true, delete all orphaned contents.
<i>searchResultPhase</i>	Type: Boolean  Default Value: True
<i>orderItemPhase</i>	Type: Boolean  Default Value: True  If true or <i>allPhases</i> = true, delete all expired order items.
<i>aiuContentPhase</i>	Type: Boolean  Default Value: True  If true or <i>allPhases</i> = true, cleans the AIU content to free up space when pruning AIPs.
<i>emptyEventPhase</i>	Type: Boolean  Default Value: True  If true or <i>allPhases</i> = true, delete all events that are no longer referenced (retention removed).  If setting an event fulfillment (using the job or via REST) before records have been ingested, it is recommend that you disable the <i>emptyEventPhase</i> parameter.

## Clean Up Purge Candidate List and Applications Job

The Clean Up Purge Candidate List and Applications job has two responsibilities: it cleans up cancelled or disposed purge lists and finishes disposition of applications after the initial disposition had marked all of items in the application and the clean job had run.

The following is the process to dispose of an application, assuming nothing in the application is under hold or under longer retention:

1. Run the Generate Purge Candidate Lists job.
2. Approve the purge candidates list for the application.
3. Run the Disposition job.
4. Run the Clean job.
5. Run the Clean Up Purge Candidate List and Applications job.

This job cleans up purge lists for applications so it is recommended that purge lists processed by the Disposition job be reviewed before running this job.

This job cannot be scoped to an application.

## Close Job

Closes eligible xDB libraries and aggregates. It is important to have this job scheduled because indexes are created on that library only after the Close job has been executed.

The default interval value for the Close job has increased from 5 to 15 minutes. This increase on the interval is only for an upgrade if the interval was not changed from the default value. The interval value is the number of minutes that must pass before the job can be repeated. On an upgrade, the schedule will only be updated if the value has not changed from the 5 minute default value.

The following parameters can be configured:

Parameter Name	Description
<i>phaseToProcess</i>	Type: ALL_PHASES/POOLED_ONLY/AGGREGATE_ONLY  Default Value: ALL_PHASES  Mandatory: No
<i>closeDelay</i>	Type: Integer  Value: positive integer (>= 0)  Default Value: 5 minutes (300 seconds)  Mandatory: No  Indicates the delay (in seconds) to wait after the eligible close date before processing the AIP.
<i>threadNumber</i>	Type: Integer  Default Value: 10 – If the CPU utilization is considered too high, change this value to 5.  Indicates the maximum number of parallel threads.

Prior to running the Close job, limit (open files) values with Linux IA Server. For example, if the customer will ingest 10,000 SIPs during the day and close them at night, this value should be set to more than 30,000. Failure to reset the value will cause the job to fail. Furthermore, the job will restart after each failure, impacting the CPU usage of the IA Server and system performance.

When closing more than 50,000 AIPs in one aggregate, the maximum java heap size for IA Server should be increased to 8GB to avoid an out of memory error.

Parameter Name	Description
<i>cancelApproved</i>	Flag on whether to cancel approved purge list that was not disposed. For upgraded systems, this value is set to true (to match original behavior).  Type: Boolean  Default Value: false

Mandatory: No

## Generate Purge Candidate List

The Generate Purge Candidate List job will generate purge lists for records that are eligible for disposition. Once disposition has been run to dispose of the records, the purge list status will be set to disposed. The Generate Purge Candidate List job sets any purge lists that have not been processed by the disposition job to cancelled and generates new lists.

## Post Ingest Processing Job

After data has been ingested and application configuration has changed, the Post Ingest Processing job recomputes partitioning keys and indexes for the already ingested AIPs, which helps alleviate performance issues during searches on a large amount of data.

There are three scenarios in which you would run the job:

1. If, after being in production for some time, and the criteria of a search has changed. This type of change can hinder search speed. To resolve the issue, you can add more indexes, remove old indexes and add new partitioning keys to improve search speed.
2. When the configuration of an application has been successfully tested but, when moved to a production environment, the application has ingested a large amount of data, which has impacted the speed of the application's searches. To resolve the issue, the application must be re-configured in the production environment.
3. If you did not perform performance testing in a development environment and moved an application directly to a production environment. Now the speed of the application's searches is slow. To resolve the issue, you can add additional indexes and partitioning keys, and re-index any previously ingested data.

The Post Ingest Processing job is executed manually against an application. Once the job has been executed, it selects a set of AIPs based on the defined properties and completes your specified [actions](#).

Only AIPs in the complete state are impacted by the job.

If an AIP has been cached out, once the Post Ingest Processing job is executed:

1. The cached out object is restored.
2. The Post Ingest Processing job processes the designated AIPs in the holding.
3. A backup is made of the updated AIPs, as long as the package is:
  - In the private ingestion mode.
  - Pooled with a closed library.
  - An aggregate with a closed library.

When the library is opened, no backup is performed. The backup occurs only once the library is closed.

4. The object is cached out.

It is strongly recommended that you test the Post Ingest Processing job in a development environment. If the following criteria is met, the test is successful, and the job can be moved to your production environment:

- If all the AIPs are properly updated with new indexes and partitioning keys.
- If all the data is searchable after the update.
- If there is no performance gap in searches and ingestion after the indexes and partitioning keys are updated.

The Post Ingest Processing job is designed to be safe if to execute it a second time. After the job is executed the first time, the `configurationHash` property of the AIP object is updated with a new value. If the job is executed again, the job skips all AIPs that were updated during the first run of the job.

The following information can be configured for the Post Ingest Processing job:

1. Enter a **Description** for the job.
2. The Post Ingest Processing job can only be executed manually in a single run. There is no possibility to schedule the job. In the **Schedule By** field, indicate the job's schedule by selecting **Manual**. A Developer or Administrator will then execute the job manually.
3. In the **Apply To** field, select the applications the job will be executed against.
4. Set the following **Properties**:
  - **actions**: Enter a list of actions that is executed on the selected AIPs. This property is mandatory. The defined actions are listed as comma-separated values. The order of action list is the order of the action execution on the selected AIPs. The available actions include:
    - *update-xdb-metadata-indexes*: For every selected AIP object from the system repository database, the corresponding AIP pdi file or xDB library (with customer structured data) is calculated. All of the existing indexes are removed from that file and new ones are calculated based on the available *pdi.xml* file for the application's configuration.

It is critical to good performance that re-indexing the metadata repository occurs before you recompute the partitioning keys or any other actions that uses indexes.

Depending on the ingestion mode, the indexes are created on different objects in xDB: either in the pdi document file (*eas\_pdi.xml* – file with structured business data) or on the xDB library (container for the *eas\_pdi.xml* file):

- Private Mode: Indexes are always at the pdi document level. The xDB library is always in read-only state.
- Pooled Mode: Indexes are always at the pdi document level. The xDB library is either opened (library in read-write state or closed (library in read-only state).
- Aggregate Mode: If the xDB library is:
  - Opened: The number of AIU is small if the document has fewer AIUs than allotted in the *aiu.threshold* property. There are no indexes.

The number of AIUs is large if the document has more AIUs than allotted in the *aiu.threshold* property. Indexes are on pdi document level. The xDB library is in a read-write state.

- Closed: Indexes are at the xDB library level and the library is in a read-only state.
- *update-partitioning-keys*: For every selected AIP object from the system data repository, all of the existing partitioning keys are removed and new ones are calculated based on the available *pdi.xml* file for the application's configuration.

After updating the partitioning keys, do not forget to update AIC and query objects with the new partitioning keys, as these objects will still reflect the old search configuration.

**Note:** It is critical that re-indexing the metadata repository occurs before recomputing the partitioning keys or any other actions that use indexes. Otherwise, system performance may be impacted.

- **holdingName**: Specify the holding within the application for which the AIPs are selected. This property is mandatory.
- **pdiSchemaName**: An application can have different PDI schemas in the same holding. Enter the property value that also selects the AIPs to be processed.

If the Post Ingest Processing job fails to run, there are two places to view logs that will identify the reason why the job failed, as well as identify which parameters need to be updated prior to running the job again:

- 1. Clicking the **Post Ingest Processing** job link in the **Job Name** column.
-  2. Click the button. The reasons why the job failed are displayed.
- Review the <INFOARCHIVE\_ROOT>\logs\\*.log files that contain information about updated AIPs, page handle number and AIP date ranges.

Each time the Post Ingest Processing job has been executed, and the partitioning keys have been changed, be sure to update the AIC and query objects with the new partitioning keys.

If this is not done for the metadata indexes, and some AIPs contain old indexes while other AIPs contain the new indexes, data will still be searchable, but searches will take considerable time to execute.

Furthermore, if some AIPs contain the old partitioning keys while other AIPs contain the new partitioning keys, the search depends on the AIC and query configuration:

- If the AIC uses the old partitioning key configuration, only AIPs that were not updated during the job execution are searchable;
- If the AIC uses the new partitioning key configuration, only the updated AIPs are searchable;
- If the AIC does not use any partitioning keys, all AIPs, whether they were updated or not, are searchable.

For offline AIPs using Glacier, it is possible that, when running the Post Ingest Processing job, some resources, such as the xDB library backup, are offline. If the xDB library is offline, then there is no way to cache-in an AIP in order to update. There is no way to backup an AIP after an update in order to have a backup in synch with AIP. If a resource is offline, then the AIP is not going to be updated, but a request to become on-line is issued and the job goes into "SKIP" state, which indicates that the job is not currently doing anything. The job is run only in case all the resources are online.

In some cases, you may want to trigger the Post Ingest Processing by scripts and not by the job. The execution of the post ingest actions is performed synchronously, right after an AIP has been ingested. Refer to the REST API documentation for further information.

To satisfy compliance requirements, a post-ingest log generated. After the AIP package has been processed, then the log file is attached to the AIP package as its content. If the AIP has never been processed, then there is no such a log file. Every AIP package has its own log file.

The name of the log file is post.ingest.log.gz. It can be accessed with REST or via IA Web App:

The following is an example of the information found in the log:

```
[INFO] [2017-06-23T15:18:27.854+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestServiceImpl] -Starting post ingest processing for Aip with id:89bceb9a-b2c4-4632-
83de-c43f2c752ec6
[INFO] [2017-06-23T15:18:27.854+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestServiceImpl] -Run action 'UPDATE_XDB_METADATA_INDEXES' on aip '89bceb9a-b2c4-
4632-83de-c43f2c752ec6'
[INFO] [2017-06-23T15:18:27.858+03:00] [56] [com.emc.ia.ingestion.postprocessing.
```

---

```

PostIngestIndexUpdater] -AIP xdb library 'bb32ccf0-67aa-4c80-b6cd-a8eb48d571a1' has been
set to read-write mode.
[INFO] [2017-06-23T15:18:27.858+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestIndexUpdater] -Removing old AIP metadata indexes
[INFO] [2017-06-23T15:18:27.874+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestIndexUpdater] -AIP metadata indexes have been cleaned.
[INFO] [2017-06-23T15:18:27.890+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestIndexUpdater] -Creating new AIP metadata indexes.
[INFO] [2017-06-23T15:18:27.925+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestIndexUpdater] -AIP metadata indexes have been created.
[INFO] [2017-06-23T15:18:27.944+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestIndexUpdater] -AIP xdb library 'bb32ccf0-67aa-4c80-b6cd-a8eb48d571a1' has been
set to read-only mode.
[INFO] [2017-06-23T15:18:27.944+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestServiceImpl] -Run action 'UPDATE_PARTITION_KEYS' on aip '89bceb9a-b2c4-4632-83de-
c43f2c752ec6'.
[INFO] [2017-06-23T15:18:27.960+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestPKeysUpdater] -Removing old partitioning keys for aip '89bceb9a-b2c4-4632-83de-
c43f2c752ec6'.
[INFO] [2017-06-23T15:18:27.980+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestPKeysUpdater] -Old partitioning keys for aip '89bceb9a-b2c4-4632-83de-
c43f2c752ec6' have been removed.
[INFO] [2017-06-23T15:18:27.981+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestPKeysUpdater] -Creating new partitioning keys for aip '89bceb9a-b2c4-4632-83de-
c43f2c752ec6'.
[INFO] [2017-06-23T15:18:28+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestPKeysUpdater] -New partitioning keys for aip '89bceb9a-b2c4-4632-83de-
c43f2c752ec6' have been created.
[INFO] [2017-06-23T15:18:28+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestPKeysUpdater] -AIP '89bceb9a-b2c4-4632-83de-c43f2c752ec6' has been updated with
new partitioning keys.
[INFO] [2017-06-23T15:18:28.013+03:00] [56] [com.emc.ia.ingestion.postprocessing.
PostIngestServiceImpl] -'ConfigurationHash' property has been updated for aip '89bceb9a-
b2c4-4632-83de-c43f2c752ec6'.

```

A post-ingest log is also generated. After the AIP package has been processed, then the log file is attached to the AIP package as its content. If the AIP has not been processed ever, then there is no such a log file. Every package has its own log file.

The name of the file is `post.ingest.log.gz` and can be accessed in the Application > Packages > AIP content page:

## Refresh Metrics Job

Calculating the metrics information in the InfoArchive Dashboard can take a significant amount of time. Therefore, the Dashboard retrieves most of its information from pre-populated values and the Refresh Metrics job populates these values. The job will scan the system and populate the metrics information. You can decide how often the metrics information should be updated, as it depends on the individual use cases.

This job should not be application scoped and does not have any parameters.

## Remove Policy Job

This job does a mass removal of the policy and can be scoped to specific applications.

The following parameters can be configured:

Parameter	Description
<i>retentionPolicyName</i>	Name of retention policy that will be removed from items. Retention policy must be defined and name is case sensitive.
<i>type</i>	Optional field to specify the type in which the job will remove the policy. Possible values include: <ul style="list-style-type: none"><li>• application</li><li>• aip</li><li>• aiu</li><li>• table</li><li>• row</li></ul> Row refers to a table row.

## Requalification Job

This job is necessary to run if a retention policy is changed and the changes need to be retroactive.

Cannot be scoped to an application and does not have any parameters.

## Trigger Event Policy Job

The Trigger Event Policy job triggers events based on a trigger file that you provide. Usually this is a product of another system that generates the event (for example, HR system would indicate when the employee left the company). The values of the trigger file contain the context that groups the records together. For example, if you are keeping employee records until 5 years after the employee leaves the company, then you would want to group the records around a common field (for example, context). The context in this case would be the employee number. When the event policy is applied, a context would have been specified. When the event needs to be triggered, a context and a trigger date need to be specified.

This job has to be application scoped.

The following parameter can be configured:

Parameter	Description
<i>triggerFile</i>	<p>This is a path to a trigger file that contains a list of triggers (context, trigger date and condition). The following illustrates the format of the file:</p> <pre>&lt;?xml version="1.0"?&gt; &lt;triggers&gt; &lt;event&gt; &lt;context&gt;00457&lt;/context&gt; &lt;triggerdate&gt;2010-01 -31&lt;/triggerdate&gt; &lt;condition&gt;condition&lt; /condition&gt; &lt;/event&gt; &lt;event&gt; &lt;context&gt;00345&lt; /context&gt; &lt;triggerdate&gt;2014-02-28&lt;/triggerdate&gt; &lt;condition&gt;condition&lt;/condition&gt; &lt;/event&gt; &lt;/triggers&gt;</pre> <p>The location is relative to where the server is deployed. If multiple IA Servers are installed, it is recommended that you use a network location (versus a local path).</p>

## Populating Event Dates for the Trigger Event Policy Job

An XML file is used to populate event dates for the Trigger Event Policy job:

Name	Description
<i>context</i>	Enter the context for the event (for example, employee number).
<i>triggerdate</i>	Enter the date when event happened or is planned to happen in a YYYY-MM-DD format. A date must be within the following range: 1000-01-01 – 2999-12-31.
<i>condition</i>	Enter the name of the condition, which must match the condition specified on the retention policy. The value is case sensitive.

The following illustrates how to populate the XML file:

```
<?xml version="1.0"?>
<triggers>
<event>
<context>89</context>
<triggerdate>2016-02-28</triggerdate>
<condition>tradeverSION</condition>
</event>
<event>
<context>77</context>
<triggerdate>2016-02-28</triggerdate>
<condition>tradeverSION</condition>
</event>
</triggers>
```

## Using the Jobs Tab

The **Jobs** tab allows the Administrator and Developer to:

- Create a job or edit an existing job.
- View information for all of the jobs.
- Run a job.

Jobs are displayed in a table that contains the following information:

Column	Description
Job Name	Indicates the name of the job.  Click the name of a job to view its run history. For more information, see <a href="#">Viewing a Job's Run History</a> .
	A menu that allows the Administrator to: <ul style="list-style-type: none"> <li>• Edit a job. For more information, see <a href="#">Editing a Job</a>.</li> <li>• Run a job or starting a schedule. For more information, see <a href="#">Running a Job</a>.</li> <li>• Suspend a job. For more information, see <a href="#">Suspending a Job</a>.</li> </ul>
Description	A description of the job.
Applied To	Indicates the scope of the job and whether it is applied to: <ul style="list-style-type: none"> <li>• System</li> <li>• All Applications</li> <li>• Specific Systems</li> </ul>
Last Run	Indicates the date the job was last executed.
Last Run Status	Indicates the status of the last execution of the job. Possible values include: <ul style="list-style-type: none"> <li>• Scheduled: The job is set to schedule.</li> <li>• Running: The job is currently running.</li> <li>• Success: The job was executed successfully.</li> <li>• Failure: The job failed to execute.</li> <li>• Skipped: The job was skipped.</li> </ul>
Next Run	Indicates the next scheduled run of the job.
Job Condition	Indicates the current job status. Possible values include: <ul style="list-style-type: none"> <li>• Active: The Administrator is able to run the job.</li> <li>• Suspended: The Administrator suspended the job's current run.</li> </ul>

An **Information** tab contains the custom properties of a selected job.

## Viewing a Job's Run History

The Administrator is able to view the run history by clicking the name of a job. The Job Run History window contains the following information:

Column	Description
<b>Scheduled Date</b>	Indicates the date of the job schedule.
<b>Scheduled By</b>	Indicates the name of the person who initiated the job run.
<b>Start Time</b>	Indicates the time the start time of the job run.
<b>End Time</b>	Indicates the end time of the job run
<b>Status</b>	Indicates the status of the last execution of the job. Possible values include: <ul style="list-style-type: none"> <li>• Scheduled: The job is set to schedule.</li> <li>• Running: The job is currently running.</li> <li>• Success: The job was executed successfully.</li> <li>• Failure: The job failed to execute.</li> <li>• Skipped: The job was skipped.</li> </ul>
<b>Application Name</b>	If the job was applied to an application, the name of the application is indicated.

The administrator is able to delete a selected job instance or all job instances. You cannot delete a job instance if the job is currently running:



- Click  to delete a particular job instance. You will be prompted to confirm that you want to delete the selected job instance.
- Click **X CLEAR ALL COMPLETED** to delete all job instances. You will be prompted to confirm that you want to delete all of the job instances.

It is important to note that, when you click **X CLEAR ALL COMPLETED**, the system does not stop scheduled job instances. It is not possible to delete a scheduled job instance. You can, however, still delete individual histories for job instances that have completed.

## Creating a Job

One of the reasons for creating a job is that a customer can run a specific job manually.

When determining how often a job runs, consider the fact that running a job too often can impact system performance.

1. On the **Jobs** tab, click **+**.
2. Select the job type you want to act as the template for the job being created. The new job will inherit all configuration and property values from this existing job.
3. Click **Next**.
4. Enter the following information:

Field	Description
Job Name	Enter a unique name for the new job.
Handler	Indicates the Java handler that executes after the completes its run.
Description	Enter a description of the job being created.
Applied To	<p>Specify the scope of the job and whether it is applied to:</p> <ul style="list-style-type: none"><li>• System: If selected, the job will not be dependent on any application and the JobHandler will not receive the application during execution. If prompted, redefine values of the job properties.</li><li>• Specific Applications: If selected, select the applications the job can be applied to. Click <b>Select all</b> to select all of the applications.</li></ul> <p><b>Note:</b> 'Select All' selects of the items in the current list. If a user creates a new application, and wants the job to execute for this new application, open this page and select the new application.</p>

5. Click **Next**.
6. Enter the following information:

Field	Description
Repeatable	Indicate whether the job is a repeatable job.
Schedule By	<p>Indicate the if job is to be executed:</p> <ul style="list-style-type: none"> <li>• Manually</li> <li>• Interval: If selected, enter <ul style="list-style-type: none"> <li>— Interval: Specify the number of minutes that must pass before the job can be repeated. If you configure a job to run every minute on an interval, it does not mean the job will run every minute. It means it will run a minute after the last job finishes.</li> <li>— Indicate the maximum number of attempts the JobInstance will be rescheduled after failing to execute successfully.</li> <li>— Expiration Interval: Indicate the number of minutes the job will run before logging a failure.</li> <li>— Retry Interval: After an instance execution fails, indicate the number of minutes the server should wait to reschedule the job.</li> </ul> </li> <li>• Expression: If selected: <ul style="list-style-type: none"> <li>— Enter the expression that defines the job schedule by specifying when the job will run by the second, minute, hour, day of the week, day of the month, month or any combination of these options. The expression supports "Cron Expression" syntax.</li> </ul> <p>Once the expression has been entered, the system indicates the job's next run date and time.</p> <ul style="list-style-type: none"> <li>— Indicate the maximum number of attempts the JobInstance will be rescheduled after failing to execute successfully.</li> <li>— Expiration Interval: Indicate the number of minutes the job will run before logging a failure.</li> <li>— Retry Interval: After an instance execution fails, indicate the number of minutes the server should wait to reschedule the job.</li> </ul> </li> </ul>

7. Click Next.
8. Review the information you have entered. When satisfied that the information is correct, click Finish.

The job now appears in the table on the **Jobs** tab.

## Editing a Job

If you schedule a job, and then subsequently change the job's duration, you must schedule the job again.

If you modify a job to run on an interval and set the duration to '0', the system will treat the job as if it has been set to manual.

When determining how often a job runs, consider the fact that running a job too often can impact system performance.



1. On the **Jobs** tab, click  for the job being edited and select **Edit Job**.
2. Edit the following information:

Field	Description
Description	Enter a description of the job being created.
Schedule By	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>• Manually</li> <li>• Interval: If selected, enter <ul style="list-style-type: none"> <li>— Interval: Specify the number of minutes that must pass before the job can be repeated. If you configure a job to run every minute on an interval, it does not mean the job will run every minute. It means it will run a minute after the last job finishes.</li> <li>— Indicate the maximum number of attempts the JobInstance will be rescheduled after failing to execute successfully.</li> <li>— Expiration Interval: Indicate the number of minutes the job will run before logging a failure.</li> <li>— Retry Interval: After an instance execution fails, indicate the number of minutes the server should wait to reschedule the job.</li> </ul> </li> <li>• Expression: Enter the expression that defines the job schedule by specifying when the job will run by the second, minute, hour, day of the week, day of the month, month or any combination of these options. The expression supports "Cron Expression" syntax.</li> </ul> <p>Once the expression has been entered, the system indicates the job's next run date and time.</p> <p><b>Note:</b> If you specify a cron hour of expression, it is based on UTC time. When a search is executed, the date-time shown in the result screen shows is the local time (browser/client local time). The tooltip, however, shows the recorded/ingested date-time value.</p> <p>The column filter also allows you to enter time in your local time.</p>
Max Attempts	Indicate the maximum number of attempts the JobInstance will be rescheduled after failing to execute successfully.
Expiration Interval	Indicate the number of minutes the job will run before logging a failure.

Field	Description
Retry Interval	After an instance execution fails, indicate the number of minutes the server should wait to reschedule the job.
Apply To	Specify the scope of the job and whether it is applied to: <ul style="list-style-type: none"> <li>System: If selected, redefine values of the job properties.</li> <li>Specific Applications: If selected, select the applications the job can be applied to. Click Select all to select all of the applications.</li> </ul>

3. Click Save.

## Running a Job



1. On the **Jobs** tab, click for the job being executed and select one of the following:
  - If the Schedule By setting is 'Manual', click Run.
  - If the Schedule By setting is 'Interval' or 'Expression', click Start Schedule.

## Suspending a Job Schedule

When a job schedule is suspended, the job framework removes all scheduled job instances.



1. On the **Jobs** tab, click for the job that is running and select Suspend. The Job Condition will be 'Suspended'. Click Resume to continue the job's schedule.

## Deleting a Job

You are able to delete a job definition when it is in a suspended state. While on the **Administration > Jobs** tab, click to delete a particular job definition. When prompted to confirm the deletion, click **DELETE**.

The job definition no longer appears in the **Administration > Jobs** tab.

## Reviewing the Logging Information for Jobs

When a job is started, the following information may be available in the first log entry for the job instance:

Field	Description
<b>name</b>	Indicates the name of the job. This field is always present.
<b>handler</b>	Indicates the handler for the job. This field is always present. For jobs that were cloned, this field helps identify what the job does.
<b>properties</b>	Indicates the properties that the job execution used. Job parameters are shown in curly brackets (for example: type =aip, retentionPolicy=Policy A).
<b>scheduled by</b>	Indicates the user who ran job. This field is not shown for scheduled jobs.
<b>application</b>	Indicates the application that the job instance was scoped to.
<b>attempt</b>	Indicates the attempt for running the job. This field is always present. Value is always one unless the job has been configured for auto retry on failure. Does not increment when doing a manual retry.

The following is an example of the log entry after the execution of the Remove Policy job:

[Jan 3, 2018 4:48:17 PM]: INFO

```
Starting job: name=Remove Policy, handler=RemovePolicyJob, properties={type=, retentionPolicyName=Policy A}, application=PhoneCallsGranular, scheduled by sue@iacustomer.com, scheduled at 2018-01-03T21:48:17.804Z, attempt=1
```

## Managing the Log History

For any given job, it is possible to click on the job to view the history of the job. If you want to clear the job history, click **Clear All Completed**. In previous iterations of InfoArchive, this menu was **Delete All** and would have also stopped the schedule for jobs. Use the suspend menu from the job definition list to stop the job schedule.

**Note:** From the listing of the job information, it is possible to see upcoming scheduled jobs. Scheduled jobs will show that it was Scheduled by SYSTEM, whereas jobs that were run manually will indicate the user who initiated the job.

## Troubleshooting Issues with Jobs

<b>Why is the option to download logs for jobs not provided?</b>	Even if the job is scoped to an application, the logs for jobs are stored in the System application. If the system application is not installed, the option is not provided. If this does happen, those logs can be accessed on the machine in which IA Server resides under the logs > iaserver > context-shifting.
<b>Why are the Generate Purge List job or Clean Up Purge Candidate List and Applications job failing?</b>	The Generate Purge List job and Clean Up Purge Candidate List and Applications job require that the System application be installed.
<b>Why is the Dispose Purge Candidate List job failing?</b>	The Dispose Purge Candidate List job is now application-scoped and requires at least one application to be configured for it. If the Dispose Purge Candidate List was scoped to an application and the application is disposed, the Dispose Purge Candidate List job will not run unless it is associated to another application.
<b>Why is my application still being shown after running the Dispose Purge Candidate List job twice?</b>	<p>The process for disposing an application has changed.</p> <p>Instead of running the Disposition job twice, the Clean Up Purge Candidate List and Applications' job is responsible for finishing disposition after the Clean job is run.</p> <p>A second reason is that application had items under hold or longer retention and marking the content in the application was not done.</p>

<b>Why does the Generate Purge List job fail to run and issue an exception error?</b>	<p>This issue occurs if a large operation is being performed (for example, a hold is being applied to a large set of records).</p> <p>If you receive the exception, try running the Generate Purge List job after the large operation is complete.</p> <p>It is recommended that, when doing operations that will affect a large number of records under retention or hold, to do this during off-times to avoid contention for locking objects. If the apply hold operation fails due to a LOCK_NOT_GRANTED exception, try the operation later.</p>
<b>Why do I receive an error message when I click Retry to rerun a job?</b>	Refresh your browser to see if the job is already running.

## Managing User Accounts and Permissions

### Managing Groups

The **Groups** tab allows the Developer and Administrator to administer which groups can access specific InfoArchive functionality. For example, you can permit a user in the Administrator group to perform a compliance task, such as creating a retention policy.



Click to learn what actions can be performed by each user role.

A check mark specifies whether a group can perform the actions of a specific user role.

A drop-down list allows the Developer or Administrator to toggle between the following:

- Show all groups
- Show only groups with assigned roles
- Show only groups without assigned roles
- Show only obsolete groups with assigned roles

### Managing Permissions

The **Permissions** tab allows the Developer or Administrator to control which groups can access specific applications. You can choose whether to allow access by application or by group. You can display all groups or only groups that can access an application.

The permissions model in InfoArchive has two parts. A user is a member of one or more groups. The groups are mapped with roles, which determine which actions the user can perform on items in the archive.

Additional permissions can be associated with applications, to restrict access to specific groups. By default, if there are no groups associated with an application, all users have the ability to perform actions allowed by their role permissions. The **Permissions** tab shows which groups have access to an application.

## Restricting Access to an Application or Search

You might want to restrict access to an application or search to protect sensitive data. For example, you might want to restrict access to the Audit application because audits often provide information about applications that end users should not be able to access. You can restrict access to an application and all of its searches, or you can restrict access to just a particular search.



**Caution:** If you restrict an application's access to a group that you are not a member of, then you will lose all access to the application. If you add a group restriction to an application, you should ensure that you are a member of at least one of the groups that has access to the application.

If a user cannot access an application, check if group limitations are specified for that application. You can use IAWA or the REST API to check which groups a user is a member of.

It is recommended that you limit the groups that can access the Audit and Reports sample applications.

If you do not want auditors to execute searches in certain applications, restrict the groups for those applications to only specific groups.

### To restrict access to an application and its searches:

1. On the **Administration > Permissions** tab, in the three drop-down lists, select **By Application**, the name of the application, and **Show all groups**.
2. In the **Restrict Access To** column, select the groups that you want to be able to access the audit application and its available searches.
3. Click **SAVE**.

### To restrict access to a search:

1. On the **Applications** tab, select the application.
2. Edit the search that you want to restrict access to.
3. On the **Permissions** tab, in the **Restrict Access To** column, select the groups that you want to have access to the search.
4. Click **SAVE**.

# Performing a Byte Count on Application Data

InfoArchive includes a standalone Java tool to calculate file size and character count for tables, XML files or SIP.zip files in an application. The tool automatically chooses a unit (bytes, KB, MB, etc.) in which the smallest file has a value greater than 1. All files have the same unit for easy comparison. Formatting data in a file is not included in the final calculation.

Access the tool in the <INFOARCHIVE\_ROOT>\xDB\lib\core\cassandra\metrics-core-3.1.0.jar directory.

The following parameters are required to perform a byte count:

Parameter	Description
Use one of the following: <ul style="list-style-type: none"> <li>• <i>-xml</i></li> <li>• <i>-sip</i></li> <li>• <i>-table</i></li> </ul>	The <i>-xml</i> parameter is used to calculate the byte size of either an XML file or table.  The <i>-sip</i> parameter is used to calculate the byte size of a SIP.zip file.  The <i>-table</i> parameter is used to calculate the byte size of the XML and the size of the attached content.  You must enter one of the above parameters.
<location>	You must specify a directory or file that the byte count is performed on.
<i>-csv</i> <csv-location>	While it is not mandatory, you can generate a report that contains the information returned in a byte count. The report is in .CSV format. The <i>-csv</i> allows you to specify the name of the report.

## Usage Examples

To perform a byte count on a table named `BASEBALL-MASTER-01.xml` that is stored in the Baseball application, the following command is used:

```
C:\projects\IA\infoarchive>tools\metrics.bat -xml applications/
Baseball/tables/BASEBALL-MASTER-01.xml
```

When performing a byte count using the *-xml* parameter, the following information is returned:

Column	Description
File size	Indicates the compiled data size of the table or XML file.
Data size	Indicates the character count of the table or XML file.

To perform a byte count on all of the tables stored in the Baseball application, the following command is used:

```
C:\projects\IA\infoarchive>tools\metrics.bat -xml applications/
Baseball/tables
```

To perform a byte count on all of the tables stored in the Baseball application and generate a report called `baseball.csv`, the following command is used:

```
C:\projects\IA\infoarchive>tools\metrics.bat -xml applications/
Baseball/tables -csv baseball.csv
```

To perform a byte count on the SIP.zip files stored in the Trades application, the following command is used:

```
C:\projects\IA\infoarchive>tools\metrics.bat -sip applications/
Trades/sips
```

When performing a byte count using the `-sip` parameter, the following information is returned:

Column	Description
File size	Indicates the size of the <code>eas.pdi.xml</code> file.
Data size	Indicates the character count of the <code>eas.pdi.xml</code> file.
Content size	Indicates the byte count of any additional files (for example, images) stored in the <code>sip.zip</code> file.
Sip size	Indicates the compiled data size of the <code>sip.zip</code> file.

## Conducting Periodic Heartbeat Checks on InfoArchive

For customers who operate InfoArchive in a lights out model, which allows the administrator to monitor servers by remote, InfoArchive allows you to conduct periodic heartbeat checks to ensure that the xDB, InfoArchive and web application servers are running. The heartbeat check allows you to check a few vitals and parameters.

The following servers allow you to conduct these types of checks, each with a default `/health` endpoint:

Server	Health Endpoint
xDB	<code>http(s)://&lt;XDB_HOST_NAME&gt;:2915/health</code>
IA Server	<code>http(s)://&lt;IAS_HOST_NAME&gt;:8765/health</code>
IA Web App	<code>http(s)://&lt;IAWA_HOST_NAME&gt;:8080/health</code>

These endpoints can be used to ensure that the servers are up and running. In this initial implementation, HTTP 200 is returned if the server is, in fact, up. This functionality can be used with automated tools to check the status and availability of the related services.



# Chapter 6

---

## Compliance – General Concepts

Compliance in InfoArchive consists of:

- Retention policies,
- Holds,
- Collections and legal matters, and
- Disposition (also referred to as purges).

Compliance is an important InfoArchive feature because it not only allows the customer to retain data for a designated amount of time, but it also allows you to indicate when data can be disposed or purged. Disposition is vital because data stored in a repository costs the customer money and, if the data is to be retained indefinitely, you will continue to incur the costs to store this data. It is in your company's best interests to retain sensitive data only as long as the data is required. After that date, it should be disposed.

Once an application is Active (for example, no longer has the In-Test status), the only way a company can get rid of stored data is through disposition process.

Data can be retained with the following methods:

- Retention managers can use InfoArchive's interface to apply retention directly to applications, AIPs and tables.
- Retention can be applied automatically to content during ingestion (AIP).
- Two jobs can be configured to apply retention to records in either tables or packages (AIPs). Customers are able to apply retention using rules. Refer to [Defining How a Retention Policy is Applied to Records in an AIP or Table](#) for more information.

Holds can be applied to prevent disposition. Data is never destroyed (disposed) until approval is given. There is a minor caveat that, if retention is not applied, it is possible that, if the application is in test, that the data can be destroyed. This is meant for cleaning up test systems. All retention and holds must be removed before deleting the application data (for applications in test mode). The retention manager can remove retention from all of the packages in an application to facilitate this. This operation does not affect individual records that may have holds or retention applied. If retention was applied to records, you have the option of using the Remove Policy job. To remove all holds, the retention manager needs to delete each of the hold sets. If the collection feature is being used, the E-Discovery administrator needs to remove the collection from the matter.

## Compliance-Related Roles

Retention Managers perform the following compliance-related duties:

- Create retention policies and holds.
- Apply and remove retention policies.
- Apply and remove holds.
- Approve or reject a purge candidate list (used during disposition).

E-Discovery Administrators perform the following compliance-related duties:

- Create collections and legal matters.
- Apply or remove collections and legal matters.

## The Retention Lifecycle

The following steps outline the retention lifecycle:

1. A retention policy is applied to data.
2. Wait for time to elapse or an event to occur.
3. Items are bundled for disposition approval.
4. The Retention Manager approves the disposition of the purge candidate list.
5. Data is disposed.

These steps are elaborated in the following sections.

## What is a Retention Policy?

If data is ingested into the system, and a retention policy is not applied, the items will remain in the archive indefinitely. Retention policies dictate when the items in the repository should be disposed. Retention management supports a controlled disposition process for the Retention Manager.

A retention policy specifies the rules for how long to retain the data. It is a set of guidelines that describes what data will be archived, how long it will be kept, and other factors concerning the retention of the data. Its objectives are to keep important information for future use or reference, to organize information so it can be searched and retrieved at a later date and to dispose of information that is no longer needed. Data can only be disposed when the retention period expires. Retention policies can also disable disposition even after the retention period has expired.

There are four different retention policies:

Retention Policy Type	Description
Fixed Date	<p>A holding will be retained until a particular date. This means that there is no aging and that all records associated to this policy will be eligible for disposition on the date specified in the policy. This policy is useful if the disposition date is known in advance.</p> <p>For instance, you want to retain records until January 1, 2020. Records that have the retention policy applied after January 1, 2020 are immediately eligible for disposition.</p> <p><b>Tip:</b> All records using the same fixed date retention policy can share the same retained set. Individual retention should be set for false.</p>
Duration	<p>Specifies a length of time to keep a record from a date based on the record. For example, you opt to keep all transactions for 90 days from their creation date.</p> <p>The duration policy is good for when records age based on an attribute associated to the record (for example, creation date).</p> <p>For example, all trades entering the system must be kept for 120 days from their creation date. A duration policy of 120 days is applied to the transactions and the creation date is used as the base date. After 120 days, the trades will be eligible for disposition.</p> <p><b>Tip:</b> If all trades are using the same policy, you can group a day's worth of trades into one retained set and not set the individual flag. This would mean that the date you are using will be stored on the retained set and not on the individual policy applications. This would make for more efficient updates if you need to change the duration of the policy (for example, to 150 days).</p> <p> <b>Caution:</b> If applying a retention policy directly to tables or packages, customers using Isilon should never reduce the duration of a retention policy once it has been set. Doing so causes the Requalification job to fail. The issue occurs because Isilon ensures that data is kept as long as the originally stated date.</p> <p>To resolve the issue, change the retention policy back to its original duration (increasing the duration or moving the retention to a later date).</p> <p>If you are applying a duration retention policy, all of the records should include the Date field. Otherwise, only a subset of the records may be protected. You will then have to refer to the logs to see which records are protected. The intent is for a duration retention policy, the system needs to know the base date to start aging from. If the base date is not known for records, consider using an event-based retention policy instead.</p> <p> <b>Caution:</b> Customers using ECS cannot change retention policies once they have been applied, or apply additional policies to packages.</p>

Retention Policy Type	Description
Event	<p>Specify a length of time to keep records from when an event has happened. All records are grouped together using a context so that, when the event is triggered, all records using that context, regardless of their policy, will start aging.</p> <p>Event policies are good when a record will be retained until a specific condition or conditions are met. For instance, you want to retain records until the date the employee leaves the company. The context that you would use to group all the employees records would be their employee IDs.</p> <p>For example, Bill Steele (Customer ID BILLS01) cancels all his policies with the company. The company wants to keep all the policy documents generated for 5 years from the date Bill cancels his policy and all correspondence Bill had with the company for 3 years. Bill cancels his policy on December 15, 2007. The event fulfilled with the date of December 15, 2007. All records belonging to Bill have a context of BILLS01. The insurance policy records have a 5 year event policy applied to them and all the correspondence records have a 3 year event policy applied to them. When the event is triggered for Bill's records, the insurance policy documents will be eligible for disposition on December 15, 2012 and the correspondence records will be eligible for disposition on December 15, 2010.</p> <p>Event policies can only be applied using the jobs.</p>
Mixed Mode	<p>This policy is a combination of the Duration and Event policy types. It means that the records will age like a duration policy unless an event happens. If the event happens, then the record will age based only on the event date and will qualify for disposition even though the duration qualification date is shorter.</p> <p>If the event date would cause a qualification date that is later than the duration-based date, the duration-based date is used.</p> <p>For example, there is a 10-year retention policy, and the aging started in January 2001. If the event is set to be July 1, 2018, the qualification date would remain as January 2010, and not change to be July 1, 2018 (assuming no aging was required after the event was fulfilled). If, however, you have a 10-year retention policy, and the aging started today, but the event is fulfilled a week later, the record is immediately eligible for disposition.</p> <p>Mixed mode policies can only be applied using the jobs.</p> <p>If you are applying a mixed retention policy, all of the records should include the Date field.</p> <p>If rules are used to apply the policy, and the base date is not specified, retention is applied but the base date is not set. Aging commences only if the event is set.</p> <p>In the case of the original Apply Policy To Records job, if the retention policy is mixed mode and a date field is not specified, the retention policy will not be applied to anything. If the date field is specified, but the record specifies a null date, then the record will never begin aging.</p>

At the end of the retention period, all data will be destroyed if the policy's disposition strategy is set to Destroy All. The Retention Manager is able to Disable Disposition for a retention policy, which means that data governed by the policy will not be added to a purge candidate list or disposed. Even if the Disable Disposition is not checked, disposition will not automatically occur until the Retention Manager approves a purge candidate list.

## What is a Retention Set?

A retention set is a logical container that references data under retention, including:

- Whether or not items in the set are aging together
- The type of items in set (for example, application, package, table or record).

When a retention policy is applied to data, that data is held in a retention set.

Retention sets are created and managed by the Retention Manager.

To easily view an application's retention sets in IA Web App, select an application and access the **Retention Sets** tab.

## What is a Hold?

A hold provides the ability to block deletion or disposition either temporarily or indefinitely. InfoArchive uses holds to prevent data from being deleted, even if the retention policy allows for the data to be included in a purge candidate list. Holds can be applied to applications, archival information packages (AIPs)/tables, or archival information units (AIUs)/records, depending on whether you are working with a SIP- or table-based archive. Holds are created and managed by the Retention Manager.

There are two types of holds:

- Legal: The assumption is that the hold will eventually be removed.
- Permanent: The assumption is that the hold will never be removed.

## What is a Hold Set?

Similar to a retention set, a hold set is a logical container that references data with a hold applied against it. A hold set is created when a hold is applied to one or more items.

To easily view an application's hold sets in IA Web App, select an application and access the **Hold Sets** tab.

## Granularity

Retention policies and holds can be applied to:

- Applications
- Archival information packages (AIPs) or tables
- Archival information units (AIUs) or table rows are identified as records

Retention policies can be applied to applications via the Application Info screen. Retention policies can be applied to packages and tables from the package and table screen. Retention policies can be applied to records via the jobs. Furthermore, retention can be applied to items via the IAShell and REST API.

## What is Disposition?

Disposition refers to the controlled process of what to do when the required aging has been met by the retention policies. Currently, the only disposition action is to destroy the item. Disposition is only done after the required approvals have been given.

Data under retention will eventually qualify for disposition. Qualification is the process that determines when an item qualifies for disposition. It is possible that a qualification date cannot be calculated. For example, for event-based retention, if the event has not been fulfilled, a qualification date will not be set. To qualify for disposition:

- The retention period must have expired or the event must have been fulfilled, as required by the retention policy.
- There must be no holds applied against the data.

To qualify for disposition, the retention period must have expired. Holds effectively block disposition until removed.

Once data qualifies for disposition, data can be disposed. For the purposes of this guide, disposition, purging, and data destruction are synonymous.

The customer must execute jobs to determine which data qualifies for disposition and is added to a purge candidate list. The two main jobs include:

- Generate Purge Candidate List: This job creates purge candidate lists for items that are eligible for disposition, meaning that the retention period has expired and the items are not under hold. By default, this job runs on a weekly schedule.

It is important to ensure that the Dispose Purge Candidate List job runs on a similar schedule. If the Generate Purge Candidate List job runs before the Dispose Purge Candidate List job, purge candidate lists with the status of Approved and Under Review will be marked as Cancelled.

- Dispose Purge Candidate List: This job executes the disposition of approved purge candidate lists. By default, this job runs on a weekly schedule.

It is important to have this job scheduled. Once an application is active, this job is the only way to remove content from archive.

Before data is disposed, its disposal must be approved by the Retention Manager.

The following outlines the disposition lifecycle:

1. Data qualifies for disposition.
  2. The Generate Purge Candidate List job is executed. The purge candidate list will have the status of Under Review.
  3. At this point, the Retention Manager can approve or reject the purge candidate list.
  4. The Dispose Purge Candidate List is executed. All approved purge candidate lists are disposed.
- After disposition runs, users will no longer be able to view or download the content.

## What is a Purge Candidate List?

During the disposition process, purge candidate lists are generated and used to track approvals.

A purge candidate list contains:

- Data that qualified for disposition (when the list was generated), but may no longer. Even though the item is under hold, it still technically qualifies for disposition, it just will not be disposed until all holds are removed.
- Information related to data in an application that has been disposed.
- Information related to data in an application that was set for disposition but was cancelled.

Purge candidate lists are always associated with an application. Lists are created per application and type of object.

To easily view an application's purge candidate lists in IA Web App, select an application and access the **Purge Lists** tab.

Once a purge candidate list is generated, the Retention Manager reviews the contents and chooses to ignore, reject or approve the list:

- If the purge candidate list is ignored, the items in the purge candidate list will be added to the new list the next time the Purge Candidate List Generation job runs.
- If a purge candidate list is rejected, items in the purge list will not be eligible for inclusion in new lists until the state of the rejected list is changed.
- If the purge candidate list is approved, items in the list will be disposed. If a hold is subsequently placed on items in the list or a longer retention policy applied to items in the list, those items will not be disposed, even though approval was given.

## What is Granular Disposition?

Granular disposition, also known as record-based retention, allows the customer to specify retention on individual records so that records can be disposed independently. This provides more customer control, as they can choose to use package or table retention as an alternative or can decide to use both.

When applying retention to records, search criteria must identify what to protect as well as which fields to show when viewing the record. Customers can either use the Apply Policy to Record job or the Apply Retention Rule to Records job.

A recommendation is to use the same search set and search when applying retention. If different searches or search sets are used, a different purge list will be created for each search defined, as well as which retention policy was applied.

If a table or a package is up for disposition, and some of their records cannot be disposed, the table or package will be re-factored. In the case of a package, the package will be modified.

Customers are encouraged to apply retention to the records or apply retention to the packages (using the holding configuration or the document class).

If the purge list for the package is approved, disposition can be done on the records even if the purge list for the records was not approved.

**Tip:** It is recommended that you put holds on the records that should not be disposed and then approve the purge lists.

## Timing of Applying Holds when Using Granular Retention

Retention can be applied to records using one of the jobs and ensure that the records are eligible for disposition. You use one of the following jobs, either the:

- [Apply Retention Policy to Records job](#)
- [Apply Retention Rule to Records job](#)

If a hold is applied before the Generate Purge Candidate List job runs, the records will not be included the purge candidate list.

If you remove the hold from the records and then run the Generate Purge Candidate List, the old purge candidate list will be cancelled and the items will appear in the new purge list.

If you:

1. Apply the hold to the records again.
2. Approve the purge candidate list.
3. Run the Disposition job.

When viewing the purge candidate list, you will see that the held items are still included in the list. When viewing the details of the purge candidate list in the side panel, however, you will see some of the items were skipped.

**Note:** It does not matter if the records were table rows or AIUs, granular disposition works the same for table and SIP archives.

Essentially, if the items are eligible for disposition before a hold is applied, the item is included in a purge candidate list. Applying a hold keeps the item in the purge candidate list (causing it to be skipped if the purge list is disposed).

# The Retention Lifecycle

The following steps outline the retention lifecycle:

1. A retention policy is applied to data.
2. Wait for time to elapse or an event to occur.
3. Items are bundled for disposition approval.
4. The Retention Manager approves the disposition of the purge candidate list.
5. Data is disposed.



# Chapter 7

## Compliance Related Tasks

### Using the Retention Policies Tab

The **Retention Policies** tab allows the Retention Manager to:

- View and edit the details of a retention policy.
- Create and delete a retention policy.

Each retention policy is displayed in a table that contains the following information:

Column	Description
Policy Name	Indicates the name of the retention policy.
Policy Category	Indicates the category associated with the policy.
Aging Strategy	Indicate the aging strategy (Fixed Date, Duration, Event, and Mixed).
Approved Date	Indicates the name of the person or organization that approved the policy (optional).
Policy Approver	Indicates the name of the person or organization that approved the policy (optional).
In Use	Indicates whether the retention policy has been referenced or applied.
Last Modified Date	Indicates when the retention policy was last modified. This value is shown in the user's local time zone.

A side panel contains additional properties of a selected retention policy:

Attribute	Description
Description	A description of the retention policy (optional)
Created By	The user that created the retention policy. If this retention policy was imported from configuration, this value will be the user that imported the configuration.
Notes	Any additional information provided that can aid the Retention Manager (optional).
Created Date	The date the retention policy was created (only date is shown).
Modified By	The user that last modified the retention policy.
Custom Attributes	A list of custom attributes specified to further define the application. These values can only be set when importing a retention policy from configuration.

# Creating a Retention Policy

1. On the **Compliance > Retention Policies** tab, click +.
2. Enter the following information:

Field	Description
Policy Name	Enter a unique name for the policy.
Description	Enter a description for the policy.
Policy Category	Enter or select a policy category (for example, you may want to have a category for policies applicable to e-mail messages and another category for policies applicable to voicemail messages).
Aging Strategy	<p>Select the retention type for the policy being created. If you select:</p> <ul style="list-style-type: none"> <li>• <b>Fixed Date</b>, enter the date after which the item can be disposed.</li> <li>• <b>Duration</b>, indicate how long the policy will retain the item. Specify the duration in years, months, weeks, or days. Annual cutoff can be set if you want disposition to occur at the end of the company's fiscal year). If you select an annual cutoff for the policy, specify the cutoff day and month.</li> </ul> <p> <b>Caution:</b> Customers using Isilon should never reduce the duration of a retention policy once it has been set. Doing so causes the Requalification job to fail. The issue occurs because Isilon ensures that data is kept as long as the originally stated date.</p> <p>To resolve the issue, change the retention policy back to its original duration (increasing the duration or moving the retention to a later date).</p> <ul style="list-style-type: none"> <li>• <b>Event</b>, select or enter the condition that has to be met before a holding can be disposed. You can also indicate annual cutoff. If you select an annual cutoff, specify the cutoff day and month. Refer to <a href="#">Event-Based Retention</a> for further information.</li> <li>• <b>Mixed</b>, indicate how long the policy will retain the item if the event does not occur. Specify the duration years, months, weeks or days. Select or enter the condition that may be met and then indicate how any additional time after the event has happened for how the policy will retain the items. You can also indicate a cutoff date for either if the event does not happen or if the event happens. If you select an annual cutoff, specify the cutoff day and month.</li> </ul> <p>A date must be within the following range: 1000-01-01 –2999-12-31.</p>
Approved Date	Enter the date that the retention policy was approved for use (optional). There is no business logic associated with this date.

Field	Description
Policy Approver	Enter the name of the person or organization that approved the policy (optional).
Notes	Enter any relevant policy information you want to communicate.
Disable Disposition	Click to ensure that items protected by the retention policy will not appear on a purge list or be disposed.

3. Click **Create**.

## Applying Retention

Multiple retention policies can be applied to the same item. For more information, see [Applying Multiple Retention Policies](#).

If a retention policy or hold is applied, in order to delete entire application (including data), the policy and hold need to be removed from any item in the application, which includes archival information packages (AIPs), tables or records.

## Choosing Where to Apply Retention and the Consequences

The following table gives an overview of various approaches that can be taken on how to apply retention.

Approach	Pros	Cons
Apply to application	Simple and only one item is ever up for disposition	Cannot use hardware protection
Apply to package/table	Simple and each table or package can use hardware protection	May not meet business requirements for retention Back up of table done at schema (size of back up)
Apply to records	Most flexible	Most overhead, each record could be aging independently

**Apply to application:** If everything in the application needs to be disposed together, consider applying retention to the application. This is the simplest approach, which is appropriate for application decommissioning. If any items in the application (for example, AIP or table) are put under hold, however, disposition of the application will not proceed.

**Apply to package:** For SIP-based archiving, retention can be applied to the package when it is ingested. A default retention policy can be defined on the application, on the holding, or on the ingested package (through the retention class). Only one retention policy would be applied, precedence is package > holding > application. If nothing is defined, no retention policy is defined.

**Apply to table:** For table based archiving, retention can be applied to the table but not during ingestion. Retention can be applied to individual tables using the IA web application.

**Apply to records:** Retention can also be applied to records for both application types. For SIP-based archiving, retention can be applied to AIUs during ingestion via either using rules or via a duration retention policy. The holding wizard can be used to specify a duration retention policy and a search.

For table applications, it is not possible to apply retention to records during ingestion.

For both types, there are two jobs that can be used to apply retention to records: Apply Retention Policy to Records and Apply Retention Rule to Records.

## Applying Multiple Retention Policies

It is best to avoid applying multiple retention policies (single source). If multiple policies have been applied, the longest retention policy is used. For example, if a duration retention policy is applied for 5 years from now, and a fixed retention policy was applied to retain until 2050, the item will not be eligible for disposition until 2050.

If a shorter retention policy is applied, a new date will not be pushed to the hardware (Isilon). For Isilon storage, dates can only be pushed further into the future.

The ability to apply multiple policies is not supported by ECS.

InfoArchive also allows you to apply retention to records. When viewing a record in the search results, the longest retention policy and source are shown. Sources can be the application, table/package or record.

One or more retention policies and holds can be applied to AIPs, AIUs, applications and rows within an application.

A table or package will not be completely disposed if one of its children cannot be disposed.

The following outlines the effect of applying multiple retention policies from different sources:

- It is recommended that you pick one retention strategy instead of mixing strategies.
- If retention is applied to a package (or table) and records, the following is the behavior:
  - Purge lists are created for both the package and table.
- If a package is approved, the approval of purge lists for the records is ignored:
  - If records either have longer retention or holds, the package is re-factored.
  - If the purge list for the package is approved, eligible records will be disposed even if the purge lists for those records are not approved or rejected.
- If specific records need to be kept, apply holds before approving the purge list for the package instead of rejecting the purge list for records.
- If a package no longer has any records after disposition, the package is marked for purge.
- Tables are no longer destroyed if retention is applied directly, only the records. If the table is no longer needed, consider applying retention to the application once everything is removed to destroy the application.

## Verifying that an Item is Protected

Depending on the type of item, the procedure is different:

- Application
- Table
- Package
- Record

## Verifying Using the Application Info Tab

To verify if an application is protected, **SELECT** the desired application from the **Applications** tab and then navigate to the **Application Info** tab. From here, you can see all of the retention policies and holds. The Retention Manager can remove any [retention policies](#) and [holds](#) from the **Application Info** tab.

## Verifying Using the Application's Tables Tab

If the application is a table archive, the **Tables** tab provides an overview on whether the table is under retention or hold.

The screenshot shows the 'Tables' tab of the Retention Manager interface for the 'Baseball' application. The left panel displays a list of tables: TEAMSHALF, TEAMSFRANCHISES, TEAMS, and SERIESPOST. Each table row includes columns for Table Name, Retention, Hold, Records, and Last Modified Date. The 'TEAMSFRANCHISES' table is currently selected, highlighted with a blue background. The right panel shows the detailed schema for the 'TEAMSFRANCHISES' table, listing columns: ACTIVE, FRANCHID, FRANCHNAME, and NAASSOC, along with their data types (VARCHAR) and various properties like Encrypted and Indexed.

Table Name	Retention	Hold	Records	Last Modified Date
TEAMSHALF	✓	✓	52	Feb 9, 2017 1:45:43 PM
<b>TEAMSFRANCHISES</b>	✓	✓	120	Feb 9, 2017 1:45:40 PM
TEAMS	✓	✓	2775	Feb 9, 2017 1:45:37 PM
SERIESPOST	✓	✓	298	Feb 9, 2017 1:45:35 PM

To view if retention is applied directly to a specific table, click the table name. The following messages may be displayed:

- **No Retentions found** or
- **No Holds found**,

These messages indicate that retention is not directly applied to the table.

**Note:** If there are individual records within a table are under retention, the table itself is not considered under retention.

The Retention Manager can apply a policy or hold to a selected table from the Tables tab. The retention manager can also remove retention from the table.

## Verifying Using the Application's Packages Tab

If the application is for a SIP archive, the **Packages** tab specifies whether a retention policy, hold or both have been applied to the application's packages.

Type	Name	Phase	Holding	Reception Date	Online	Retention	Hold	Records
Completed	PhoneCalls-CC-1000010-1	PhoneCalls	Feb 9, 2017 ...	✓	✓			10
Completed	PhoneCalls-CC-1000010-1	PhoneCalls	Feb 9, 2017 ...	✓	✓			3
Completed	PhoneCalls-CC-1000010-1	PhoneCalls	Feb 9, 2017 ...	✓	✓			3
Completed	PhoneCalls-CC-1000010-1	PhoneCalls	Feb 9, 2017 ...	✓	✓			3

**Summary**  
 Name: PhoneCalls-CC-1000010-1  
 AIP ID: 28659d67-ae2f-49ec-9aad-5b75b382506d  
 AIU count: 10  
 CI count: 1  
 PDI file size: 7575  
 PDI character count: 1729  
 CI size: 41123

To view retention directly applied to a specific package, click the package name. The following screen shot shows that the PhoneCalls-policy has been applied to the selected package:

Name	Description	Store	Size	Modified Date
ci.container	Aggregated content file of the package	file_store_01	41136	Feb 9, 2017 1:51:00 PM
ingest.log.gzip	Compressed ingestion log	file_store_01	2197	Feb 9, 2017 1:51:12 PM
pdi.xml.gzip.crypto	Encrypted and compressed PDI XML	file_store_01	928	Feb 9, 2017 1:51:00 PM
receive.log.gzip	Compressed reception log	file_store_01	320	Feb 9, 2017 1:50:58 PM
rendition.csv.gzip	Dataset for analytics	file_store_01	519	Feb 9, 2017 1:51:12 PM
ri.xml	XML table of contents of the package	file_store_01	701	Feb 9, 2017 1:51:00 PM
sip.xml	XML description of the package	file_store_01	580	Feb 9, 2017 1:50:58 PM

Retention Policy Name	Aging Strategy	Base Date	Qualification Date
PhoneCalls-policy	Duration	2010-02-01T00:00+01:00	2010-05-01T00:00+01:00

No Holds found

The items contained in the package are also displayed.

The **No Holds found** message indicates that the hold is inherited from the application.

**Note:** If the individual records within a package are under retention, the package itself is not considered under retention.

The Retention Manager can apply a policy or hold to a selected package from the **Packages** tab. It is also possible to remove the retention policy or hold from the package.

## Verifying Retention in Search Results

To verify if retention on a record, execute a search.

The Details panel on the right side of the screen contains the retention information for the selected record:

The screenshot shows a search results page for 'Debut Date Range Search' with the query 'First Name: Don'. The results table displays 10 out of 132 records, with columns for First Name, Last Name, Debut, and Bats. A red box highlights the 'Details' panel on the right, which provides specific retention details for the selected record (the first one in the list).

First Name	Last Name	Debut	Bats
Don	Aase	1977-07-26T00:00:00.000	R
Don	Arlich	1965-10-02T00:00:00.000	L
Don	August	1988-06-02T00:00:00.000	R
Don	Baylor	1970-09-18T00:00:00.000	R
Don	Bessent	1955-07-17T00:00:00.000	R
Don	Black	1943-04-24T00:00:00.000	R
Don	Blasingame	1955-09-20T00:00:00.000	L

**Details Panel Data:**

- Base Date: Mar 1, 2017 12:00:00 AM
- Projected Disposition Date: Dec 31, 2018 7:00:00 AM
- Retention Policy: Yes
- On Hold: No
- Longest Retention Policy Name: RetainForOneYear
- Longest Retention Source Type: Application
- Longest Retention Event Based: No

The Details panel allows you to:

- See the base date and projected disposition date.
- See if the record is protected by a retention policy.
- See that the record is not under hold.
- See the name of the longest retention policy governing the selected record.
- See the source of where that longest retention policy was applied. The possible values may be application, table, package or record.
- See if the longest retention policy is event-based and, if the event has been set, you can see that value.

## Table Archiving – Application

For table archiving, the upper-level of a set of data is an application. A retention policy or a hold can be applied at the application level or at the row level (individual object within the application).

Once the retention period of an application has expired, everything in application is to be disposed. However, applying a hold policy to items in an application prevents the disposition of the application.

The following table outlines various scenarios regarding the disposition of an application:

Scenario	What Happens
<p>Application with one retention policy applied to it:</p> <ul style="list-style-type: none"> <li>• Duration: 1 day</li> <li>• Base Date: January 1, 2017</li> </ul>	<p>All rows have the same retention policy applied to them and, therefore, have the same base date</p> <p>The retention policy 1 is processed on January 1, 2017. The application and all of its rows are disposed, along with all supporting tables.</p>
<p>Application with two retention policies applied to it.</p> <ul style="list-style-type: none"> <li>• Retention Policy 1: <ul style="list-style-type: none"> <li>— Duration: 1 day</li> <li>— Base Date: January 1, 2017</li> </ul> </li> <li>• Retention Policy 2: <ul style="list-style-type: none"> <li>— Duration: 1 day</li> <li>— Fixed Date: January 2, 2017</li> </ul> </li> </ul>	<p>Only the application is put under retention. The rows are protected, and have the same base date/fixed date, but it is the application that is governed.</p> <p>Retention Policy 1 is processed on January 1, 2017 but the application will not be eligible for inclusion in a purge list.</p> <p>Retention Policy 2 is processed on January 2, 2017. Once the purge list is approved, the application and all of its rows are disposed, along with all supporting tables.</p>
<p>Application with a retention policy applied to it. The application also contains one row with a retention policy applied to it.</p> <ul style="list-style-type: none"> <li>• Retention Policy 1 (applied to application): <ul style="list-style-type: none"> <li>— Duration: 1 day</li> <li>— Base Date: January 1, 2017</li> </ul> </li> <li>• Retention Policy 2 (applied to row): <ul style="list-style-type: none"> <li>— Fixed Date: January 2, 2017</li> </ul> </li> </ul>	<p>On January 1, 2017. nothing is disposed (as the application is either completely disposed or not).</p> <p>Only on January 2, once everything in the application is eligible for disposition, then the application will be disposed.</p>

Scenario	What Happens
<p>Application with a retention policy and a hold applied to it.</p> <ul style="list-style-type: none"> <li>• Retention Policy 1:           <ul style="list-style-type: none"> <li>— Duration: 1 day</li> <li>— Base Date: January 1, 2017</li> </ul> </li> <li>• Hold Policy 1</li> </ul>	<p>The application is not eligible to be put into a purge list on January 1, 2017 because of Hold Policy 1.</p> <p>When Hold Policy 1 is removed from the application, the application is now eligible to be put into a purge list.</p> <p><b>Note:</b> An approval process is required prior to disposition.</p>
<p>Application with a retention policy and a hold applied to one row.</p> <ul style="list-style-type: none"> <li>• Retention Policy 1 (applied to application):           <ul style="list-style-type: none"> <li>— Duration: 1 day</li> <li>— Base Date: January 1, 2017</li> </ul> </li> <li>• Hold Policy 1 (applied to a row):           <ul style="list-style-type: none"> <li>— All rows have Retention Policy 1 with base date of January 1, 2017</li> <li>— One row has Hold Policy 1 applied to it.</li> </ul> </li> </ul>	<p>The application is eligible to be put into a purge list on January 1, 2017. However, when the list is processed, even if approved, no disposition will be done because the hold is applied to an item in the application. This is similar to the case where a record had longer retention than the application.</p>

## SIP Archiving

The main unit of data in a SIP archive is an AIP that contains AIUs. Retention and hold policies can be applied to AIPs and AIUs. Below are some scenarios for SIP based archiving:

Scenario	What Happens
An AIP with a retention policy applied to it: <ul style="list-style-type: none"> <li>• Duration: 1 day</li> <li>• Base Date: January 1, 2017</li> </ul>	The retention policy is processed on January 1, 2017 and the AIP is marked for a purge. At this point, the records will not be returned in search. After the confirmation job is run, after re-running disposition, the package is removed. Nothing happens on January 1, 2017. On January 2, 2017, the AIP is eligible to be included in a purge list.
An AIP with two retention policies applied to it. <ul style="list-style-type: none"> <li>• Retention Policy 1:               <ul style="list-style-type: none"> <li>— Duration: 1 day</li> <li>— Base Date: January 1, 2017</li> </ul> </li> <li>• Retention Policy 2:               <ul style="list-style-type: none"> <li>— Fixed Date January 2, 2017</li> </ul> </li> </ul>	Nothing happens on January 1, 2017. On January 2, 2017, the AIP is eligible to be included in a purge list.
An AIP with a retention policy and hold applied against it. <ul style="list-style-type: none"> <li>• Retention Policy 1 (applied to AIP):               <ul style="list-style-type: none"> <li>— Duration: 1 day</li> <li>— Base Date: January 1, 2017</li> </ul> </li> <li>• Hold Policy 1 (applied to AIP)</li> </ul>	The hold policy prevents the AIP from being included in a purge list. Once the hold policy is removed, the AIP is eligible to be included in a purge list.
AIP that has a retention policy applied to it that contains an AIU with a hold applied to it. <ul style="list-style-type: none"> <li>• Retention Policy 1 (applied to AIP):               <ul style="list-style-type: none"> <li>— Duration: 1 day</li> <li>— Base Date: January 1, 2017</li> </ul> </li> <li>• Hold Policy 1 (applied to AIU)</li> </ul>	The AIP is included in a purge list on January 1, 2017. The AIP will be pruned and only the one AIU under hold will remain in the AIP. A confirmation is not required because the AIP is not fully disposed.

It is possible to apply a retention policy to an AIU. Applying retention to an AIU causes the parent package to be pruned if disposed. The package will be eligible for inclusion in a purge list. Records show up in a purge list (records are put into a different purge list than packages).

If a date was pushed to the storage system that is retention aware, the hardware may not decrease the date. This means that if all retention is removed from a package, even if the customer wants to remove the content, they may have to wait until the retention date is met and use specific tools (specific to the storage system) to remove from the storage once that date is met.

## Record-Based Retention

Retention can be applied directly to records through jobs or via configuration on the holding. For an AIP, the fields shown when expanding the purge list match the search that was used. If different search templates are used, a different purge list is generated. The full disposition of packages and tables are delayed until all governing policies on records have been satisfied (which means it will be pruned).

## Mechanisms for Applying Retention

The following table outlines which retention policy types can be applied using the different mechanisms:

		Mechanisms			
		REST API: AIP, Table Record, AIU or Application	Manually via IA Web App	Via a Job: Record or AIU	Via Ingestion: SIP or Table
Policy Types	Duration	Yes	Yes	Yes	Yes
	Fixed Date	Yes	Yes	Yes	Yes
	Event Based	Yes *	No	Yes	No
	Mixed	Yes	No	Yes	No

\* Applying a event-based or mixed retention policy to applications, tables or AIPs is not supported.

## Using the Applications Tab

The Application Info tab allows the Retention Manager to apply or remove retention policies/holds from an application.

Other users, such as the Administrator or Developer, can access the **Application Info** tab but cannot perform any actions other than reviewing the details of the selected application.

It is important to note that, for AIPs, retention and holds do not go into effect until the AIP is confirmed to be received. It is possible when ingesting to auto-confirm.

## Applying a Retention Policy to an Application

If multiple retention policies are applied to an application, all policies must be honored.

1. Select the application the retention policy is being applied to.
2. On the **Application Info** tab, click **Apply Retention Policy**.
3. Select the retention policy being applied to the application and click **Next**.

4. Review the retention policy details to verify that it is the correct policy to apply to the application. If applicable, enter a new **Base Date**. Click **Next**.
5. Enter the following information:
  - a. **Retention Set Name:** Enter a unique name for the set.
  - b. Enter a **Description** for the policy.
  - c. Click **Next**.
6. Review the information you have entered. When satisfied that the information is correct, click **Finish**.

The retention set applied to the application is listed in the **Retention Sets** tab.

## Applying a Hold to Search Results

The Retention Manager can apply a hold to search results. The hold must first be created in order to apply it to the one or more of the records returned from the search.

When applying a hold to many items in a set of search results, it is recommended that you disable the Disposition job until the hold is applied to avoid a potential race condition.

The application of holds via search results uses a batch mechanism. From the Background Requests tab, the status of the order item indicates the number of batches that were created.

The batch size for operations is controlled by the <INFOARCHIVE\_ROOT>/config/iaserver/application.yml file. Refer to the section in the application.yml file that states:

```
configuration for the batch order item framework
```

**Note:** Changes to these values require a server restart.

1. In the desired table application, execute a search.

The results of the search are displayed on the **Record Search** tab. The number of items in the search results is displayed under the name of the search. The following screen shot shows that 12 results were returned in the search:

The screenshot shows a navigation path: Home > Baseball > Record Search > Player Search with Advance Control > Results. A red arrow points to the 'Select all 12' link in the search results table header.

<input type="checkbox"/>	Last Name	Birth Year
<input type="checkbox"/>	Ortiz	1919
<input type="checkbox"/>	Ortiz	1975
<input type="checkbox"/>	Ortiz	1969

Displaying 1 - 10 of 12      Page 1

- Click the box or boxes beside the items to be placed under the hold. The number of items selected appears in the **Selected # X**. To de-select, simply click the X or click the desired box or boxes. Alternately, to select all of the items returned in the search, click **Select all #**. To deselect the items, click **All # X**.

Once one or all of the items are selected, the **APPLY HOLD** button is displayed.

**Note:** For table applications, the search must specify a table, otherwise, the option to apply the hold is not given.

- Click **APPLY HOLD**.

A three-step wizard allows you to choose a hold and enter the details for the hold set being created.

- In the **Choose Hold** step:

- Select the hold you want to apply to the selected items.

You are able to search for a specific hold. Enter the name of the hold in the **Search for a Hold** field and click **Enter**.

- In the **Type** field, indicate whether you want the hold to be a Legal or Permanent Hold.

- Click **NEXT**.

5. In the **Specify Hold Set** tab:

- a. Enter a name for the hold set that you are creating.
- b. Optionally, enter a description for the hold set.
- c. Click **NEXT**.

If a filter was specified on the search result, if select all is chosen, only the records that matched the filter are protected.

6. Once you have reviewed and verified the information is correct, click **FINISH**.

A message is displayed directing you to check the status of the request in the **Background Requests** tab. Once the hold has been applied, and the Status of the request is complete, the new set will appear on the **Hold Sets** tab for the application.

## Applying Retention to Records

This section illustrates how to apply retention to records. In the following procedure, the Baseball application is used. Because the MASTER database does not contain any date fields, a fixed retention policy is used.

It is possible to apply retention to individual items and not create a set. This is only permitted if the item is marked as aging individually. What this means is that, after applying retention, a retained set will not be created, but you can look at the object and see if the hold is applied.

**Note:** For packages that are being ingested, the system no longer displays a retained set since each package (if package retention is required) ages independently.

1. Create a fixed retention policy with the name `Baseball-FixedDateInPast`. Use a date in the past so it will be available for disposition immediately.
2. As an administrator, create a new job using the `Apply Policy to Records` job as the base.
  - a. Give the job a unique name.
  - b. Change the 'Apply to' for the `BaseBall` application only.
  - c. Set the following properties:

Name	Value	Description
<code>searchName</code>	Search By Player Name	
<code>searchSet</code>	Decrypted First Name	
<code>searchCriteriaFile</code>	<code>&lt;data&gt;&lt;firstName&gt;Don&lt;/firstName&gt;&lt;lastName&gt;Smith&lt;/lastName&gt;&lt;/data&gt;</code>	This could have also been a file location on the server that has this content.
<code>retentionPolicyName</code>	<code>Baseball-FixedDateInPast</code>	

Name	Value	Description
contextType	attribute	Leave as default. This attribute is used to determine how to set the base date but since we used a Fixed Retention policy, doesn't matter.
context		
retentionDateAttribute		
trigger		Trigger only matter for Event and Mixed Retention Policies
triggerCheckAttribute		
triggerCheckValue		
triggerDateAttribute		

- d. Keep the Schedule By as the default, manual.
- 3. Run the new job that you created.
- 4. As a retention manager, view the retention sets.

You might notice that the first name is missing. Because the field was encrypted, the system will not show encrypted fields to the retention manager when viewing the records in either the Retention Sets or the set or the Purge Lists tabs. However, the retention manager will see the fields if a regular search is done (and the developer can control which fields are shown on the search based on the search sets).

- 5. As the administrator, create a new job using the Generate Purge Candidate List job. Set to manual and run it. Wait about a minute for it to finish running.
- 6. As the retention manager, navigate to the **Purge Lists** tab where the list is displayed.

**Note:** The name of the purge list is automatically generated and depends on how many other applications have items ready for disposition.

- 7. Approve the list for disposition.

**Tip:** If only some of the records should not be disposed, run the search and apply a hold to any records that should not be disposed. This can be done as long as the Disposition job has not been run.

- 8. Run the Dispose Purge Candidate List job. It is recommend that you clone the job and set to manual so you can execute it when desired. Normally, disposition is done on a schedule.

When the job has completed, the MASTER table will have those records removed.

## Applying a Retention Policy to an AIP

- 1. Select the AIP the retention policy is being applied to.



2. Click and select **Apply retention**.
3. Select the retention policy you want applied to the application and click **Next**.
4. Review the retention policy details to verify that it is the correct policy to apply to the application. The fields that are displayed depend on the Aging Strategy of the retention policy. For more information, see [Creating a Retention Policy](#).
5. Enter the following information and click **Next**:
  - **Retention Set Name:** Enter a unique name for the policy.
  - Enter a **Description** for the policy.
6. Review the information you have entered. When satisfied that the information is correct, click **Finish**.

The AIP now indicates that there is a retention policy applied to it.

## Using Jobs to Apply Retention

There are two jobs that apply retention to objects in the system:

- The [Apply Retention Policy to Records job](#)
- The [Apply Retention Rule to Records job](#)

### Using the Apply Retention Policy to Records Job

The Apply Retention Policy to Records job applies a retention policy to AIUs or table rows. The job uses a pre-configured search to determine which AIUs or table rows need to have retention applied. The job allows for criteria to be specified to pinpoint the results. Each result will have the configured retention policy applied to each AIU/table row.

You must select the application to run the job against. Searches are specific to applications so you can only select one application at a time. This job has to be application scoped.

Once executed, if an error is encountered, the error message indicates the number of records that were skipped. If specifying an event or mixed policy, the job will fail if the context is not known.

**Note:** Even if the job fails, it will attempt to apply retention to as many records as possible.

The `searchCriteriaFile` property is required to provide values for all input fields for the query. When this property does not contain all of the declared input fields, execution of the XQuery fails and an XQuery exception is issued. Ensure that, for each declared input field, a value (possibly empty) is defined. If one or more input fields are missing, the job will fail and an error message is issued.

The following parameters can be configured:

Parameter	Description
<i>searchName</i>	This is the name of the search to execute. The search must be defined in the application and must be ready.
<i>searchSet</i>	Search composition within the search to use.
<i>searchCriteriaFile</i>	<p>This is a path to a criteria file that contains what you want to narrow the results to. The following is an example of a SIP search:</p> <pre>&lt;data&gt;&lt;criterion&gt;&lt;name&gt;CustomerID&lt;/name&gt;&lt;operator&gt;EQUAL&lt;/operator&gt;&lt;value&gt;000103&lt;/value&gt;&lt;value&gt;391&lt;/value&gt;&lt;/criterion&gt;&lt;/data&gt;</pre> <p>Example of a table search: &lt;data&gt;&lt;customerID&gt;16&lt;/customerID&gt;&lt;/data&gt;</p> <p>The location is relative to where the server is deployed. If multiple IA Servers are installed, it is recommended that you use a network location (versus a local path).</p> <p>It is also possible to provide the necessary contents of this file in-line into the property instead. This may be a more convenient option because you may not have the ability to modify the server's files system in all deployments.</p>
<i>retentionPolicyName</i>	All four types of retention policies are supported. Depending on the type of policy, other job properties might have to be set.
<i>contextType</i>	<p>Used for Event policies. There are two possible values:</p> <ul style="list-style-type: none"> <li>Attribute: The job gets the context from an attribute in the data. The attribute is set in the context field.</li> <li>Fixed: The job uses the value in this field as the context for the events.</li> </ul> <p>For example, if you want to have all records for an employee age for 5 years after she/he has left the company, make the context the employee ID, since it will be the same for all documents. InfoArchive groups all documents with the same context together. Trigger the event using this context. All records associated with this context (for example, employee number) will be eligible for disposition in 5 years).</p>
<i>context</i>	The value is dependent on the contextType property. Either the attribute (its value) to be used as the context or a value entered into this field will be used as a context for all records that are returned by the search.

Parameter	Description
<i>retentionDateAttribute</i>	This value is used for Duration and the Duration portion of Mixed retention policies. Duration policies that are applied to records need to have a date to calculate the age of the record. The date will be taken from the attribute specified in this field. If the attribute does not have a date, the record will be skipped, meaning a policy will not be applied
<i>trigger</i>	Event policies can be triggered using the records data. There are two possible values for this property: <ul style="list-style-type: none"> <li>True: The job will attempt to trigger the event policy that was applied to the records.</li> <li>False: Set the value to false if not using either event or mixed retention policy.</li> </ul>
<i>triggerCheckAttribute</i>	The job has the ability to use an attribute to determine if the trigger should be performed. For example, if the data has a field that indicated if the employee has left the company (for example, <i>hasLeft</i> = true). The attribute is ' <i>hasLeft</i> ' and the value is 'true'. In this case, the <i>TriggerCheckValue</i> would be set to 'true'. If the value in the attribute is anything but the trigger check value, then the event would not be triggered.
<i>triggerCheckValue</i>	This is value to check against for whether the event should be triggered.
<i>triggerDateAttribute</i>	The job requires a date to trigger the event. This property is an attribute name where the job will get the trigger date (for example, if the event is to keep all employee records 5 years after the employee leaves the company). There would be an attribute called ' <i>terminationDate</i> ' that contains the date the employee left the company. Putting <i>terminationDate</i> in the <i>TriggerDateAttribute</i> property would instruct the job to fetch the trigger date from the <i>terminationDate</i> attribute.

## Using the Apply Retention Rule Job

The Apply Retention Rule job applies retention to records using the rule service. This job has to be scoped to an application. The job:

1. Executes the search with the criteria specified in the job parameters.
2. Calls the rules engine for each of the rows returned in the search result.
3. Applies retention to those records that the rule has identified.

The job supports the following parameters:

Parameter	Description
<i>Rule Name</i>	Optional parameter to specify a specific rule in the system to evaluate the records. If this is not set, the system will use all APPLY_RETENTION rules configured for the specific application.
<i>Search</i>	Name of the search to use for determining the records for evaluation.
<i>Search Set</i>	Name of the search set within the search to use.
<i>Search Criteria File</i>	Use to narrow down the result records. See below for examples of the file.
<i>Developer Mode</i>	If this is set to Yes, the job will just evaluate the rule and not actually apply the retention. The job will output the results returned from the evaluate rule (for example, ApplyRetentionBean) to the job log file.

The following is an example of a search SIP criteria file:

```
<data>
 <criterion>
 <name>CustomerID</name>
 <operator>EQUAL</operator>
 <value>000103</value>
 <value>000147</value>
 <value>391</value>
 </criterion>
</data>
```

The following is an example of a table criteria file:

```
<data>
 <birthYear>1981</birthYear>
</data>
```

## Using Rules to Apply Retention

If a search is to be used for applying holds or retention (via the job) for table applications, the column in the search will not be shown when viewing the hold (retention) set or the purge list, for any of the following reasons:

- The field has been marked as masked (option only available for table searches).
- The field has been marked to encrypt the search results.
- The field has not been mapped to a field using the xDB binding (option only available for table searches). This option is only possible if the search has defined both an xDB schema and xDB table. When editing the column, these values are read-only but can be set when editing the main search.

If some complex logic is being used for the field in the XQuery, it is recommended not to map the field as the field will not be shown correctly when viewing the record in purge lists, or sets.

When writing rules, all rules are evaluated, and it is possible that multiple rules could fire causing a record to have multiple retention policies applied. If you expect to have multiple rules apply to a single record, it is recommended that you not specify the remove all retention option on the rule. Depending on which rule is acted on first, it is possible to get either both or only one of the retention policies applied after the rule finishes (for example, one of the rules was meant to replace retention and the other did not).

If this behavior is not desired, Drools provides a mechanism to configure that exactly one rule will fire.

The following example (for apply hold) illustrates how to use activation groups and salience to provide an ordering. More information about activation-groups can be found in Drools documentation:

```
rule "Apply hold to customer ID 000103 and 000147"
salience 3
activation-group "matchrule"
when
$aiuRecordBean:AiuRecordBean();
eval($aiuRecordBean.getRecordRows().get("CustomerID").equals("000103") || $aiuRecordBean.
getRecordRows().get("CustomerID").equals("000147"))
then
ApplyHoldBean $applyHoldBean = new ApplyHoldBean();
$applyHoldBean.setHold("PhoneCallsGranular-hold");
// the hold set name is always mandatory
$applyHoldBean.setHoldSetName("Test Rule for specific customers");
$applyHoldBean.setId($aiuRecordBean.getId());
insert($applyHoldBean);
System.out.println("Apply hold for records " + $aiuRecordBean.getId() + " is the ID for set " +
$applyHoldBean.getHoldSetName());
end
rule "Apply hold to customers whose first name start with N"
salience 2
activation-group "matchrule"
when
$aiuRecordBean:AiuRecordBean();
eval($aiuRecordBean.getRecordRows().get("CustomerFirstName").startsWith("N"))
then
ApplyHoldBean $applyHoldBean = new ApplyHoldBean();
$applyHoldBean.setHold("PhoneCallsGranular-hold");
// the hold set name is always mandatory
$applyHoldBean.setHoldSetName("Test Customers starting with N");
$applyHoldBean.setId($aiuRecordBean.getId());
insert($applyHoldBean);
System.out.println("Apply hold for records " + $aiuRecordBean.getId() + " is the ID for set " +
$applyHoldBean.getHoldSetName());
end
rule "No match"
salience 1
activation-group "matchrule"
when
$aiuRecordBean:AiuRecordBean();
then
System.out.println("Rule does not match for records " + $aiuRecordBean.getId());
end
```

The example above was meant for debugging to determine why none of the rules were matching and to ensure that only one hold was applied.

## Running Rules to Apply Retention for a Table Application

1. Install the sample PhoneCallsGranular application. This application creates two retention policies, mixed and event.

This application creates two retention policies, mixed and event.

The holding is configured to not specify a retention class (as retention will be on records).

We install two sample rules, one for applying retention and another for fulfilling events.

2. Navigate to <http://localhost:8080> and login as an administrator.
3. In the **Administration > Jobs** tab, edit the Apply Retention Rule to Records job.
4. Set the apply to the PhoneCallsGranular application and update the following properties:

Property	Value	Notes
<i>ruleName</i>		Can specify to limit job to run only one rule
<i>searchName</i>	FirstName_Operator	Needs to be Ready
<i>searchSet</i>	Set 1	
<i>searchCriteriaFile</i>	<data></data>	Can be used to narrow the criteria of which records to evaluate the rule against
<i>developerMode</i>	false	Set to true to test match without performing action

Rules are found under `resources > content > rules`. The rules are not compiled, they can throw runtime exceptions, which stops all rule processing.

It is imperative that any fields referenced by the rule must be defined as fields in the search composition.

For SIP searches, the search set specified indicates which fields retention managers see when expanding the purge lists (or retained sets).

For table searches, all fields in the table are shown except encrypted fields when viewing the purge list.

Rules are found under `resources > content > rules` for the sample application. This is controlled by the ANT script. Rules need to be uploaded either using IA Shell or the ANT script.

When writing rules:

- You need to be familiar with Drools to create rules
- You need to create the bean to do the action:
  - Remember to set the id to identifier from the row:

```
$applyRetentionBean.setId($aiuRecordBean.getId());
```

- Remember to insert the bean into working memory:

```
insert($applyRetentionBean);
```

- If you are using event or fixed date retention, there is no need to set base date

**Tip:** If you are running the job regularly, and the policy value changes, consider setting replace retention to true if you do not want to run-requalification:

- If the same policy was applied multiple times, the policy will only be applied once
- Otherwise, if the rule runs again, the retention policy will not be applied again

Any System.out.println will go to the console when running the server.

## Using the Rule Engine to Apply a Retention Policy or Hold to Records

Customers can use the rules engine to retention policies to individual records, either AIUs or table rows. For instance, you can apply a retention policy to an application using the data in a result set that is returned after executing a search.

Rules are stored as SDX objects in the system. Currently, the only method to create the rules is via REST API. Rules can be applied automatically to records during ingestion or via the search results. A mechanism allows the system to, for instance, calculate the retention date based on a base date derived from the record's metadata.

The following illustrates how InfoArchive can use rules to automatically determine which retention policy to apply:

- In the last version of a trade, a rule can be set up to apply retention to all trades in the version tree using the creation date of the last version as a base date to calculate retention duration. The rules can be expressed based on the metadata values on the record. Rules may require additional inputs, such as a trigger (for example, trade symbol or systemid that identifies all trades in the version tree).
- If the following values are in a record:

```
<RECORD-CLASS>LNS1040</RECORD-CLASS>
<RECORD-TYPE>0000000</RECORD-TYPE>
<RECORD-RETENTION-MARKET1>USD</RECORD-RETENTION-MARKET1>
<REGION1>Americas</REGION1>
<RECORD-RETENTION-MARKET2>JPY</RECORD-RETENTION-MARKET2>
```

A retention policy can be applied, using the base date as the value of the event date field in the record.

- If a record's creation date is between 08/30/2015 to 08/30/2016, and sender is "John Doe", then apply retention policy B.

In the following scenario, the user is applying a retention policy using result set data that was returned from a search. Refer to the InfoArchive REST API documentation to learn how to set up a rule.



1. In the **Administration > Jobs** tab, click > **Edit job** for the Apply Retention Policy to Records job.
2. Set the **Apply To** field to Specific Applications and select the desired application. Because searches are specific to an application, you cannot apply the job to more than one application.

3. In the Properties section, update the **searchName**, **searchSet** and **searchCriteriaFile**. The searchCriteriaFile allows the user to narrow the parameters of the search (for example, search for specific customer IDs).
4. Save and run the Apply Retention Policy to Records job.

The new retained sets appear in the **Retention Sets** tab for the selected application.

## Defining How a Retention Policy is Applied to Records in an AIP or Table

It is important to define how a retention policy is applied to records in an AIP or table. The rules engine allows customers to:

- Easily define the rules for each type of record (for example, by application, table or SIP).
- Allow the rule to access the record metadata.
- Allow for rules to be executed on a scheduled basis.
- Not recompile core InfoArchive code.
- Provide protection so that customers cannot run any malicious code.

The customer needs to know:

- Does the record need to be protected now?
- What is the policy that needs to be applied?
- What is the retention date to calculate age from?

## Rules for Retention Application

The following simple rules govern how to apply retention to objects:

### Ingestion

During SIP ingestion, if the retention policy is configured in the Retention Class, the retention policy will be applied to the package. There is no application of policy to individual AIUs during ingestion. The retention class is configured in the SIP, holding or application. There is no retention application on tables during ingestion.

### Job

There are two jobs that apply retention to objects in the system:

- `ApplyPolicyToRecordsHandler` is the job to apply retention to AIUs and table rows.
- `TableApplyRetentionHandler` is the job to apply retention to tables.

The `ApplyPolicyToRecordsHandler` job takes a search result and applies the configured retention policy to all of the result rows of a search.

## Rule Files

The rules are created in a separate file `<filename>.DRL`, which is a text file with a format similar to XML. Customers have to generate their own files and upload them into InfoArchive using the REST API or IAShell.

The rules are stored in xDB using an SDX object along with the content.

Rules are scoped by

- Application: Rules use the metadata to determine the retention policy, base date and event.
- Type of Rule: Each rule has to be specific to the type because of the return information (apply is different than remove).

Only rules specific to the application being run (for example, within a job) are executed for the result set. you specify within the rule which application and type of policy (apply, remove, etc.). The rule service then matches the rule to the application and type of operation. Only those rules are executed. There will be an option within the jobs, as well as ingestion, to specify the rule to use.

## Loading Rules

Rules are loaded before the rule service is called to evaluate the rules. This is due to the fact that each IA Server has an instance of the rules engine loaded, and the rules loaded into it. If a new rule is added or updated, the rules will have to be reloaded into the rule system before the rules can be executed.

The following is a sample rule file for the PhoneCalls application. There are two rules contained in this rule file. The first rule checks if the customer ID is '000103' or '000147', then it applies Policy C. The second rule checks if the customer is '00391', then apply Policy D.

```
// Rule for applying policy to SIP records
package com.emc.ia.retention.rules
//list any import classes here.
import com.emc.ia.retention.rules.beans.ApplyRetentionBean;
import com.emc.ia.retention.rules.beans.AiuRecordBean;
//declare any global variables here
rule "Apply policy to customer ID 000103 and 000147"
when
 $aiuRecordBean:AiuRecordBean();
 eval($aiuRecordBean.getRecordRows().get("CustomerID").equals("000103"))
 || $aiuRecordBean.getRecordRows().get("CustomerID").equals("000147"))
then
 ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();
 $applyRetentionBean.setRetentionPolicy("Policy C");
 $applyRetentionBean.setBaseDate($aiuRecordBean.getRecordRows().get
 ("CallStartDate"));
 $applyRetentionBean.setEventContext($aiuRecordBean.getRecordRows().get
 ("CustomerID"));
 $applyRetentionBean.setTriggerEvent(false);
 $applyRetentionBean.setIndividualRetention(true);
 $applyRetentionBean.setRetainedSetName("Policy C : " + $aiuRecordBean.
 getRecordRows().get("CustomerID"));
 $applyRetentionBean.setId($aiuRecordBean.getId());
```

```

insert($applyRetentionBean);

System.out.println("Apply Policy A for records for customer ID : " +
$aiuRecordBean.getRecordRows().get("CustomerID") + " AIU Id : " +
$aiuRecordBean.getId() + " Retention Policy : " + $applyRetentionBean.
getRetentionPolicy());
end
rule "Apply policy to customer ID 000391"
when
 $aiuRecordBean:AiuRecordBean();
 eval($aiuRecordBean.getRecordRows().get("CustomerID").equals("000391"))
then
 ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();
 $applyRetentionBean.setRetentionPolicy("Policy D");
 $applyRetentionBean.setBaseDate($aiuRecordBean.getRecordRows().get
("CallStartDate"));
 $applyRetentionBean.setEventContext($aiuRecordBean.getRecordRows().get
("CustomerID"));
 $applyRetentionBean.setTriggerEvent(false);
 $applyRetentionBean.setIndividualRetention(true);
 $applyRetentionBean.setRetainedSetName("Policy D : " + $aiuRecordBean.
getRecordRows().get("CustomerID"));
 $applyRetentionBean.setId($aiuRecordBean.getId());
 insert($applyRetentionBean);

 System.out.println("Apply Policy B for records for customer ID : " +
$aiuRecordBean.getRecordRows().get("CustomerID") + " AIU Id : " +
$aiuRecordBean.getId() + " Retention Policy : " + $applyRetentionBean.
getRetentionPolicy());
end

```

If the *when* clause evaluates to true, the *then* clause will be executed. An apply retention result bean is created, populated with the relevant information and then returned to the rule service.

## Access to Metadata

The rule will need to access the metadata of the record to determine the policy and the retention date. The rule needs to access:

- Metadata of the record, which is a simple map based on table or SIP metadata.
- Result bean

The rules are run against a search result. A record bean is inserted into the rules engine working memory. This is the root record bean. There are specific beans for AIPs, AIUs, tables and table rows that have specific information in a fixed field. All dynamic column data that the customer ingests into InfoArchive will be handled via a map. The ID of the record is set and returned in the apply retention result bean.

The columns for each row is stored in a map. The key to the map is the name of the column, and the value is the value of the column. For instance, for the PhoneCalls application, using Date\_Operator search, the columns that would be the keys to the map are as follows:

```

ID
CustomerID
CustomerFirstName
CustomerLastName
CallFromPhoneNumber
CallToPhoneNumber
CallStartDate

```

CallEndDate

The column name is found in the Column Name field of the result list.

To declare the AIU record bean variable inside the rule, use the following line:

```
$aiuRecordBean:AiuRecordBean();
```

This creates a variable within your rule called `aiuRecordBean`.

You will first need to import the bean into the rule:

```
import com.emc.ia.retention.rules.beans.AiuRecordBean;
```

To access this variable, access it in a manner similar to how Java accesses member variables

```
$aiuRecordBean.getRecordRows()
$aiuRecordBean.getId()
```

If your rule evaluates to true, return the `ApplyRetentionBean` to tell the rule engine how to apply the retention policy:

Parameter	Description
<code>RetentionPolicy</code>	Name of the retention policy to apply to this record.
<code>BaseDate</code>	For duration-based policies (Duration and Mixed) this is the date the retention system will use to start aging from for this record.
<code>EventDate</code>	For event-based policies (Event and Mixed) this is the event date (trigger date) that the retention system will use to start aging the record.
<code>TriggerEvent</code>	Trigger the event-based on the context specified in the Context field.
<code>EventContext</code>	Context for event-based policies. Allows for grouping of records and all records using this context and policy will use the same event date.
<code>Id</code>	This is the AIU or table row ID. This value must be set from the <code>AiuRecordBean/TableRowRecordBean : \$applyRetentionBean.setId(\$aiuRecordBean.getId())</code> .
<code>NewRetainedSet</code>	Indicates to the create a new retained set for this record. If the retained set name is not set, the retention system will generate the name.
<code>RetainedSetName</code>	This is the name of the retained set to put this record in. If the name does not exist, the retained set will be created.

Parameter	Description
<i>ReplaceRetention</i>	If the record has retention applied, the retention system will remove the existing retention and then apply the policy indicated in the Retention Policy field.
<i>IndividualRetention</i>	Indicates to the retention system to age this record individually with its own date.

## Record Bean

```
/*
package com.emc.ia.retention.rules.beans;
import java.util.Map;
public class RecordBean {
 private Map<String, String> recordRows;
 private String id;
 public Map<String, String> getRecordRows() {
 return recordRows;
 }

 public void setRecordRows(Map<String, String> recordRows) {
 this.recordRows = recordRows;
 }
 public String getId() {
 return id;
 }
 public void setId(String id) {
 this.id = id;
 }
}
```

## AipRecordBean

```
/*
package com.emc.ia.retention.rules.beans;
import com.emc.ia.systemdata.oais.Aip;
public class AipRecordBean extends RecordBean {
 private Aip aip;
 public Aip getAip() {
 return aip;
 }
 public void setAip(Aip aip) {
 this.aip = aip;
 }
}
```

## AiuRecordBean

```
/*
package com.emc.ia.retention.rules.beans;
```

```
public class AiuRecordBean extends AipRecordBean {
}
```

## TableRecordBean

```
*/
package com.emc.ia.retention.rules.beans;
import com.emc.ia.systemdata.sql.Table;
public class TableRecordBean extends RecordBean {
 private Table table;
 public Table getTable() {
 return table;
 }
 public void setTable(Table table) {
 this.table = table;
 }
}
```

## TableRowRecordBean

```
*/
package com.emc.ia.retention.rules.beans;

public class TableRowRecordBean extends TableRecordBean {
}
```

Only AIUs and table rows are using the rule engine to apply retention. The rule service passes either the AiuRecordBean or the TableRowRecordBean into the rule for each row returned from the search. The rule service determines the type of application and passes the appropriate bean. The rule service populates the AIP and table for each of the rows, as well.

## Execution Rules

The rules will be executed by one IA Server. There is no requirement for multiple servers to execute the rules at the same time.

Rules need to be accessible by all IA Servers so that the process (job or ingestion) that is running on a server can access to the rules.

There will be a rule service that can be called from anywhere in InfoArchive to execute the rules. The two use cases are for the input to the rule be a search result (AIU/table row) or the result of an XQuery (ingestion of AIU).

The record bean will be populated with the record data, inserted into working memory and then the rule will be executed.

```
private AiuRecordBean createAiuResultBean(ResultRow resultRow, UUID
applicationId) {
 AiuRecordBean aiuRecordBean = new AiuRecordBean();
```

```

Map<String, String> recordRows = new HashMap<String, String>();
// need to iterate through the columns and populate the column bean;
for (ResultColumn resultColumn : resultRow.getColumns()) {
 LOGGER.debug("AIU Column Data : {}:{}",
 resultColumn.getName(),
 resultColumn.getValue());
 recordRows.put(resultColumn.getName(), resultColumn.getValue());
}
aiuRecordBean.setId(resultRow.getId());
aiuRecordBean.setRecordRows(recordRows);
aiuRecordBean.setAip(aipRepository.findOnePartitioned(UUID.fromString(
 (aiuTypeProcessor.getPartitionKey(resultRow.getId())), applicationId)));
return aiuRecordBean;
}

private TableRowRecordBean createTableRowResultBean(ResultRow resultRow) {
 TableRowRecordBean tableRowRecordBean = new TableRowRecordBean();
 Map<String, String> recordRows = new HashMap<String, String>();
 // need to iterate through the columns and populate the column bean;
 for (ResultColumn resultColumn : resultRow.getColumns()) {
 LOGGER.debug("Table Column Data : {}:{}",
 resultColumn.getName(),
 resultColumn.getValue());
 recordRows.put(resultColumn.getName(), resultColumn.getValue());
 }
 tableRowRecordBean.setId(resultRow.getId());
 tableRowRecordBean.setRecordRows(recordRows);
 tableRowRecordBean.setTable(tableRepository.findOne(UUID.fromString(
 (tableRowTypeProcessor.getPartitionKey(resultRow.getId())))));
 return tableRowRecordBean;
}

private void insertResultRow(KieSessionBean kieSession, ArchiveType
archiveType, ResultRow resultRow, UUID applicationId) {
 if (archiveType == ArchiveType.SIP) {
 kieSession.insert(createAiuResultBean(resultRow, applicationId));
 } else {
 kieSession.insert(createTableRowResultBean(resultRow));
 }
}
}

```

## Rules Object

The rules are stored in the xDb database as an SDX object (for scoping by Application and type of rule) and as a content object (DRL file).

```

package com.emc.ia.retention.rules;
import javax.validation.constraints.NotNull;
import org.apache.commons.lang3.builder.EqualsBuilder;
import org.apache.commons.lang3.builder.HashCodeBuilder;
import org.apache.commons.lang3.builder.ToStringBuilder;
import org.apache.commons.lang3.builder.ToStringStyle;
import org.springframework.data.xdb.annotations.Document;
import org.springframework.data.xdb.annotations.DocumentRef;
import org.springframework.data.xdb.annotations.Library;
import org.springframework.data.xdb.annotations.PathValueIndex;
import org.springframework.data.xdb.annotations.PathValueIndex.Path;
import com.emc.compliance.audit.VersionedObject;
import com.emc.ia.systemdata.core.Application;
import com.emc.ia.systemdata.core.Content;
import com.xhive.dom.interfaces.XhiveLibraryIf;
import com.xhive.index.interfaces.XhiveIndexIf;
@Document(filter = false)
@Library(value = "Rule", options = XhiveLibraryIf.CONCURRENT_LIBRARY)
@PathValueIndex(value = {
 @Path(value = "application", type = XhiveIndexIf.TYPE_STRING),
 ...
})

```

```
@Path(value = "name", type = XhiveIndexIf.TYPE_STRING),
@Path(value = "type", type = XhiveIndexIf.TYPE_STRING) },
options = XhiveIndexIf.CONCURRENT, name = "Application+Name+Type")
@PathValueIndex(value = {
@Path(value = "application", type = XhiveIndexIf.TYPE_STRING),
@Path(value = "type", type = XhiveIndexIf.TYPE_STRING) },
options = XhiveIndexIf.CONCURRENT, name = "Application+Type")

public class Rule extends VersionedObject<Rule> {
 private String name;

 @DocumentRef(lazy = true)
 private Application application;
 @DocumentRef(lazy = true)
 private Content drl;
 @NotNull
 private RuleType type;
 public String getName() {
 return name;
 }
 public void setName(String name) {
 this.name = name;
 }
 public Application getApplication() {
 return application;
 }
 public void setApplication(Application application) {
 this.application = application;
 }
 public RuleType getType() {
 return type;
 }
 public void setType(RuleType type) {
 this.type = type;
 }
 public Content getDrl() {
 return drl;
 }
 public void setDrl(Content drl) {
 this.drl = drl;
 }
 public enum RuleType {
 APPLY_RETENTION,
 REMOVE_RETENTION,
 APPLY_HOLD,
 REMOVE_HOLD,
 TRIGGER_EVENT
 }
}
```

## REST API

The capabilities of the REST API are normal CRUD operations that fetch the rule.

The REST API is located off the application resource:

<http://localhost:8765/systemdata/applications/<application id>/rules>

HTTP	Description
POST	Creates a rule.
GET	Gets the list of rules, one rule or content of the rule.
PUT	Updates the rule.

For get operations the following parameters are supported:

Parameter	Example	Description
<i>name</i>	?name='MyRule'	Returns a list of rules that match the name within an application.
<i>spel</i>	?spel?type=='APPLY_RETENTION'	Returns a list that matches the spEL expression. <b>Note:</b> Calculated fields are not available.

For post operations, the following parameters are supported:

Parameter	Example	Description
<i>storeName</i>	?storeName='file_store_01'	The name of the store for the content file.
<i>file</i>	Content-Disposition: form-data; name="file"; filename="demo.drl" Content-Type: text/plain	Multi-part file that contains the name of the file to upload.

The request body will be the rule structure. The local variables that need to be set are:

```
name
type
 APPLY_RETENTION
 REMOVE_RETENTION
 TRIGGER_EVENT
```

## IAShell

The following is an example of how to create the rule using IAShell. First, create a file called rule.xml:

```
<?xml version="1.0"?>
<configuration>
 <object typeAlias="rule" checkExistSpEL="?[name=='my-rule1']" var-
 set="myRuleSelf">
 <create>
 <name>my-rule1</name>
 <type>APPLY_RETENTION</type>
 </create>
 <content storeName="file_store_01" format="drl" file="c:/cli/
 xml/rule.drl"/>
 </object>
```

```
</configuration>
```

IAShell command:

```
configure --from "xml/rule.xml"
```

## Applying Retention to Records for SIP

If retention is to be applied to records, there are two choices:

- Apply retention at ingestion; or
- Apply retention later using a job.

If applying retention to records at ingestion, this slows down the ingestion process, particularly if the AIP contains a large number of AIUs.

Applying a retention policy using a job can scale better, but reevaluating the search criteria each time can be costly

If applying retention during ingestion, there are two methods:

- Use a duration based retention policy (configurable via the holding wizard); or
- Use rules (requires manual configuration).

Using the holding wizard is certainly simpler than uses rules. Rules gives more flexibility and must be used if a duration retention policy will not suffice.

## Applying Granular Retention Using the Holding Wizard

While you are able to apply record-based retention via the rules-based retention mechanism, it is a complex procedure, as a rule has to be created and the job must be configured.

The holding wizard allows the Developer to set granular retention without the need for rule-based retention. Granular retention simply means that each AIU can have its own unique disposition date.

When applying granular retention, the holding wizard allows you to select a base date from a specific column in the AIU metadata (for example, invoice date). The Developer also selects a fixed duration retention policy that should be applied.

1. Complete steps 1 to 5 in the [Holding wizard](#).
2. Configure the following information in the **Retention** step of the wizard.  
The **Retention > Retention Policy** step of the wizard allows you to select a retention policy to be applied to packages defined by this holding.
  - a. Set the **Retention at ingestion** field to **Granular**. Retention will be applied to each record.  
If a retention policy is selected to be applied during ingestion, a **Retention Policy Details** panel is displayed.
  - b. Select a retention policy to apply for the holding.

- c. Select a field that will represent the base date for the holding.
  - d. Complete the following fields that are used to determine which fields to display when a Retention Manager views the retained set or, once the records become eligible for disposition, the purge candidate list. Specifically, the columns on the master list (not the side panel) will be visible:
    - Enter a **Search Name**.
    - Enter a **Search Set Name**.
  - e. Click **NEXT**.
3. The **Retention > Retention Classes** step of the wizard allows you to add or remove retention classes.
- To add a new retention class:
- 
- a. Click 
  - b. Enter a name in the **Retention Class Name** field.
  - c. Select a policy from the **Retention Policy** list.
4. The **Retention > Retention Partitioning** step of the wizard allows you to select a partitioning scheme for the retention data:
- a. Select a **Partitioning Scheme for package based retention**.
  - b. Select a **Partitioning Scheme for record based retention**.
5. Review the information for the holding. When satisfied, click **FINISH**.
- To export the holding configuration in a declarative or YAML format, click the down arrow beside the **FINISH** button.

## Checking that Data is Protected

You can verify that data is protected via the REST API and IAShell.

Another method to verify that data is, in fact, protected is to use IA Web App. Below is a section dedicated to [Using the Retention Sets Tab](#). Also refer to the following sections:

- [Verifying Using the Application Info Tab](#)
- [Verifying Using the Application's Tables Tab](#)
- [Verifying Using the Application's Packages Tab](#)
- [Verifying Retention in Search Results](#)
- [Viewing Items in a Retained Set](#)

## Using the Retention Sets Tab

When a retention policy is applied to records, the records are tracked as a retention set. The **Retention Sets** tab shows the retention sets that are associated with the selected application.

Each retention set is displayed in a table that contains the following information:

Column	Description
Retention Set Name	The name of the retention set.
Item Type	Indicates the item that comprises the retention set (for example, application, package, table, etc.).
Created Date	The date the retention set was created.
Associated Policy	Indicates the retention policy applied to the retention set.
Aging Strategy	Indicates the type of retention policy.
Items	Indicates the number of items that are contained in the retention set.
Qualification Date	The date the retention qualifies for disposition. This value can be blank if using apply retention jobs.

## Viewing Items in a Retained Set

1. Select the application in which the retention set is stored.



2. In the **Retained Sets** tab, click for the retention set you want to view. The following information is displayed:
  - The name of each item contained in the retention set,
  - The state of each item,
  - Whether a hold has been applied against each item, and
  - The number of records contained in the item. The information that is displayed depends on the type of item. For example:
    - If retention was applied directly to the application, information about the application is displayed.
    - If retention was applied to a package, the state of the package is displayed.
    - If retention was applied to a table, the number of records in the table is displayed.

**Note:** Retained sets are no longer shown for AIPs that are ingested where retention is applied automatically.

## Example of Granular Retention

The following is set of instructions for understanding how granular retention works using the PhoneCallsGranular application:

1. Install the application.

The path you choose can be the location that you decided to install.

2. Login as Adam and navigate to the **Administration > Jobs** tab. Edit the Apply Retention Rule to Records job.
3. Enter the following values:
  - a. Ensure that the Schedule By field is set to **Manual**.
  - b. Ensure the Apply To field is set to **Specific Application** and that the PhoneCallsGranular application is selected.

Enter the following values in the Properties section:

Property	Description
ruleName	Leave blank. If you only want one rule to run, use this job.
searchName	<p>Value: FirstName_Operator</p> <p>This could be any of the searches defined for the application. The search must be in Ready status. Case sensitive.</p>
searchSet	<p>Value: Set 1</p> <p>Needs to match the name of a search set. Controls which fields are shown when viewing and which fields are exported.</p>
searchCriteriaFile	<p>&lt;data&gt;&lt;/data&gt;</p> <p>This is the query parameters for the search. We use no criteria as we want the rule to evaluate all records. If your search set specified mandatory parameters, you will need to specify the parameters.</p>
developerMode	<p>Value: False</p> <p>Leave as default. Turning on is meant to test if your rule is firing as expected and does not apply the policy to the record.</p>

The job must be scoped to an application and the expectation is that there is at least one rule defined in the application.

4. Run the Apply Retention Rule to Records job.

The interface does not automatically refresh itself so click on the job name to refresh the page. Check the Status column to ensure the job executed successfully.

If you are running the server manually, you can also see information about the job:

```

Console2 - bin\infoarchive-server.bat
File Edit View Help
Console2 - bin\infoarchive-server.bat | Console2 - bin\infoarchive-server.bat | Console2 - bin\infoarchive-server.bat
be5c-d6b1e21de10f:aiu:2 Retention Policy : Policy B
Apply Policy B for records for customer ID : 000103 AIU Id : ea2bc294-d074-4956-
be5c-d6b1e21de10f:aiu:3 Retention Policy : Policy B
Apply Policy A for records for customer ID : 000549 AIU Id : 4ac2954e-1ae5-4d81-
b84c-9d25760cf2ab:aiu:3 Retention Policy : Policy A
Apply Policy A for records for customer ID : 000435 AIU Id : a07fb9f7-209d-456e-
aaa0-6f35b5032896:aiu:1 Retention Policy : Policy A
Apply Policy A for records for customer ID : 000022 AIU Id : a07fb9f7-209d-456e-
aaa0-6f35b5032896:aiu:2 Retention Policy : Policy A
Apply Policy A for records for customer ID : 000323 AIU Id : 49a93b1c-9467-4bfb-
961e-85d306c6630a:aiu:3 Retention Policy : Policy A
Apply Policy A for records for customer ID : 000823 AIU Id : 49a93b1c-9467-4bfb-
961e-85d306c6630a:aiu:4 Retention Policy : Policy A
Apply Policy A for records for customer ID : 000702 AIU Id : c2b675fa-743c-465f-
8646-ce9aa5331b37:aiu:3 Retention Policy : Policy A
Apply Policy A for records for customer ID : 000103 AIU Id : ea2bc294-d074-4956-
be5c-d6b1e21de10f:aiu:3 Retention Policy : Policy A
2017-04-11 17:19:55.242 [54] [a3e5fe91-d171-4fa3-b9a5-cbf478d6d2ba] [{}]
{system} job: name=Apply Retention Rule to Records, application=PhoneCallsGranular,
scheduled by adam@iacustomer.com, scheduled at 2017-04-11T21:19:50.414Z, attempt=1
- Could not delete temporary file when loading rules C:\Users\saudes\AppData\Local\Temp\rule4487062482155067115.drl
Ready
25x80

```

Ignore the message about the delete temporary file. Notice that some of the records have policy A or B applied.

5. Login as Rita the retention manager to see details about the two retention policies in the **Compliance > Retention Policies** tab.

Policy Name	Policy Category	Aging Strategy	Approved Date	Policy Approval	In Use	Last Modified
Policy B	Event			✓	Apr 11, 2017 4:5...	
Policy A	Mixed			✓	Apr 11, 2017 4:5...	
PhoneCalls-pol...	Duration			✓	Apr 11, 2017 2:2...	

**Policy A**

**Policy Name:** Policy A  
**Description:**  
**Policy Category:**  
**Aging Strategy:** Mixed  
**Retain for:** 50 Years  
**Except Condition:** NoLongerNeeded  
**Then retain for:** 0 Years  
**Disposition Strategy:** Destroy All

Because the mixed policy is for 50 years, the records will not qualify yet (unless the event is fulfilled). The event based retention policy will not start aging until the event happens.

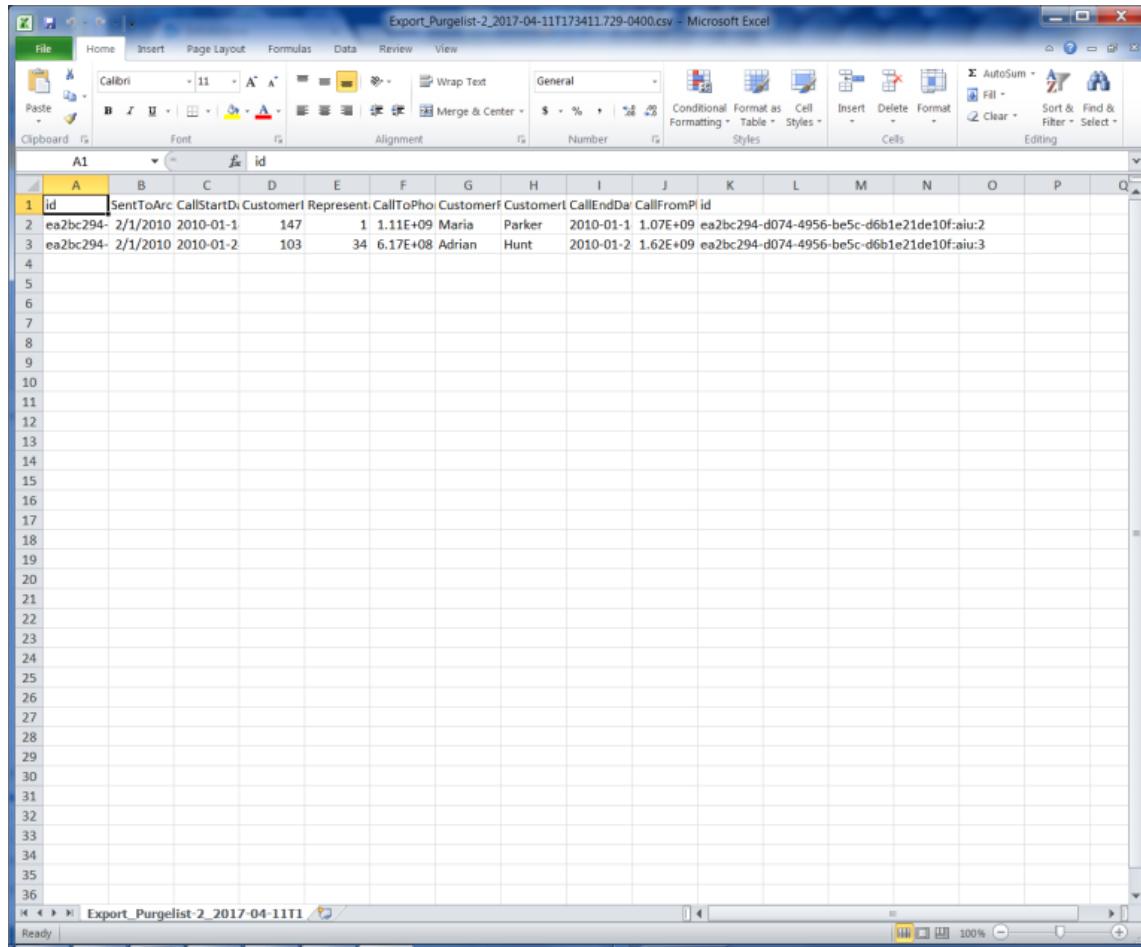
6. Login again as Adam the Administrator and navigate to the **Administration > Jobs** tab. Edit the Trigger Event Rule job to fulfill some events.

7. Only update the Apply To field by selecting the PhoneCallsGranular application.
8. Run the Trigger Event Rule job.
9. Edit the Generate Purge Candidate List job by updating the Schedule By field to **Manual**.
10. Run the Generate Purge Candidate List job.
11. Login as Rita the Retention Manager. Access the **PhoneCallsGranular application > Purge Lists** tab.

Expand the purge list to view the following:

The screenshot shows the Infoarchive application interface. At the top, there's a navigation bar with tabs: Applications (selected), Dashboard, Compliance, Background Requests, and a user profile for rita@iacustomer.com. Below the navigation bar, the main menu includes Record Search, Retention Sets, Hold Sets, Purge Lists (selected), Application Info, Packages, and Holdings. The current page is 'Purge Lists' under the 'PhoneCallsGranular' application. On the left, a sidebar shows an 'Associated Policy' dropdown set to 'All'. The main content area displays a table with one row, labeled 'Displaying 1 - 1 of 1'. The table columns are: Purge List Name, Type, Associated Policy, Item, Created Date, and Status. The single row shows 'Purgelist-2', 'Rec...', 'Policy B', '2', 'Apr 11, 2017 5:30:28 ...', and 'Under Review'. To the right of the table, a detailed view of 'Purgelist-2' is shown with the following fields: Type (Records), Retention Policy (Policy B), Items (2), Created Date (Apr 11, 2017 5:30:28 PM), Modified Date (Apr 11, 2017 5:30:28 PM), Modified By (system), and Status (Under Review). A note at the top right says 'Actual dispositions may change.' with a help icon.

12. Export the purge list.
  13. Navigate to the **Background Requests** tab and click **Download** for the exported data.
- When you double-click the .csv file, the following should be displayed:



**Note:** It is possible to control which fields are exported from the search screens. The ID field is always exported as the first column, even if no columns were configured to be exported.

14. Since we know that we have the correct records, approve the purge list.
  15. Login as Adam the Administrator. Edit the Dispose Purge Candidate List job and ensure that the Schedule By field should is set to **Manual**.
  16. Run the Dispose Purge Candidate List job.
  17. Login as Rita the Retention Manager. Access the **PhoneCallsGranular > Purge Lists** tab. Expand the purge list:

Purge Lists

Associated Policy: All

Purge List Name	Type	Associated Policy	Items	Created Date	Status
Purgelist-2	Records	Policy B	2	Apr 11, 2017 5:30:28 PM	Disposed

**EXPORT**

Displaying 0 - 0 of 0

Purgelist-2

Type: Records  
Retention Policy: Policy B  
Items: 2  
Created Date: Apr 11, 2017 5:30:28 PM  
Modified Date: Apr 11, 2017 5:42:10 PM  
Modified By: system  
Status: Disposed  
Disposed: Apr 11, 2017 5:42:10 PM  
Items Disposed: 2  
Items Skipped: 0

In this case, the two items were disposed and, because the packages still have items in it, we did not require a confirmation and the package can be pruned.

**Note:** If records had have been put on hold, they will still show up in this report (as they were not disposed) and it is possible to export what was left in the purge list.

18. Navigate to the Packages tab. Notice that two records have been removed from the first package:

Packages

Select all 10

Displaying 1 - 10 of 10

Type	Name	Phase	Holding	Receptio...	Online	Retention	Hold	Records
PhoneC...	PhoneC...	Comple...	PhoneC...	Apr 11, 2...	✓			8
PhoneC...	PhoneC...	Comple...	PhoneC...	Apr 11, 2...	✓			3
PhoneC...	PhoneC...	Comple...	PhoneC...	Apr 11, 2...	✓			3
PhoneC...	PhoneC...	Comple...	PhoneC...	Apr 11, 2...	✓			3
PhoneC...	PhoneC...	Comple...	PhoneC...	Apr 11, 2...	✓			3
PhoneC...	PhoneC...	Comple...	PhoneC...	Apr 11, 2...	✓			4
PhoneC...	PhoneC...	Comple...	PhoneC...	Apr 11, 2...	✓			3
PhoneC...	PhoneC...	Comple...	PhoneC...	Apr 11, 2...	✓			3
PhoneC...	PhoneC...	Comple...	PhoneC...	Apr 11, 2...	✓			3

REFRESH

PhoneCallsGranular-CC-1000010-1

**Summary**

Name: PhoneCallsGranular-CC-1000010-1  
AIP ID: ea2bc294-d074-4956-be5c-d6b1e21de10f  
AIU count: 8  
CI count: 1  
PDI file size: 7575  
PDI character count: 1729  
CI size: 41123  
Phase: Completed  
State: Completed  
Return code: OK  
Return message:   
Part of aggregate: false

19. Click the package to see logs about the disposition that was done on the package:

## Compliance Related Tasks

**Packages > PhoneCallsGranular > PhoneCallsGranular-CC-1000010-1**

**Package content**

Name	Description	Store	Size	Modified Date
aiu.disposition.log.gzip		file_store_01	410	Apr 11, 2017 5:42:10 PM
ci.container	Aggregated content file of the pack...	file_store_01	41136	Apr 11, 2017 4:56:53 PM
ingest.log.gzip	Compressed ingestion log	file_store_01	1879	Apr 11, 2017 4:56:54 PM
pdi.xml.gzip.crypto	Encrypted and compressed PDI XML	file_store_01	928	Apr 11, 2017 4:56:53 PM
receive.log.gzip	Compressed reception log	file_store_01	313	Apr 11, 2017 4:56:53 PM
rendition.csv.gzip	Dataset for analytics	file_store_01	451	Apr 11, 2017 5:42:10 PM
ri.xml	XML table of contents of the package	file_store_01	701	Apr 11, 2017 4:56:53 PM
sip.xml	XML description of the package	file_store_01	588	Apr 11, 2017 4:56:53 PM

Displaying 1 - 8 of 8

**Package Retention**

No Retentions found

**Package Hold**

Rules can be found in the following directory: <INFOARCHIVE\_ROOT>/examples/applications/PhoneCallsGranular/config/rules.

The following is an example of the two rules that were used to apply retention to different records:

```

// Rule for applying policy to SIP records
package com.emc.ia.retention.rules

//list any import classes here.
import com.emc.ia.retention.rules.beans.ApplyRetentionBean;
import com.emc.ia.retention.rules.beans.AiuRecordBean;

//declare any global variables here

rule "Apply policy to customer ID 000103 and 000147"
when
 $aiuRecordBean:AiuRecordBean;
 eval($aiuRecordBean.getRecordRows().get("CustomerID").equals("000103") || $aiuRecordBean.getRecordRows().get("CustomerID").equals("000147"))
then
 ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();
 $applyRetentionBean.setRetentionPolicy("Policy B");
 $applyRetentionBean.setBaseDate($aiuRecordBean.getRecordRows().get("CallStartDate"));
 $applyRetentionBean.setEventContext($aiuRecordBean.getRecordRows().get("CustomerID"));
 $applyRetentionBean.setTriggerEvent(false);
 $applyRetentionBean.setIndividualRetention(true);
 $applyRetentionBean.setReplaceRetention(true);
 $applyRetentionBean.setRetainedSetName("Policy B : " + $aiuRecordBean.getRecordRows().get("CustomerID"));
 $applyRetentionBean.setId($aiuRecordBean.getId());
 insert($applyRetentionBean);

 System.out.println("Apply Policy B for records for customer ID : " + $aiuRecordBean.getRecordRows().get("CustomerID") + " AIU Id : " + $aiuRecordBean.getId() + " Retention Policy : " + $applyRetentionBean.getRetentionPolicy());
end

//rule "Apply policy to customer ID 000391"
rule "Apply policy to customers that start with A"
when
 $aiuRecordBean:AiuRecordBean;
 eval($aiuRecordBean.getRecordRows().get("CustomerFirstName").startsWith("A"))
then
 ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();
 $applyRetentionBean.setRetentionPolicy("Policy A");
 $applyRetentionBean.setBaseDate($aiuRecordBean.getRecordRows().get("CallEndDate"));
 $applyRetentionBean.setEventContext($aiuRecordBean.getRecordRows().get("CustomerID"));
 $applyRetentionBean.setTriggerEvent(false);
 $applyRetentionBean.setIndividualRetention(true);
 $applyRetentionBean.setRetainedSetName("Policy A : " + $aiuRecordBean.getRecordRows().get("CustomerID"));
 $applyRetentionBean.setId($aiuRecordBean.getId());
 insert($applyRetentionBean);

 System.out.println("Apply Policy A for records for customer ID : " + $aiuRecordBean.getRecordRows().get("CustomerID") + " AIU Id : " + $aiuRecordBean.getId() + " Retention Policy : " + $applyRetentionBean.getRetentionPolicy());
end

```

# Event-Based Retention

Event based retention can be applied using the following mechanisms:

- Using the Apply Retention Rule to Records job
- Via ingestion

The concept of event-based retention revolves around the principle that retention aging does not commence until an event happens. Usually, however, an event has a context associated to it. For example, a retention policy may want to protect employee data associated but not start the retention aging until the employee leaves the organization.

To achieve this, InfoArchive stores events separately from a retention application. This separation allows the ability for multiple retention policies to share the same event and not require the customer to set an event date for each application. This means that if a retention policy is applied, and the event has already been fulfilled, retention aging immediately commences.

## What are Compliance Events?

The compliance system provides the ability for a retention policy to specify a condition. For example, a condition could be EmployeeTermination or TradeCloseDate.

When the retention policy of this type (event or mixed) is applied, the system needs to know what the context will be. For example, for an employee termination, we need to know which Employee ID or, if this was a trade, close the trade transaction identifier.

So if a retention policy is applied to a record, the event has a name and a context so that the system can differentiate between employee A versus employee B leaving the company. It is important to note that since multiple records could be associated with that employee, the same event name and context could be used for multiple records so the event only has to be fulfilled once and then all records can start aging.

Events are modeled at the tenant level, which means events can be shared across applications.

Events are automatically created when applying retention if the retention policy is mixed or event based. Event or mixed retention policies can only be applied to records by using jobs (either using rules or not).

If using rules, the event context must be specified and the rule can be configured to define how to define the retention sets. The retention parameters on the bean can be set to either age individually or together. If multiple records are put into the same set and the set is aging together, one event governs the event that will be used as the base date.

If the Apply Policy to Records job is used, there are parameters that dictate how to determine, based on metadata in the record, what the event context is. Typically, the record will have a field that defines the context such as Employee id). This job always creates retention sets that age individually and the name of the retention policy to use is a mandatory parameter for the job.

## Basic Flow



### Define Retention Policy

The retention policy needs to be either event or mixed and a condition must be specified.

### Apply Retention to Records

There are three ways to apply retention to records, two of them involve rules:

- Apply retention via the Apply Retention Policy to Records job
- Apply retention via the Apply Retention Rule to Records job
- Configure ingestion to use rules to apply during ingestion

It is also possible to configure that the event has been fulfilled when running the Apply Retention Policy to Records job.

**Note:** When applying retention, because there is only one condition, setting the event context is sufficient.

### Fulfill Events

There are two ways to fulfill events:

- Trigger events using the Trigger Event Policy job
- Trigger events using the Trigger Event Rule job

The first option uses an XML file to define which events have been fulfilled:

```
<?xml version="1.0"?>
<triggers>
 <event>
 <context>Morgan</context>
 <triggerdate>2016-02-28</triggerdate>
 <condition>EmployeeTermination</condition>
 </event>
 <event>
 <context>Steve</context>
 <triggerdate>2015-02-28</triggerdate>
 <condition>EmployeeTermination</condition>
 </event>
```

---

```
</triggers>
```

For the Trigger Event job, the job runs a search and decides programmatically which events to fulfill. The PhoneCallsGranular sample application provides a sample rule for triggering events:

```
// Rule for triggering events for SIP records
package com.emc.ia.retention.rules
//list any import classes here.
import com.emc.ia.retention.rules.beans.TriggerEventBean;
//declare any global variables here

rule "Trigger Event"
when
then
 TriggerEventBean $triggerEventBean = new TriggerEventBean();
 $triggerEventBean.setContext("000103");
 $triggerEventBean.setDate("2015-02-28");
 $triggerEventBean.setCondition("NoLongerNeeded");
 insert($triggerEventBean);

 TriggerEventBean $triggerEventBean1 = new TriggerEventBean();
 $triggerEventBean1.setContext("000147");
 $triggerEventBean1.setDate("2015-02-28");
 $triggerEventBean1.setCondition("NoLongerNeeded");
 insert($triggerEventBean1);
 System.out.println("Trigger Event for AIU Records");
end
```

## How Does Aging Work?

For a mixed retention policy, the event is optional and the aging starts using the base date. However, if the event does happen, it takes priority and chronological path. For a mixed retention policy, a base date must be provided.

For an event retention policy, the aging does not start until the event is fulfilled. Base dates specified at policy application time are ignored when applying event based retention policies.

The following are examples of how aging works:

- In one case, you can specify an event retention policy in which the policy is to start the aging after the employee leaves the organization.
- In a different case, you can specify a mixed retention policy in which, if the event does not happen, InfoArchive keeps the records for 7 years. However, if the event does happen, you may choose to have the record immediately eligible for disposition (no aging) or extend the aging.

## Applying an Event-Based Retention Policy

The following steps outline the process of applying an event-based retention policy:

1. The Retention Manager defines a retention policy. An event is also defined and given a name. When a retention policy is created, the Retention Manager does not need to know what the context will be.

2. A process identifies some content that must be protected by the retention policy. The process groups the content and makes separate calls to apply the retention policy using a different context (for example, the Employee ID).  
If additional information becomes available for the same employee, the information can be added to the existing retained set. Alternatively, if the Retention Manager prefers that the information to age independently, the same event (and context) can be reused so that the event only needs to be set once for that employee.
3. When applying the retention policy, the system verifies if the event (and the requested context) exists. If the event already exists, the event does not have to be created. This is important because it could be possible that an event is fulfilled even before the data is ingested into the archive.

## Fulfilling Events with the Trigger Event Policy Job

The Trigger Event Policy job triggers events based on a trigger file that the customer provides. Usually this is a result of another system that would generate the event (for example, an HR system would indicate when the employee left the company). The values of the trigger file contain the context that groups the records together. For example, if you are keeping employee records until 5 years after the employee leaves the company, then you would want to group the records around a common field (for example, context). The context in this case would be the employee number. When the event policy is applied, a context would have been specified. When the event needs to be triggered, a context and a trigger date need to be specified.

The Trigger Event Policy job has to be scoped to an application .

There following parameter can be configured:

Parameter	Description
<i>triggerFile</i>	This is a path to a trigger file that contains a list of triggers (context, trigger date and condition).  The location is relative to where the server is deployed. If multiple IA Servers are installed, it is recommended that you use a network location (versus a local path).

The following illustrates the format of the file:

```
<?xml version="1.0"?> <triggers> <event>
<context>00457</context> <triggerdate>2010-01
-31</triggerdate> <condition>condition</condition>
</event> <event> <context>00345</context>
<triggerdate>2014-02-28</triggerdate>
<condition>condition</condition> </event> </triggers>
```

## Populating Event Dates for the Trigger Event Policy Job

An .xml file is used to populate event dates for the Trigger Event Policy job:

Name	Description
<i>context</i>	Enter the context for the event (for example, employee number).
<i>triggerdate</i>	Enter the date when the event happened or is planned to happen in a YYYY-MM-DD format. A date must be within the following range: 1000-01-01 – 2999-12-31.
<i>condition</i>	Enter the name of the condition, which must match the condition specified on the retention policy. The value is case sensitive.

The following is an example of the .xml file:

```
<?xml version="1.0"?>
<triggers>
<event>
<context>89</context>
<triggerdate>2016-02-28</triggerdate>
<condition>tradeverSION</condition>
</event>
<event>
<context>77</context>
<triggerdate>2016-02-28</triggerdate>
<condition>tradeverSION</condition>
</event>
</triggers>
```

## Choosing Between Event-Based and Mixed Retention Policies

Use the following to determine whether to use an event-based or mixed retention policy:

- If the event may not happen, then you should consider a mixed mode aging retention policy (for example, keep the records for 10 years unless the person dies, in which case, the system keeps the records for one year after death).
- On the other hand, if the event must happen, consider an aging retention policy (for example, keep the records until one year after the person leaves the company. Otherwise, keep the records indefinitely).

## Fulfilling Events with Rules

Event-based policies age based on an event. The event that triggers the aging process may not happen until after the policy is applied to the records. The event that triggers the event might come from an external systems (for example, an HR system).

The following rule can look to an external system to get the event for a customer whose account has been closed. The context is set to the customerId and the date for the event is the date the account was closed:

```
// Rule for triggering events for SIP records
package com.emc.ia.retention.rules

//list any import classes here.
```

```
import com.emc.ia.retention.rules.beans.TriggerEventBean;
//declare any global variables here

rule "Trigger Event"
when
 then
 TriggerEventBean $triggerEventBean1 = new TriggerEventBean();
 $triggerEventBean1.setContext(getCustomerId());
 $triggerEventBean1.setEventDate(getEventDate());
 $triggerEventBean1.setCondition("MyCondition");
 insert($triggerEventBean1);

 System.out.println("Trigger Event for Records");
end

function String getEventDate(){
 if(isAccountClosed(getCustomerId()))
 return getCloseDate();
 else
 return null;
}

function boolean isAccountClosed(String customerId) {
 POJO_Class myClass = new POJO_Class();
 return myClass.isAccountClosed(customerId);
}

function String getCloseDate(String customerId) {
 POJO_Class myClass = new POJO_Class();
 return myClass.getAccountClosedDate(customerID);
}

function String getCustomerId() {
 POJO_Class myClass = new POJO_Class();
 return myClass.getCustomerId();
}
```

## Process Retention Events Job

If events are triggered using the apply retention rule job, this job needs to run periodically in order to ensure that the records are re-qualified.

If rules are not triggering events, this job does not have to run.

## SIP Retention

### When is Retention Effective?

Retention and holds only come into effect after confirmation.

If retention is applied to the package, when the Disposition job runs, the package is set to a purge state. You need to run the Confirmation job. The next time the Disposition job runs, the package will be deleted.

**Note:** Once the package is set to a purge state, it is not possible to search for records in the package.

If granular retention is used, an evaluation is done against the package. If, as a result of the disposition, the package would have no records, the package will be set to a purge state. Otherwise, the package will be re-factored. It is possible to view the disposition logs from the package.

## Confirming a Purge

A confirmation is required if:

- Retention is applied on the package and the package can be disposed (for example, there are no holds or records that need more aging).
- You are using granular disposition, and all of the records for a package would be removed, the package is marked for disposition

## How Does Refactoring Work

The following use case illustrate how refactoring works when a package is partially disposed:

1. If retention is applied on the package, and some of the records are under hold or have longer retention, then the package will be refactored, meaning that no confirmation will be required.
2. If using granular disposition, and the package containing the record would still contain records, then the package will be refactored, meaning that no confirmation will be required.

When a package is refactored, disposition logs are added to package.

If a package is refactored, a back up is taken. No back up is taken, however, if the package is marked for disposition and is waiting for a confirmation.

## Hardware Retention

### When are Dates Pushed to the Hardware?

How dates are pushed to the hardware depends on the data retention is being applied to. For example, there is no hardware protection for applications because dates are not pushed to the hardware. Dates are only pushed to the hardware if retention is applied directly to a package.

Type	Hardware Protection	Requires Two Disposition Runs	Normal Method Retention is Applied
Application	No	Yes	Manually, only via the web application
Packages	Yes	Yes	Automatically during ingestion (either by retention class or via default retention policy on application)
Tables	Yes	No	Manually, usually by the web application but can be done via job
Records	No	Records usually do not require two disposition runs. For SIP applications, however, if all of the records in a package are eligible for disposition, two disposition runs will be required (and a confirmation is requested for the package).  <b>Note:</b> Records are no longer returned in new searches once the first disposition has been done.	Via job

When applying multiple retention policies, if a shorter retention policy is applied, a new date will not be pushed to Isilon storage. Dates can only be pushed further into the future.

Dates are not pushed for granular disposition because managing the dates would be problematic, as the system would be working with a BLOB of content.

It is important to consider the retention policies being applied to data and the dates that are pushed to the hardware. If a policy dictates that data is to be kept for 50 years, that data will be ineligible for disposition for 50 years. This type of long-term storage can greatly increase a company's storage costs. If you are testing the application of retention policies, it is advised that you do not retain data for a long period of time.

You should never remove a retention policy nor add additional retention policies to a package or AIP if you are using ECS storage. This only applies for SIP archiving, however, as the table archiving process does not push the date to the hardware.

Hardware retention is not applied to the backups.

## Hardware Retention Support Considerations for ECS, Centera and Isilon

The retention is applied on the unstructured content associated to the SIP (pdi.xml.gzip, ri.xml, ci.container, sip\_zip, logs, etc.) and table (ri.xml and ci.container, etc.). The hardware retention does not impact SDX and other xDB databases. When the retention is pushed at the hardware level, it is not possible to remove the retention policy until the retention at the hardware level is expired. In this case, it is only possible to extend the retention period. When you enable retention feature on the storage, it is mandatory to not set a default retention at the bucket level. If, in some situation, a default retention period is mandatory (ECS compliance mode, for example), a minimal value needs to be set (1 second with ECS Compliance mode). It is not currently possible to configure a bucket with a default retention. If you plan to use ECS with compliance mode, it is important to create the bucket on ECS first.

## Determining if a Retention Policy is in Use

You have to apply a retention policy for the policy to be considered in use. A policy is not considered in use if it is only referred to in a job configuration. If a retention policy is changed, and the previous policy was not applied, the job will not work because it will not be able to find the retention policy.

There are various methods that you can check to verify that a retention policy is in use:

- Access the **Retention Policies** tab that displays each policy in a table. One of the columns is **In Use**, which indicates whether a retention policy has been applied.
- Check an application to view the Default Retention Policy. On the **All Applications** tab, click  . The Default Retention Policy is displayed.

## Changing a Retention Policy

Retention policies can be changed but there are restrictions if the policy is in use. If the retention policy is in use, the name, the type of retention or any conditions cannot be changed (for example, you will not be able to change a fixed-date retention policy to a duration retention policy).

When editing a retention policy that is in use, you can change:

- The dates included in a duration retention policy type,
- A cutoff date, and
- The name of the policy.

You cannot change a retention policy's type if the policy is in use (for example, you cannot change a fixed date policy to an event-based policy).

You cannot change the name of the retention policy. You can, however, change the setting for disabling disposition processing.

You can also change the:

- Approved Date
- Policy Approver
- Notes



1. In the **Compliance > Retention Policies** tab, select the policy being edited and click .
2. Edit the fields, as desired.
3. Click Save.

## Editing a Retention Policy

When editing a retention policy that is in use, you can change:

- The dates included in a During retention policy type,
- A cutoff date, and
- The name of the policy.

You cannot change a retention policy's type if the policy is in use (for example, you can't change a fixed date policy to an event-based policy).

You cannot change the name of the retention policy. You can, however, change the setting for disabling disposition processing.

The customer can also change the:

- Approved Date
- Policy Approver
- Notes



1. In the **Compliance > Retention Policies** tab, select the policy being edited and click .
2. Edit the fields, as desired. For further field information, see [Creating a Retention Policy](#).
3. Click Save.

## Impact on Existing Items that have Retention Applied

When changing a retention policy, existing objects will continue to age with the settings on the retention policy at the time of application.

## Running the Requalification Job

If retention policies are changed and the changes are retroactive, this job updates the retention information for all managed objects.

Some storage systems may not support updating qualification dates.

This job is necessary to run if a retention policy is changed and the changes need to be retroactive.

This job cannot be scoped to an application and does not have any parameters.

## Limitation of Changing Retention Policies based on the Content Store

The following illustrates the rules for changing retention policies depending on the content store:

- File store – There are no restrictions
- ECS – You cannot change the retention period if applied to tables or packages
- Isilion (Smartlock) – You can only extend the retention period if it is applied to tables or packages
- Amazon S3 – As S3 does not support retention at the hardware level, there are no restrictions

InfoArchive does not push hardware dates if the retention policy is applied to records (table rows or AIUs).

## Disposition Flow

### How are Items Added to a Purge Candidate List?

When items are eligible for disposition, the Purge Candidate List Generation job determines which items are included in a purge candidate list. If holds are applied to items after the list is created, these items will not be disposed of. If approval is not given, the next time the Purge Candidate List Generation job runs, previous lists are marked cancelled and items are eligible to be placed into new lists. The job typically runs monthly, although it can also be run manually.

Purge candidate lists are always associated with an application. Lists are created per application and type of object.

Once the purge candidate list is generated, the retention manager reviews the contents. The retention manager can choose to ignore, reject or approve the list:

- If the list is ignored, the items in the purge candidate list will be added to the new list the next time the Purge Candidate List Generation job runs.
- If a purge candidate list is rejected, items in the purge list will not be eligible for inclusion in new lists until the state of the rejected list is changed.
- If the list is approved, items in the purge candidate list will be disposed. If a hold is subsequently placed on items in the list or put under a longer retention policy that hasn't elapsed, those items will not be disposed, even though approval was given.

Click on a list to view the details of a specific purge list. The details panel contains the following information, depending on the Status of the purge candidate list:

- If the list was disposed, the panel indicates the number of items that were disposed or the number of items skipped.

When disposing packages, sometimes items are skipped (for example, if an additional retention policy was applied to the items.) The customer needs to run the Confirmation job and, the next time the Disposition job runs, the packages will be removed.

- If the list was disposed, the panel indicates the disposal date.
- If the list was cancelled, the panel indicates the cancellation date.

The purge candidate list also indicates the items with qualification dates in the past that could have been disposed sooner.

The following types of objects can be included in a purge list:

- Applications
- Tables
- Table rows
- AIPs
- AIUs

When disposition is successfully run, and the items are purged, the Clean job needs to be executed to clean up the unstructured content.

If a package has a five year retention policy, and a retention policy is applied to search results extending the duration of retention for some records to seven years, the package will be included in a purge list but skipped, even if the package is approved.

**Note:** Once an application is marked Active, the only way to remove records is via disposition.

If retention is applied to records that are included in cached out packages that become eligible for disposition, the first time the Generate Purge List job runs, those items will be included in a purge candidate list.

If retention is applied directly to the AIP, the AIP will be added to the purge candidate list even if it has been cached out.

If a package is approved, the approval of purge lists for the records is ignored:

- If records either have longer retention or holds, the package is re-factored.
- If the purge list for the package is approved, eligible records will be disposed even if the purge lists for those records are not approved or rejected.

If records need to be kept, apply holds before approving the purge list for the package instead of rejecting the purge list for records

If a package no longer has any records after disposition, the package is marked for purge.

Tables are no longer destroyed if retention is applied directly, only the records. If the table is no longer needed, consider applying retention to the application once everything is removed to destroy the application.

## Using Rules to Automatically Approve Purge Candidate Lists

Customers have the option of automatically approving a purge list when it is generated. Customers can also determine which purge list will contain a record.

To enable this feature, use the rule type GENERATE\_PURGE\_LIST to add a generate purge list rule to an application. The record's metadata will be passed into the rule and, therefore, help the rule designer to determine which purge list to place the record in. This process is similar to the Apply Retention rule (refer to [Using Rules to Apply Retention](#) for further information).

By default, approval will be required for all purge lists unless the rule sets the **RequiresApproval** flag to *false*. For example, the following rules, for both SIP and table archives, will put the records in purge lists based on the customer ID. Therefore, all records with the same customer ID will be placed in the same purge list. The **RequiresApproval** is set to *false* to indicate that the purge list does not require approval and will be set to approved once the Generate Purge Candidate List job finishes.

The following is the AIU purge list rule:

```
package com.emc.ia.retention.rules

//list any import classes here.
import com.emc.ia.retention.rules.beans.GeneratePurgeListBean;
import com.emc.ia.retention.rules.beans.AiuRecordBean;

rule "Apply policy to trades"
when
 $aiuRecordBean:AiuRecordBean();
then
 GeneratePurgeListBean $generatePurgeListBean = new GeneratePurgeListBean();
 $generatePurgeListBean.setPurgeListName("PurgeList : " + $aiuRecordBean.
 getRecordRows().get("CustomerID"));
 $generatePurgeListBean.setRequiresApproval(false);
 $generatePurgeListBean.setId($aiuRecordBean.getId());
 insert($generatePurgeListBean);
end
```

The following is the table row purge list rule:

```
package com.emc.ia.retention.rules

//list any import classes here.
```

```
import com.emc.ia.retention.rules.beans.GeneratePurgeListBean;
import com.emc.ia.retention.rules.beans.TableRecordBean;

rule "Apply policy to trades"
when
 $tableRecordBean:TableRecordBean();
then
 GeneratePurgeListBean $generatePurgeListBean = new GeneratePurgeListBean();
 $generatePurgeListBean.setPurgeListName("PurgeList : " + $tableRecordBean.
 getRecordRows().get("CustomerID"));
 generatePurgeListBean.setRequiresApproval(false);
 $generatePurgeListBean.setId($aiuRecordBean.getId());
 insert($generatePurgeListBean);

end
```

## How Does Disposition Work

### Application Disposition

If any item in the application is under hold, disposition will be skipped for the application.

An application is eligible for inclusion in a purge list even if there are items in the application that have longer retention. The application, however, will be skipped when disposition is run.

The recommended practice is that, if application disposition is used, to not associate retention directly to packages, tables or records.

There is no hardware protection for applications because dates are not pushed to the hardware.

When an application is deleted or disposed, all resources within the application are deleted. This includes:

- searches
- rules
- tables / packages (and their records)
- background tasks
- collections
- job instances

Just as a hold applied to an object prevents the disposition of an application's data, any collections associated with a legal matter will also prevent the deletion of application data.

To delete an application that is under retention:

1. Wait for the retention period of the retention policy applied to the application to elapse.
2. Run the Generate Purge List job.
3. Approve the purge candidate list.
4. Ensure that there are no holds applied to any items within the application. Ensure that no collections are associated with a matter. Also ensure that no packages or records have retention applied that have a qualification date in the future (or have not qualified if event based).

5. Run the Disposition job. The Disposition job must be scoped to the application (multiple applications can be scoped).
6. Run the Clean job.
7. Run the Clean up Purge Candidate List and Application jobs.

**Note:** If any items in the application are under hold or cannot be disposed, the purge candidate list will be set to be disposed, but the application will be skipped. Nothing will be disposed in this case.

## AIP Disposition

Package disposition requires an additional step after running disposition the first time. By default, after disposition is run, the packages are marked to be waiting for confirmation. The intent is that the Administrator runs the Confirmation job manually, which marks all packages confirmed. Once done, the next run of disposition will remove the packages unless a hold or a new retention policy is applied.

If any item in the package is under hold, the package will not be skipped when disposition runs. If not all of the records in the package are eligible for disposition (either longer retention or under hold), the AIP will be re-factored and a new AIP will be created with those records removed.

If there are AIUs in the package are under hold, when the package is re-factored, a log is added to the package and can be viewed from the list of files that are part of the AIP.

There are audits that can be enabled to distinguish if the AIP was disposed or partially disposed as well as an audit for each AIU.

This also means that there will be an audit if the either the AIP is disposed or its children were disposed.

If a package is marked for purge, all records in the package no longer show up in the search results.

## AIU Disposition

If all the AIUs in a package are eligible for disposition and approval is given, when running the Dispose Purge Candidate List job, the records are skipped and the package is marked for purge. Once the Confirmation job runs, run the Dispose Purge Candidate List job and, at this point, the audit for disposing the AIUs will be created. The Clean job needs to be run clean up the content for the AIUs.

If only some of the AIUs in a package are eligible for disposition and approval is given, when running the Dispose Purge Candidate List job, the records will be processed and there is a configurable audit for the partial disposal of the AIP as well as for the disposition of each AIU. The Clean job needs to be run to clean up the content for the AIUs that were disposed. If content was shared between multiple AIUs, the content is not removed until all references are removed.

## Disposition of AIPs and AIUs

If the AIPs and the AIUs are both eligible for disposition, a purge list will be created for the AIP and one or more purge lists will be created for the AIUs.

If the purge list for an AIP is approved, even if the purge lists for the records are not approved, disposition of the AIP will happen.

In this case, it is better that, instead of rejecting the purge lists for the AIU, to instead apply holds to the records that should not be disposed.

If the purge list for the AIP is not approved, but the purge list for the records is, nothing will be disposed. This is why it is better to apply the retention to the AIP or the AIU (records) but not both.

## Table Disposition

Retention can be applied directly through either the job, via IA Web App or REST.

After the purge list has been approved, the tables in the purge list can be deleted.

A table can inherit retention from the application and, if it does, both policies must be eligible for disposition for table disposition to proceed. For example, if a five year retention policy is applied to the application, and the package has a seven year policy, the application will be skipped until the retention has elapsed for all items in the application.

There is no hardware protection for the unstructured content for a table.

The default script to create a table application ensures that there is a location for both content information (CI) and back ups. If, however, you have already created table applications, it is important to setup a database so that there is a back up store to be used by the application. If this is not done, the table type processor will not initiate the disposition process.

## Table Record Disposition

If all the records in table are eligible for disposition and approval is given, when running the DisposePurgeCandidateList job, the records will be processed. The table will not be destroyed even if all of the records were removed. In this case, a dispose audit is generated for the table. There is a configurable audit for each table row that is to be disposed.

If only some table rows in a table are eligible for disposition and approval is given, when running the DisposePurgeCandidateList job, the records will be processed and there is a configurable audit for the partial disposition of the table, as well as for the disposition of each row.

## Disposition of Tables and Table Rows

If the tables and the table rows are both eligible for disposition, a purge list will be created for the table and one or more purge lists will be created for the table rows.

If the purge list for a table is approved, even if the purge lists for the records are not approved, disposition of the table will happen.

In this case, it is better that, instead of rejecting the purge lists for the table rows, to instead apply holds to the table rows that should not be disposed.

If the purge list for the table is not approved, but the purge list for the table rows is, nothing will be disposed. This is why it is better to apply the retention to the table or the table rows but not both.

## Disposition in SIP-Based Applications

There are additional steps required to dispose of a package. After the Dispose Purge Candidate List job runs, the packages will not be destroyed. The records in the package, however, will not be returned in searches. The package's state will be updated to "Waiting for confirmation". The Confirmation job needs to run, either manually or a scheduled run. After the confirmation job runs, the next time the Dispose Purge Candidate List job runs, the packages will be removed. It should be noted that:

- Putting a hold on the package delays its removal.
- Before the DisposePurgeCandidateList job runs, the dates can be checked on the package.

To reclaim the storage for unstructured content, the Clean job must run.

## When is the Confirmation Job Required?

If you had rejected a package, and the Disposition job had marked some packages for purge, running the Confirmation job confirms everything outstanding (and that the next run of the Disposition job will cause the packages to be deleted).

## Disposition of a Tables

When a table is disposed, the metadata for the table is not destroyed.

It is important to note that a table is not automatically put under retention during ingestion. A retention policy can be applied through the IA Web App or via the Apply Retention Policy To Table job.

Only approved purge lists will be disposed. The Purge Lists tab allows the user to complete this process by reviewing the details of a purge list and clicking the associated APPROVE button.

For purge lists that apply to individual records (retention policy applied to individual records by means of a background job), the details may not be accessible on the Purge Lists tab.

There may be a couple of reasons for this:

1. The search is not ready, which can happen when changes are made to a search but the search is still in draft mode.
2. The developer has chosen to restrict the search set used when applying retention to not grant the retention manager access. If this is the case, since the retention was applied by one of two jobs, determine what the search and search set were, and allow the retention manager access.

One possible solution is to run the Remove Policy job to remove the retention policy, update the search to specify the table, and then rerun whatever job was used to apply the retention policy.

**Note:** The system will not let you update the search while it is in use by a policy or hold application.

## Disposing a Table with Records Under Hold

This section illustrates how to dispose a table while there are records under hold. The Baseball application is used in the following example:

1. [Create a fixed retention policy](#) where the date is in the past (as we want to see a disposition).
2. Create a [legal hold](#).
3. Apply the retention policy to the Master table.
4. Execute a search using the Debut Date Range Search by typing Don in the **First Name** field.
5. Apply a hold to the result set.  
Navigate to the **Background Requests** tab to wait for the hold to be applied
6. Once the hold is applied, execute the Generate Purge List job (either clone and change to manual or start schedule).  
Navigate to the **Purge List** tab for the Baseball application and you should see the Master table available for disposition.  
**Note:** It is difficult to tell if some of the records are under hold.
7. Approve the purge list.  
Even once you approve the purge list, the records you selected in step 5 will not be disposed.
8. Run the Disposition Purge List job. It is recommend that you clone the job and set to manual so you can execute it when desired. Normally, disposition is done on a schedule.  
In this case, only the items under hold will be left in the MASTER table.

## Running the Dispose Purge Candidate List Job

The Dispose Purge Candidate List job executes the disposition of approved purge lists.

It is important to have this job scheduled. Once an application is active, this job is the only way to remove content from archive.

This job must be scoped to one or more applications to run.

When this job runs, the job creates order items that can be viewed from the job instance. These are the following steps that it processed:

- Determine what needs to be processed per approved purge list.
- Create roll forward objects (for disaster recovery).
- Dispose
- Dispose update (to determine which records were disposed when either disposing or partially disposing parents).
- Backup (backup is done per processed AIP or schema).
- Dispose cleanup to clean up any intermediate objects used by disposition processing.

## Using the Purge Lists Tab

The **Purge Lists** tab allows the Retention Manager to:

- Review purge lists, which contain the items eligible for disposition.
- Approve or reject the disposition of the items contained in a purge list.
- Cancel a previously ordered approval or rejection of a purge list.
- Reject a purge (disposition) of items contained in a purge list.
- Export tables or AIPs contained in a purge list. You cannot export the records in a purge list.

When exporting a package, you must define which fields are exported. This is not tied to the export available by search and only the metadata can be exported.

In the case of tables, the raw values of the fields are exported, and any values marked as encrypted will not be exported.

The following table describes the fields for a purge list:

Column	Description
Purge List Name	The name of the purge list.
Type	Indicates the item that is contained in the purge list (for example, application, package, etc.).
Retention Policy	Indicates the name of the retention policy that was applied to the retained set.
Items	Indicates the number of items in the purge list.

Column	Description
Created Date	Indicates the date the purge list was created.
Status	<p>Status of the purge candidate list:</p> <ul style="list-style-type: none"> <li>• Under Review: The Retention Manager is currently reviewing the list to determine if the contents are eligible for disposition.</li> <li>• Approved: The items contained in the list are eligible for disposition.</li> <li>• Rejected: The list was rejected by the Retention Manager. The following actions can be performed on a purge list with the 'Rejected' status: <ul style="list-style-type: none"> <li>— Cancel Rejection: Takes the state back to 'Under Review'.</li> <li>— Cancel Purge: Takes the state back to 'Cancelled'. If the purge is not cancelled, the items in the purge list will not be eligible for new purge lists.</li> </ul> </li> <li>• Disposed: The items contained in the list were disposed.</li> <li>• Cancelled: The list was rejected by the Retention Manager. The Administrator then ran the Purge Candidate List Generation job, which cancelled the previous list.</li> </ul>

## Performing Actions to a Purge List

The available actions a Retention Manager is able to perform on a purge list depends on the Status of the list:

Status	Available Actions
Under Review	The Retention Manager is able to approve or reject the purge candidate list.
Approved	The Retention Manager is able to cancel the approval of the purge candidate list.
Rejected	<p>The Retention Manager is able to perform the following actions:</p> <ul style="list-style-type: none"> <li>• Cancel the Reject: Moves the purge list back to under review, which allows you to review the list.</li> <li>• Cancel Purge: Even if a purge is cancelled, the items will still be eligible for a new purge list. If there are items in the cancelled purge list that cannot be disposed, a hold policy should be applied to the items so they are not included in the next purge list.</li> </ul>

Status	Available Actions
Cancelled	The Retention Manager is not able to perform any actions on the purge candidate list.
Disposed	The Retention Manager is not able to perform any actions on the purge candidate list.

To perform any of the above actions:

1. Select the purge candidate list in the **Purge Lists** tab.



2. Click and select the action you want to perform on the list.

## Running the Clean Up Purge Candidate Lists and Applications Job

The Clean Up Purge Candidate Lists and Applications job cleans up purge candidate lists that have been disposed or cancelled.

The Generate Purge Candidate List job generates purge lists for records that are eligible for disposition. Once disposition has been run to dispose of the records, the purge list status will be set to disposed. The Generation Purge Lists job cancels any purge lists that have not been processed by the disposition job and will generate new lists. The Clean Up Purge Candidate List and Applications job will remove a cancelled or disposed purge list. It will depend on the customer how often they would like to run this job, as it depends on how often disposition is run. This job cannot be application scoped.

## How Long Does Exported Data Stay in the Archive?

A Retention Managers can export purge list information to make informed decisions about whether or not to approve a purge list.

The following table illustrates various scenarios and how long exported data remains in the archive.

### Retention Applied Directly to Table

In this scenario, individual tables are exported:

Scenario	Exported Background Task Removed for Table	All Order Items Removed Associated with Table	Notes
Run disposition (some records under hold)	✓		Because records were under hold, exported data must be removed from archive
Run disposition (everything can go)	✓	✓	
Cancel a purge list and run the Clean up Purge Candidate Lists and Applications job	✓		
Default interval expires and Clean up Purge Candidate Lists and Applications job runs	✓	N/A	Default interval is 1 day after the task completes.

## Retention Applied Directly to Package

For completeness, if disposition is done at the application level, all background tasks are removed after the second disposition run (first run marks content, second pass is when the audit happens):

Scenario	Exported Background Task Removed for Package	All Order Items Removed Associated with Package	Notes
Run disposition (some records under hold)	✓		
Run disposition (after confirmation, everything can go)	✓	✓	
Cancel a purge list and run the Clean up Purge Candidate Lists and Applications job	✓		The purge list is exported after disposition, this will clean it up.
Default interval expires and Clean up Purge Candidate Lists and Applications job runs	✓	N/A	Default interval is 1 day after the task completes.

## Exporting Purge Lists

You can export purge candidate lists that contain table records or AIUs that contain at least one record. The fields that are exported are controlled by the search. Only fields in the search that have been mapped will be exported, which is the same functionality as viewing purge candidate lists.

When defining the search for a table, there is a step that the search must be defined for particular table and the binding fields are set. What you see when view the purge list is what you would see when you export the values. The fields that are nested queries or encrypted will not be shown in the purge list.

Exporting a purge candidate list containing table records is an asynchronous operation. Access the exported purge candidate list in a gzipped CSV format from the **Background Requests** tab. When disposition is done, the exported purge list can no longer be viewed.

If there are no records left in a purge candidate list, the ability to export the list will not be available.

It is possible, albeit rare, that:

- You export a purge candidate list; but
- The Dispose Purge Candidate List job has run and all of the records had been processed;

The operation to export the purge candidate list will fail and an error message is issued indicating that you cannot export an empty purge list

When items are disposed, a warning is issued indicating that, after disposition runs, users will no longer be able to view or download the content. If items in a purge candidate list are exported, after disposition, the exported items can no longer be viewed:

- Disposing an entire table or package will remove the order items related to the table or package.
- Partially disposing a table or package ensures that any order items that were exported are removed. The rationale is that if the table was completely exported, and some records were under hold, the archive should not be storing information about the deleted records, which includes the rendition and the search result.
- Disposing of purge list with AIU records, the order item is removed so that users cannot download information about the disposed records.
- If the purge list for AIUs is exported, but not disposed, and the purge list is cancelled, when the Clean up Purge Candidate Lists and Applications job is run, it removes the order items exported by the purge lists (if they still exist).

## Running Reports to Check for Overdue and Upcoming Dispositions

The Reports sample application contains functionality that allows you to generate two compliance-related reports.

The reports provide the information for all installed applications, including the Reports and Audits sample applications. The reports provide counts only and are updated each time the Refresh Metrics jobs is executed. The Refresh Metrics job will only update the report information, however, after the Reports sample application has been installed.

To run the reports out-of-the-box:

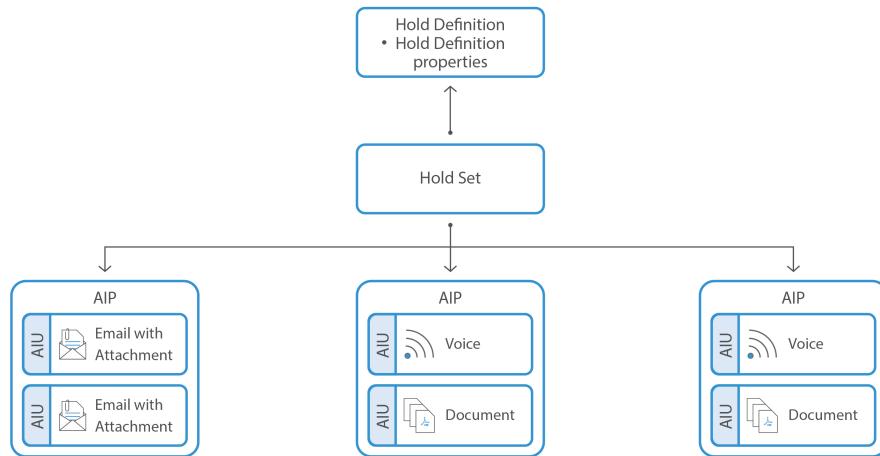
1. If the Refresh Metrics job has not been executed recently, execute the job.
2. Select the **Reports** application.
3. In the list of search sets, select either:
  - **Overdue Dispositions:** Displays the total number of records that are overdue for disposition for at least six months for archived applications. The count includes records under hold.
  - **Upcoming Dispositions:** Displays the total number of records that are scheduled for disposition for the next six months for archived applications. The count does not include records under hold.
4. Click **Search**.

By default, overdue and upcoming dispositions are displayed in graphs for each archived application. Use the drop-down list to select a specific application. While you cannot click on the bars, view the total counts by hovering the cursor over the bar.

The report results also allow you to toggle to a table view in which the same information is displayed in columns. Again, use the drop-down list to select a specific application.

## Holds

InfoArchive includes an out-of-the-box method to apply legal holds. Holds can be applied to AIPs, tables or an application and records. Once a hold is applied, the item cannot be disposed until the hold is removed. As illustrated in the following screen shot, a hold set is created when a hold is applied:



After applying a hold, the apply hold information will not be available if the search is set to draft.

There may be a number of reasons why the information is not available:

- The search is not in the Ready state.
- The search set that was used has restrictions, and the user viewing the hold set is not in one of the permitted groups.
- For a table search, the search set did not set any of the xDB element values.
- For a table search, columns marked as encrypted, masked, or nested will not be shown. If no columns are valid, the message is displayed.

**Workaround:** The Developer can update the search, correct the problem, and mark the search as Ready. There is no need to remove the hold to correct the problem.

## Using the Holds Tab

The **Holds** tab allows the Retention Manager to:

- View and edit the details of a hold.
- Create and delete a hold.

The holds managed here can be applied to applications, packages and search results by the Retention Manager. When a hold is applied, it overrides retention policies for a specified time.

Holds prevent a record from being purged, even if it is included in a retention event. A hold can be placed directly on a record or group of records for any reason (for example, legal process). A record can be associated with more than one hold at a time. If one hold is removed, the remaining holds prevent the record from being purged.

Each hold is displayed in a table that contains the following information:

Column	Description
Hold Name	Indicates the name of the hold.
Description	Provides a description of the hold.
Type	Indicates whether the hold is Legal or Permanent.
Hold Approver	Indicates the name of the person who will approve the hold.
Review Date	Indicates the date that the hold will need to be reviewed.
Requested By	Indicates the name of the person who requested the hold being created.
In Use	Indicates whether the hold has been applied.

An **Information** tab contains the custom properties of a selected hold policy.

## Creating a Hold

1. On the **Compliance > Holds** tab, click +.
2. Enter the following information:

Field	Description
Hold Name	Enter a unique name for the hold.
Description	Enter a description for the hold.
Type	Indicate whether the hold is Legal or Permanent.
Hold Matters	Enter the legal reason behind the creation of the hold.  This field is only displayed if the type of hold being created is 'Legal'.
Approved Date	Indicates the date the hold was approved.
Hold Approver	Enter the name of the person who will approve the hold.
Requested By	Enter the name of the person who requested the hold being created.
Review Date	Enter the date that the hold will need to be reviewed.
Notes	Enter any relevant hold information you want to communicate.

3. Click **Create**.

## Editing a Hold



1. In the **Compliance > Holds** tab, select the policy being edited and click .
2. Edit the fields, as desired. For further field information, see [Creating a Hold](#).
3. Click Save.

## Deleting a Hold

A hold that is associated with a legal matter should not be deleted.



1. In the **Compliance > Holds** tab, click for the hold being deleted.
2. When prompted to verify that you want to delete the selected hold, click **Delete**.

**Note:** Holds that were automatically created while creating a matter cannot be deleted. The hold will be deleted when the matter is deleted.

## Using the Hold Sets Tab

When a hold is applied to records, the records are tracked as a hold set. The **Hold Sets** tab shows the hold sets that are associated with the selected application.

Each hold set is displayed in a table that contains the following information:

Column	Description
Hold Set Name	The name of the hold set.
Item Type	Indicates the item that comprises the hold set (for example, application, package, records, table, etc.).  Records would be the value if a hold policy is applied to search results.
Created Date	The date the hold set was created.
Associated Hold	Indicates the name of the hold applied to the hold set.
Hold Type	Indicates the type of hold applied to the hold set. A hold type can be: <ul style="list-style-type: none"><li>• Legal</li><li>• Permanent</li></ul>
Items	Indicates the number of items that are contained in the hold set.
Review Date	The date the hold set is to be reviewed to determine if it eligible for disposition.

## Creating Collections

InfoArchive also provides a facility to create collections and legal matters, which translate into holds for records.

InfoArchive allows for the creation of collections of search results that can aid in the process of legal discovery. Typically, it is the E-Discovery Administrator that uses collection-related functionality. It is important to note that the E-Discovery Administrator cannot access the actual content in a repository, only the metadata of the content.

After executing a synchronous or asynchronous search for data, the user is able to save the search results as a collection. Collections can be accessed from the [Collections tab](#).

The E-Discovery Administrator is able to apply a legal hold to a collection. First, the user must assign a collection to a legal matter, which defines the legal context of the collection (for example, a specific regulatory or litigation issue). The hold continues until the matter is resolved legally. Once resolved, the matter can be closed and all the collection are released from the hold. The legal matters for each application can be accessed from the [Legal Matters tab](#).

Collections are only supported for SIP-based applications.

When creating a collection, if a filter was applied to the search results, the collection will not have the filter applied (meaning an operation on the collection will affect the original search results).

## Using the Collections Tab

The E-Discovery Administrator can create multiple collections, each one addressing a different aspect of the E-Discovery process. Each application has its own set of collections.

The **Collections** tab allows the E-Discovery Administrator to:

- View a collection
- Create a legal matter
- Apply or remove a legal matter to a collection
- Delete a collection

A table contains the name of each collection. Clicking the collection's name guides you to the **Collection Details** tab.

The **Collections** tab also provides the following information:

Column	Description
Collections Name	The name given to a collection by the E-Discovery Administrator.
Created By	Indicates the name of the user who created the collection.
Created Date	Indicates the date and time the collection was created.
Status	Indicates the status of the collection: <ul style="list-style-type: none"><li>• <b>Processing</b> indicates that an asynchronous process is in progress.</li><li>• <b>Complete</b> indicates that an asynchronous process is complete.</li></ul>
	Opens a window that displays the legal matters associated with the collection. From here, the user is able to remove legal matter(s) from a collection.
	Allows the user to: <ul style="list-style-type: none"><li>• Add the collection to a new or existing legal matter.</li><li>• Delete a collection</li></ul>

Column	Description
Collection Details	<p>A window on the right side of the screen contains further information specific to a selected collection, including:</p> <ul style="list-style-type: none"> <li>• Name: The name given to a collection by the E-Discovery Administrator.</li> <li>• Description: A description of the collection.</li> <li>• Created By: The user name of the E-Discovery Administrator who created the collection.</li> <li>• Created On: The date and time the collection was created.</li> <li>• Status: The current status of the collection (see above).</li> <li>• Records: The number of records the collection contains.</li> <li>• The <b>Collections Details</b> tab also includes the criteria used to run the search that created the selected collection.</li> </ul>
	Click to edit the name and/or description of the collection.

## Creating a Collection and Applying a Legal Matter

The following scenario illustrates the process in which the E-Discovery Administrator executes a search, creates a collection from the search results and, finally, applies a legal matter to the collection.

1. Execute a synchronous or asynchronous search.
  - a. For a synchronous search, proceed to Step 2.
  - b. For an asynchronous search, when the search request has finished processing, navigate to the **Background Requests** tab and click **View**. Proceed to Step 2.
2. Select all of the search results and click **Create Collection**.  
The **Create Collection** action ignores individual row selection using checkboxes.
3. Enter a name for the collection and, if desired, a description. Click **OK**.  
Navigate to the **Collections** tab.



4. To apply a legal matter to the collection, click the  button and select **Add to Legal Matter**.



Alternately, if the collection has not been added to a legal matter, click the  button and select **Add Collection to Legal Matter**.

5. Indicate whether you want to add the collection to an existing or new legal matter. For more information, see [Creating a Legal Matter](#).

- a. For an existing legal matter, select the legal matter from the list of available options. If desired, you can view only the legal matters that you created or search for a specific legal matter. Click **Select** and then **Save**.

- b. For a new legal matter, enter a name and description of the legal matter. Click **Save**.

The legal matter that you created will now appear on the **Compliance > Legal Matters** tab. For more information, see [Using the Legal Matters Tab](#).

The request to add the collection to the legal matter(s) runs in the background. Check the status of the request on the **Background Requests** tab.

## Using the Legal Matters Tab

The **Legal Matters** tab contains the legal matters for each application. The E-Discovery Administrator can:

- View a list of all legal matters
- Search for legal matters
- View the details of a legal matter
- Expand row to view associated collections
- Remove collection from a legal matter
- Close a legal matter
- Reopen a closed legal matter

The **Legal Matters** tab provides the following information:

Column	Description
Matter Name	Indicates the name of the legal matter.
Requested By	Indicates the name of the user who created the legal matter.
Created Date	Indicates the date and time the legal matter was created.
	Opens a window that displays the collections associated with the legal matter. From here, the user is able to remove collection(s) from the legal matter.
Collections	Indicates the number of collections associated with the legal matter.
Status	Indicates the status of the legal matter: <ul style="list-style-type: none"><li>• <b>Processing</b> indicates that the legal matter is changing, updating or involved in an update, process or communication with the server.</li><li>• <b>Active</b> indicates that the legal matter is available for use.</li><li>• <b>Closed</b> indicates that the legal matter has been closed. A legal matter can only be closed if there are no collections associated to it.</li></ul>

Column	Description
	<p>Legal matters cannot be deleted, only closed.</p> <p>Click to close a legal matter. A legal matter that has a collection associated to it cannot be closed.</p> <p>A closed legal matter can be reopened by a user with permission. Click the button to reopen the legal matter.</p>
Legal Matter Details	A window on the right side of the screen contains further information specific to a selected legal matter, including the name of the user who last modified the legal matter.

## Applying a Hold

There are different methods in which you can apply a hold to data.

To view a granular disposition scenario, where a retention policy is applied to a table and a hold is subsequently applied to individual records, refer to the Applying Retention to a [Table/Applying a Hold to Records](#) section.

If a hold is applied directly to a table, package or application that is eligible for disposition before running the Generate Purge List job, the item will not be included in a purge candidate list.

If a hold is applied to a table, package or application after the item is included in a purge candidate list, the item will remain in the purge candidate list after running disposition, but will not be disposed. The item only becomes eligible for inclusion in a purge candidate list after the hold is removed.

## Applying a Hold to an Application

1. Select the application you are applying a retention policy to.
2. On the **Application Info** tab, click **Apply Hold**.
3. Select the retention policy you want applied to the application and click **Next**.
4. Enter the following information:
  - a. Hold Set Name: Enter a unique name for the hold.
  - b. Enter a Description for the hold.
  - c. Click **Next**.
  - d. Review the information you have entered. When satisfied that the information is correct, click **Finish**.

This operation is asynchronous. Only if the operation is successful will you set the new entry in the hold sets.

The hold you applied to the application or holding is listed in the **Hold Sets** tab.

## Creating a Legal Matter

1. In the **Legal Matters** tab, click the  button.
  2. Enter a name and, if desired, a description of the legal matter. Click **OK**.
- The E-Discovery Administrator can also create a legal matter in the **Collections** tab (for more information, see [Creating a Collection and Applying a Legal Matter](#)).

## Applying a Hold to an AIP

Use the **Packages** tab to apply a hold to an AIP:

1. Select the AIP the hold is being applied to.



2. Click  and select **Apply hold**.
3. Select the hold you want applied to the application and click **Next**.
4. Enter the following information and click **Next**:
  - **Hold Set Name**: Enter a unique name for the hold.
  - Enter a **Description** for the hold.
5. Review the information you have entered. When satisfied that the information is correct, click **Finish**.

This is an asynchronous operation. The AIP will not indicate that there is a hold policy applied to it until the order completes.

## Using the REST Client to Apply a Hold

For the apply REST call for holds, it is important that, if more than one object is specified in the objectsToProtect, the type of each object is the same. The supported types include:

- Application
- AIP
- AIU (refers to a record in a package (AIP))
- Table
- Row (refers to a record in table)

The values for the objectsToProtect should come from REST calls. For example, to apply a hold to an AIP (package), pass the href for its ID.

To apply a hold to an individual record in the search results, when the search result is returned, an ID attribute is specified for each row. This is what you pass in the objectsToProtect.

When applying a hold to records, a search composition reference needs to be specified so that the record information can be shown. It is expected that the search composition matches the same search that was used to find the record IDs.

The operation will succeed even if the ID no longer is found. A recommendation is to disable the disposition job before applying holds.

## Viewing Records Under Hold

Refer to the [Using the Hold Sets Tab](#) to see how you can verify that a record is under hold.

## Using Rules to Apply Holds

Using rules to automatically apply holds allows more control over the compliance process. Using rules in this manner is similar to when a rule is used to apply retention to a record. Refer to [Using Rules to Apply Retention](#) for further information.

The sample application currently does not provide an example of this rule. It is, however, similar to the existing rule to apply retention, which is defined in the PhoneCallsGranular application.

Review the <INFOARCHIVE\_ROOT>/examples/applications/PhoneCallsGranular/config/rules/rule-applyRetentionRule file to learn how to define the rule's metadata. Refer to [Using Rules to Apply Retention](#) for further information.

On the **Administration > Jobs tab**, edit the Apply Hold Rule to Records job or create a new job based on the Apply Hold Rule to Records job. Create multiple jobs to use different rules.

If you execute the Apply Hold Rule to Records job, and no criteria is specified, the rule job will end up caching in every AIP. You may not wish to run the job on a schedule unless absolutely necessary. This applied to searches in general, as well as the Apply Retention Rule to Records, Apply Policy To Records job, and the Apply Hold Rule to Records job.

Enter the following information:

1. Ensure that the **Apply To** field is set to a specific application. In order to automatically apply a hold using a rule, the job must be scoped to a specific application.
2. In the **Properties** section:

Field	Description
<b>ruleName</b>	Specify the rule to be used for this particular job.
<b>searchName and searchSet</b>	Specify the name of the search or the search set (search composition). Once the job is executed, the search results will then be passed into the rule, which automatically applies the hold.

Field	Description
<b>searchCriteriaFile</b>	Specify information to narrow the scope of the search. You can either: <ul style="list-style-type: none"> <li>Narrow the search scope within the rule itself by selecting criteria and doing evaluations based on the column data that is passed; or</li> <li>Enter the actual search criteria. For example:</li> </ul> <pre>&lt;data&gt;&lt;/data&gt;</pre>
<b>developerMode</b>	Set to <i>true</i> to test the rule. This will output the results of the rule but it does not apply the hold to the object. Essentially, a test helps to ensure that the data returned in the search is correct before actually running the rule. Set to <i>false</i> if you do not want to test the rule.

3. Run the job. All of the records returned in the selected search or search set will be put on hold. Navigate to the **application > Hold Sets tab** to view the newly created hold set.

The rule itself looks similar to the apply retention rule:

```
package com.emc.ia.retention.rules

//list any import classes here.
import com.emc.ia.retention.rules.beans.ApplyHoldBean;
import com.emc.ia.retention.rules.beans.AiuRecordBean;

//declare any global variables here

rule "Apply hold to trades"
when
 $aiuRecordBean:AiuRecordBean();
then
 ApplyHoldBean $applyHoldBean = new ApplyHoldBean();
 $applyHoldBean.setHold("Hold Trades");
 $applyHoldBean.setNewHoldSet(false); // share hold sets
 $applyHoldBean.setHoldSetName("Trades120 : " + $aiuRecordBean.getRecordRows() .
 get("CreatedDate"));
 $applyHoldBean.setId($aiuRecordBean.getId());
 insert($applyHoldBean);

end
```

It is possible to use the PhoneCallsGranular sample application to test how this functionality works. The following scenario can also be applied to the Apply Retention Policy to Records job.

When applying the following scenario to your own applications, ensure that the `searchName` and `searchSet` properties are set to search that is in Ready status. Furthermore, the search criteria needs to match your search (and can be used to reduce the set of records to evaluate against the rules).

**Tip:** Run the search through the IA Web App and then view the payload for the search command (the search criteria is in an XML format).

- In the **Administration > Jobs** tab, click  to edit the Apply Hold Rule to Records job.
- Enter the following information:

- a. In the **Apply To** field, scope the job to the PhoneCallsGranular application.
- b. In the **Properties** section, enter the following:
  - i. Leave the ruleName blank.
  - ii. Set searchName to FirstName\_Operator.
  - iii. Set the searchSet to Set 1
  - iv. Set the seachCriteriaFile to <data></data>.
  - v. Leave the developerMode as the default value of false.
- c. Click **SAVE**.
- d. Execute the Apply Hold Rule to Records job.

The Apply Hold Rule to Records job creates two distinct hold sets based on the rules.

If applying the same scenario while with the Apply Retention Policy to Records job, the job creates eight retained sets (two objects will have retention applied without creating a set).

## Removing Retention

### Removing Retention from an Application

1. Select the application.
2. In the **Application Info** tab, click  for the retention set being removed.
3. When prompted to verify that you want to remove the selected retention set, click Remove.

### Deleting a Retention Policy

You cannot remove a retention policy that is currently in use. To determine if a retention policy is in use:

- The retention policy is applied to an item
- An application refers to the retention policy through the [Default Retention Policy](#).
- A retention class (defined with holdings) refers to the retention policy.

**Tip:** If you are using a job to apply retention to records, remember to change the name of the retention policy in the job to a new value. There are two jobs (the original and the rule based one).

1. In the **Compliance > Retention Policies** tab, click  for the retention policy being deleted.
2. When prompted to verify that you want to delete the selected retention policy, click Delete.

## Running the Remove Policy Job

The Remove Policy job removes the named retention policy from items in an application. The job can be limited to a type of retention policy.

Only one retention policy can be specified and is case-sensitive.

The following parameters can be configured:

Parameter	Description
<i>retentionPolicyName</i>	Name of retention policy that will be removed from items. Retention policy must be defined and name is case sensitive.
<i>type</i>	<p>Optional field to specify the type in which the job will remove the policy. Possible values include:</p> <ul style="list-style-type: none"> <li>• application</li> <li>• aip</li> <li>• aiu</li> <li>• table</li> <li>• row</li> </ul> <p>Row refers to a table row.</p>

## Removing Holds

### Viewing Items in a Hold Set

1. Select the application in which the hold set is stored.



2. In the **Hold Sets** tab, click  for the hold set you want to view. The information displayed depends on the Type of hold set:

Item Type	Description
Application	The application's properties are displayed.
Package	A list of packages is displayed.
Table	A list of tables is displayed.
Records	A list of records is displayed. The records displayed are based on the results of a primary search configuration. The results of a nested search configuration are not displayed.

## Removing an Item from a Hold Set

This procedure allows the user to remove a hold policy from records. Use the following procedure to remove selected items from the hold. If you want to remove all of the items from the hold set, delete the hold set.

If the items were put inside the hold set included in a collection, you will not be able to remove the hold set or items from the hold set. Instead, remove the collection from the legal matter to remove the hold.

1. Select the application.



2. In the **Hold Sets** tab, click .
3. The records under the hold set are displayed. Select the record or records you want removed from the hold set.
4. Click Remove Hold.
5. A pop-up is displayed that indicates the name of the hold set, the number of records being removed from the hold set. Enter the reason why the hold set is being removed from the records and click Remove.
6. Because this is an asynchronous operation, check the **Background Requests** tab for the status.

## Removing a Hold from an Application

1. Select the application.



2. In the **Application Info** tab, click for the hold set being removed.
3. When prompted to verify that you want to remove the selected hold set, click Remove.

This operation is asynchronous. It can, however, also be performed from the **Hold Set** tab where the operation would be synchronous.

## Removing a Collection from a Legal Matter

When a collection is removed from a legal matter, the collection is no longer under a legal hold. Only a E-Discovery Administrator is able to remove a collection from a legal matter.



1. On the **Collections** tab, click the button.  
A window displays all of the legal matters associated with the collection.
2. Select the legal matter(s) being removed.
3. Click **REMOVE COLLECTIONS FROM SELECTED LEGAL MATTERS**.

4. When prompted to confirm the removal, click **Remove**.

The request to remove the collection from the legal matter(s) runs in the background. Check the status of the request on the **Background Requests** tab.

Alternately, the same procedure can be completed from the Legal Matters tab:



1. Click the button.  
A window displays all of the collections associated with the legal matter.
2. Select the collection(s) being removed from the legal matter.
3. Click **REMOVE SELECTED COLLECTIONS FROM LEGAL MATTER**.
4. When prompted to confirm the removal, click **Remove**.

The request to remove the collection from the legal matter(s) runs in the background. Check the status of the request on the **Background Requests** tab.

## Deleting a Collection

A collection can only be deleted if it is not associated with a legal matter. Deleting a collection removes the hold on all items in the collection, unless the item is also part of another collection.



1. In the **Collections** tab, click the button for the collection being deleted.
2. When prompted to confirm the deletion, click **Delete**.

## Metrics

### Performing a Byte Count on Application Data

InfoArchive includes a standalone Java tool to calculate file size and character count for tables, XML files or SIP.zip files in an application. The tool automatically chooses a unit (bytes, KB, MB, etc.) in which the smallest file has a value greater than 1. All files have the same unit for easy comparison. Formatting data in a file is not included in the final calculation.

Access the tool in the **tools\metrics.bat** directory.

The following parameters are required to perform a byte count:

Parameter	Description
Use one of the following: • <i>-xml</i> • <i>-sip</i> • <i>-table</i>	The <i>-xml</i> parameter is used to calculate the byte size of either an XML file or table.  The <i>-sip</i> parameter is used to calculate the byte size of a SIP.zip file.  The <i>-table</i> parameter is used to calculate the byte size of the XML and the size of the attached content.  You must enter one of the above parameters.
<location>	You must specify a directory or file that the byte count is performed on.
<i>-csv &lt;csv-location&gt;</i>	While it is not mandatory, you can generate a report that contains the information returned in a byte count. The report is in .CSV format. The <i>-csv</i> allows you to specify the name of the report.

## Usage Examples

To perform a byte count on a table named `BASEBALL-MASTER-01.xml` that is stored in the Baseball application, the following command is used:

```
%INFOARCHIVE_ROOT%\infoarchive>tools\metrics.bat -xml applications/
Baseball/tables/BASEBALL-MASTER-01.xml
```

When performing a byte count using the *-xml* parameter, the following information is returned:

Column	Description
File size	Indicates the compiled data size of the table or XML file.
Data size	Indicates the character count of the table or XML file.

To perform a byte count on all of the tables stored in the Baseball application, the following command is used:

```
C:\projects\IA\infoarchive>tools\metrics.bat -xml applications/
Baseball/tables
```

To perform a byte count on all of the tables stored in the Baseball application and generate a report called `baseball.csv`, the following command is used:

```
C:\projects\IA\infoarchive>tools\metrics.bat -xml applications/
Baseball/tables -csv baseball.csv
```

To perform a byte count on the SIP.zip files stored in the Trades application, the following command is used:

```
C:\projects\IA\infoarchive>tools\metrics.bat -sip applications/
Trades/sips
```

When performing a byte count using the `-sip` parameter, the following information is returned:

Column	Description
File size	Indicates the size of the <code>eas.pdi.xml</code> file.
Data size	Indicates the character count of the <code>eas.pdi.xml</code> file.
Content size	Indicates the byte count of any additional files (for example, images) stored in the <code>SIP.zip</code> file.
Sip size	Indicates the compiled data size of the <code>SIP.zip</code> file.

## Understanding the Compliance Dashboard

The compliance Dashboard contains the following information:

- The percentage of the system under retention. This is based on the number of records. The value will be between 0 and 100.
- The percentage of the system under retention. This is based on the number of records. The value will be between 0 and 100
- The number of applications that have no records under retention.
- The number of applications that have some, but not all records under retention.
- The number of applications that have all records under retention. Generally, the expected value if a default retention policy has been applied and at least one records has been ingested.
- The number of records that have their projected disposition in the next 6 months. Each month assumes that the previous purge will be destroyed. Holds are taken into consideration.
- Represents the top hold by application. Only the top 5 are given. This value could be empty if no holds have been applied.
- The storage footprint is displayed per application:
  - For SIP archiving, the XML metadata size is the sum of the xDB library segment file size of every single AIP.
  - For table archiving, the XML data metadata size is the sum of the xDB library segment file size of every single table schema:
    - The index size is the sum of the index size of all AIP/Schema xDB libraries.
    - The content size is the sum of all associated content of a table/AIP (reversibility formats, CI container, RI XML, SIP XML, xDB Library back up, etc.).

**Note:** The disposition projection for storage will not include the information for the Audit application.

## Running the Refresh Metrics Job

The Refresh Metrics job updates the metrics for the Dashboard.

The job must be scoped to the system and does not support application scoping.

There are three scenarios in which the information will not be available from the server and the UI will show an empty list of applications:

- If the customer upgrades the system but does not refresh the metrics.
- If the customer did not install any application and refreshed the metrics.
- The metrics were never refreshed.

Calculating the metrics information in the InfoArchive Dashboard can take a significant amount of time. Therefore, the Dashboard retrieves most of its information from pre-populated values and the Refresh Metrics job populates these values. The job scans the system and populate the metrics information. The customer can decide how often the metrics information should be updated, as it would depends on their individual use cases.

This job does not have any parameters.

## Audits

### Using Audits for Compliance

InfoArchive allows the following audit events:

- Registering an audit: An event can be turned on for auditing (for example, the customer wants to control audit generation for unsuccessful login attempts).
- Creating an audit: The created audit tracks:
  - The name of the user who performed the action.
  - The name of the object involved in the action.
  - The time the operation took place.
- Viewing audit information
- Searching an audit: This allows the customer to look for specific audit events (for example, the customer wants to view any unsuccessful login attempts).

There is core information that is part of every audit. Additionally, there is event-specific information added to an audit.

### Metadata Fields

Field	Type	Mandatory	Comments
id	UUID	Yes	Automatically generated by the system
eventType	String	Yes	Event type for the audit (for example, application, AIP, table, login, retention policy, etc.)

Field	Type	Mandatory	Comments
eventName	String	Yes	Event Name (for example, view, create, edit, Delete, ingest, export, dispose, etc.).
eventSource	String	No	Additional context data indicating where the audit event took place (for example, dispose could be from the job or a user-defined action).
auditedObjectId	UUID	No	Unique identifier that is associated to the object being audited (for example, URI of the retention policy).
applicationName	String	No	Name of the application that the audit may be part of.
applicationId	UUID	No	ID of the application that the audit may be a part of
tenantId	UUID	Yes	Tenant that this audit is a part of.
createdby	String	Yes	User who created this event. The system populates this field when the audit is created.
createdDate	Date (GMT)	Yes	The date the audit was created. The system sets this field when the audit is created
SupplementalData	SupplementalData	No	<p>Describes any additional data fields that the customer wants added to the audit event. Specific to the audit event type and include information the caller deems necessary.</p> <p>The implementation of each audit event is done by each process that utilizes the audit system.</p>

## Audit Event Type

The audit event type is set to enable or disable audits for a particular audit action/process. If there is no audit configuration set for a particular audit, then auditing is not enabled for that particular event.

InfoArchive includes a fixed list of audits stored in a constants file and uses a REST to retrieve the audits. Customers can create their own audits using IA Shell.

## Application-Specific Audits

InfoArchive provides the ability to enable specific audits on an application basis. When creating an application name, the event type is optional. If the event type it is not set, then the audit is for all applications that include that particular event type.

If the audit is specific to an application, and the application name is passed, the system checks for that audit event combination. If there is no event type that exists, the system checks for an event type without the application name to use.

## Metadata Fields

Field	Type	Mandatory	Comments
id	UUID	Yes	Automatically generated by the system.
type	String	Yes	Object/Event type for the audit (for example, login, logout, application, AIP, table, hold, retention policy, etc.).
name	String	Yes	The event name (for example, create, edit, delete, dispose, etc.).
applicationId	UUID	No	ID of the application that this audit may be a part of
TenantId	UUID	Yes	Tenant for this event type.
enabled	String	Yes	Defines whether the audit is enabled.

## Searching for Audits

Audits are archived into the system as SIPs/AIPs., which allows the customer to perform searches of the audit entries.

There is a default search form and result list.

The search forms have been enhanced to leverage the new filter functionality and the default sort is a descending sort on the created date.

## Audit Entries

Audits can be configured through the InfoArchive interface to change the default configuration to track a specific event. If too many audits are enabled, however, performance will be impacted.

Three levels of audits exist:

- System
- Tenant
- Application

The following table describes the audit type than can be enable in the product for specific actions (see audit name). The event source is the source that can generate the audit.

The following compliance actions are for events based on retention:

Name	Description	Notes
Apply	Audit for when a hold or retention policy is applied	Supplemental data contains the name of the set (hold set or retained set)
Approve	Audit for when a purge list is approved for disposition	
Cancel	Audit for when approval is cancelled (allowing items in the list to be put into new purge lists)	This audit is not generated if the purge list is cancelled by the Generate Purge List job.
Close	Audit for closing	Specific to matters, a closed matter cannot have collections associated to them.
Dispose	Audit based on type for when a resource is disposed	This audit can be turned on per type.  Individual audits are not generated for AIUs or table rows when the AIP or table is disposed.
Generate	Audit for when a purge list is created	Supplemental data should include information about retention policy and the type of item put into the purge list.
Open	Audit for opening	Specific to matters.

Name	Description	Notes
Partial Dispose	Audit for if a resource is partially disposed.	This audit can be turned on per type. For packages (AIPs), this means that some records inside were removed.  Whenever a partial disposition is done, a backup is taken. In the case of the table, the backup is done at the schema level. A backup store must be defined for the application (the sample applications demonstrate how this is done). For each database or holding, a backup store needs to be defined for retention. This same store can be used for the packages and retention (stores are associated with each application).
Remove	Audit for when something is removed.	Audit is specific to the retention policy or hold.
Re-qualify	Audit for when re-qualification was done for a retention policy	This audit is per retention policy and indicates that re-qualification was done for a particular retention policy. If the re-qualification job is run, this audit is generated for every retention policy.
Revoke	Audit for when approval is revoked or if the rejection was removed	

## Using the Archive Audit Job

The Archive Audit job exports in memory audit information into SIPs so that they can be searched in the Audit application.

This job requires the audit application to be installed. After the job runs, the audits are purged so REST calls to fetch the audits will not return the archived audits.

When an audit object is created in the system, you cannot execute a search against it. The objects are collected in a temporary storage until they are archived. The Archive Audit job must be executed to allow audit objects to be searched.

Audits are organized in SIP packages by day (for example, only audit for the same day will be put in a SIP). This allows you to apply a retention policy to the archived audits and allow for proper disposition of audits.

The following parameters can be configured:

Parameter	Description
<i>MaxSipAuditEntries</i>	This is the maximum number of audit entries that will be in a SIP package. The default is 50,000 entries.
<i>Start Date</i>	Specify a date to start selecting audits from. Leaving this date empty will default to today's date. The format is YYYY-MM-DD.
<i>End Date</i>	Specify a date to end selecting audits. Leaving this field empty will default to today's date. The format is YYYY-MM-DD.

The start and end date are used to allow for archiving audits that are not current.

The normal operation is to leave the date empty and allow the job to archive all audits for the day. The job will also go back 30 days searching for audits to archive. It will stop once it finds that there are no audits to archive for a day.

When running this job, the audit for apply retention for the audits log will be archived in the next run.

This job should not be scoped to an application.

# Compliance Troubleshooting

Issue	Resolution
For SIP searches, an exception is issued when trying to apply a hold or retention policy to all or some of the items in the search results.	<p>The application is not correctly configured. The AIU ID is put on the index and not on the operation element.</p> <pre data-bbox="878 481 1405 650">&lt;data&gt; &lt;id&gt;pdi.aiu.id&lt;/id&gt; &lt;select.query&gt; declare namespace op = "urn:acme-corp:xsd:sdd:operation .1.4"; /op:operations/op:operation /op:index &lt;/select.query&gt; &lt;/data&gt;</pre>
<p>The ingestion log contains the following message:</p> <pre data-bbox="274 1072 812 1163">The retention date is not applied for AIP of format eas_retention_date in store eas_retention_date.</pre>	EAS does not push a retention date to storage for xDB back up, as the back up is changed each time a partial disposition is done.
If tables are skipped during disposition	<p>Check the logs for the following error message:</p> <pre data-bbox="866 1241 1388 1332">Cannot dispose Table [name: {}, id: {}]: Missing xdbStore (backups) on database [name: {}, id: {}]</pre> <p>If this message is logged, setup a database so that there is a back up store to be used by the application. If this is not done, the table type processor will not initiate the disposition process.</p>

Issue	Resolution
The following error message is issued when a search is executed: "Internal Server Error: no xDB library associated with search dataset."	This message is issued when the Administrator has deleted the application data. Either delete the search or run the Ant script to re-ingest the application and searches. The search should automatically be corrected. Conversely, run the Ant script to redefine the table (and not re-import the search), but then the Developer will need to modify the search to re-associate the archival collection (database), schema, and table.
The following error message is issued when you try to approve a purge candidate list: "A conflict occurred. List cannot be approved from state CANCELLED".	<p>While this message is rarely generated, if it is issued it means either:</p> <ul style="list-style-type: none"> <li>• Another Retention Manager rejected and then cancelled the purge candidate list; or</li> <li>• The Generate Purge List job is running too frequently.</li> </ul>

If a hold is placed on SIP or table data, users cannot search for information in the hold set if the search is not in a Ready state. If a hold is placed on data, the Developer must ensure that any applicable searches are in the Ready state. There is no need to remove the hold to correct the problem.

- When an invalidated/rejected AIP goes through disposition (assuming the retention was applied to package), if the user tries to export the purge list (the package), the export will not contain any records.
- If a package is invalidated or rejected, and granular retention was used on the records, those records will not be seen in the purge candidate list.
- When the Generate Purge Candidate List job is executed and detects records that would have gone into a purge list, but are in a cached-out AIP, those records will not be put into the purge list. Instead, a request is made to cache-in the AIP. The next time the Generate Purge Candidate List job runs, assuming the cache-in has completed, the records will be added to a purge list.

# Chapter 8

---

## InfoArchive for the End User

### Overview

InfoArchive is a powerful, secure, and scalable archiving solution. It preserves, maintains, and controls continuing access to valuable enterprise information assets.

InfoArchive allows you to access applications that, essentially, act as archives of data that have been ingested. Access an application to execute a search to retrieve the desired information and export the search results.

There are two types of applications in InfoArchive:

- Applications that contain data extracted from a decommissioned application: Old and outdated legacy applications are costly to maintain. Sometimes they are no longer officially supported, often run on old hardware nearing the end of its lifecycle, and contain data that is no longer being updated. The historical data in a legacy application, however, must sometimes be retained for business reasons, as well as to comply with various legal and regulatory requirements. InfoArchive allows data to be extracted from such a legacy application, transformed into a readable format and ingested, allowing you to execute searches against the archived data as well as export search results.
- Applications that contain data from other active applications. Data was extracted from a live system and ingested into InfoArchive to lessen the load on the company's system and, therefore, improve overall performance, as well as reduce the hardware requirements.

Applications in InfoArchive are either table- or package-based archives. The primary difference between the two archives is the manner in how data was ingested into InfoArchive.

A table application contains data that was extracted from tables, such as spreadsheets (structured data), transformed into XML before being ingested into InfoArchive. While table applications primarily contain structured data, they may also contain unstructured data. For example, after executing a search trying to locate a specific customer, the search results may contain a link to a PDF copy of the customer contract (unstructured content).

A package application, on the other hand, contains data that was extracted from files, such as Microsoft Office documents, print streams, image files, and videos. For example, a customer record may include contact information (structured content), a picture of the customer (unstructured content), transactions (structured content), and a contract (unstructured content). For example, a customer record could include contact information (structured content), a picture of the customer (unstructured content), transactions (structured content), and a contract (unstructured content).

# The Applications Landing Page

After logging in, the **Applications** page is displayed, and contains all of the applications that you are permitted to access. Simply click the application to select it and access the available search forms.

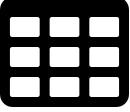
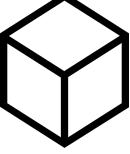
Applications can be viewed in two different ways. Toggle between the two views by clicking one of the following buttons located on the right side of the page:

	The default view that allows you to view a list of available applications in a vertical grid.
	Allows you to view the available applications in a series of cards. This view provides more information about each application than the list view.

Both views contain the following application information:

- The name of each available application.
- A description of the contents of each application's archive.
- The designated category for each application.

Each InfoArchive application displayed in a list or card also contains several icons. These icons indicate each application's type (table or package), as well as whether an application is in test or active mode:

	Indicates that the application contains data extracted from a decommissioned legacy application.
	Indicates that the application contains data extracted from an active application.
	Indicates that the application contains data that was extracted from tables.
	Indicates that the application contains data that was ingested as packages, such as Microsoft Office documents, print streams, image files, and videos.
	Indicates that the application is in test status, which allows a customer to test the application, typically, with fake data.

## Finding an Application

If dozens of applications have been created, you may have to navigate to a specific page of applications to find the one that you are looking for. Use the **Find** field to quickly locate a specific application.

	<p>Locate an application by either:</p> <ul style="list-style-type: none"> <li>• Entering the name or partial name of the desired application; or</li> <li>• Entering keywords that may appear in the application's description.</li> </ul> <p>Once your search criteria has been entered, press <b>Enter</b> on your keyboard. Applications are displayed based on the specified search criteria.</p> <p>To clear the field and return to the <b>Applications</b> page, click the <b>X</b> that appears on the right side of the <b>Find</b> field.</p>
-----------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Switching the Language of the User Interface

InfoArchive can be displayed in the following languages: Chinese, Dutch, English, French, German, Italian, Japanese and Spanish.

While you may select a language in which to view the InfoArchive interface when you initially log in, you may also select a language after you have logged in. You can switch the interface language from any page in InfoArchive.

1. Click your user name that appears in the top-right side of the page and select **Language**.
2. Select the desired language from the list and click **Select**.

The interface now appears in the selected language.

**Note:** Applications and search forms appear in the original language that they were configured with.

## Searching

### Overview

Each InfoArchive application contains a set of searches that allow you to retrieve the desired information. Access an application to execute a search to retrieve the desired information and export the search results.

Search forms are created by a developer, and provide a rich graphical interface for both entering search criteria and viewing search results. Depending on the data stored in a particular application's

archive, a search form may include such items as text fields, drop-down lists, date pickers, radio buttons, lists, etc.

The available search forms for an application appear on the left-side of the page. The currently selected form is highlighted and appears in the search form pane.

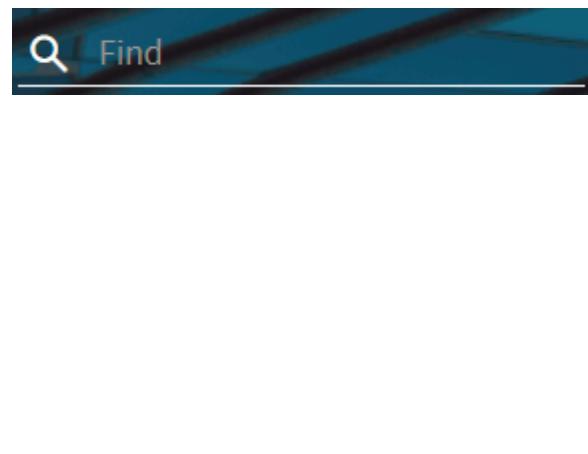
Each search may contain one or more search sets. The Developer can create different search sets, with each set having slightly different behavior in the actual search form and/or query and/or result. The Developer uses permissions make to make each search set available to different groups.

All searches appear in the panel, regardless of whether the user can access the search or not.

When a search is in Draft mode, it is only visible to users with the Developer role

## Finding a Specific Search Form

If dozens of searches have been created, you may have to navigate to a specific page of searches to find the one that you are looking for. Use the **Find** field to quickly locate a specific search form.

	<p>Locate a search form by:</p> <ul style="list-style-type: none"><li>• Entering the name or partial name of the desired search form; or</li><li>• Entering keywords that may appear in the search form's description.</li></ul> <p>Once your search criteria has been entered, press <b>Enter</b> on your keyboard. Search forms are displayed based on the specified search criteria.</p> <p>To clear the field, click the <b>X</b> that appears on the right side of the <b>Find</b> field.</p>
------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Entering Information in a Search Form

The fields available on a search form depend on which elements the developer included in the form when it was created.

Certain fields require you to enter search criteria, such as text fields, while other fields require you to select search criteria, drop-down lists and checkboxes.

The search developer has a variety of tools that can be included in a search form to help you enter search criteria.

For example, the search developer may have also included instructions in a field to help you complete the search form. In the following screenshot, the developer included the format you would use to enter dates in the **From** and **To** field:

Tooltip text can also be added to a search form. Hover the mouse over a field to view instructions about how to complete a specific field:

The following subsections provide instructions and tips when entering information in specific fields.

## Entering Dates in a Search Field

You will sometimes be prompted to enter a date or a date range to retrieve data. You can always enter a date manually, especially if the search developer has included instructions or tooltip text outlining the format you are to enter a date.

You can also click the field to select a date:

A calendar is displayed. You cycle back and forward either by month or by year:

	<p>Click to cycle back a month. In the desired month, click the date you want to enter as search criteria.</p> <p>To cycle back by year, click the month and year (for example, <b>January 2018</b> in the screenshot above) and then press this button to cycle backwards. In the desired year, click a month and then the date you want to enter as search criteria.</p>
	<p>Click to cycle forward a month. In the desired month, click the date you want to enter as search criteria.</p> <p>To cycle forward by year, click the month and year (for example, <b>January 2018</b> in the screenshot above) and then press this button to cycle forwards. In the desired year, click a month and then the date you want to enter as search criteria.</p>

## Entering Times in a Search Field

You will sometimes be prompted to enter a time or a time range to retrieve data. You can always enter a time manually, especially if the search developer has included instructions or tooltip text outlining the format you are to enter a time.

You can also click the field to select a time:

	Click to scroll the time up by the hour, minute or second. Select the desired number.
	Click to scroll the time down by the hour, minute or second. Select the desired number.

Select whether you want **AM** or **PM**.

## Using Multiple Values in a Search Field

The search developer can also configure a search form field to allow you to apply search criteria to multiple fields. For example, a text field might allow you to enter the name John to search as a first, middle or last name.

The following screenshot illustrates how a value entered in the **Employee Name** field can be used to search for a first, middle and/or last name:

The screenshot shows a search interface. At the top, there is a text input field labeled "Employee Name". Below it is a dropdown menu labeled "Search in" with options: "First Name, Middle Name, Last ...". A sub-menu is open, showing three checked checkboxes: "First Name", "Middle Name", and "Last Name".

Deselect a box to exclude it from the search criteria.

## Using Operators in a Search Field

The search developer can also configure a search form field to allow you to apply an operator to the search criteria.

An operator is, essentially, a parameter that you can apply to a search field to narrow the scope of your search. There are three types of operators a search developer can apply to a field:

- Number Operators: The following screenshot shows the number operators available to a user looking to retrieve a specific invoice number:
- String Operators: The following screenshot shows the string operators available to a user looking to retrieve a client's first name:
- Date Operators: The following screenshot shows the date operators available to a user looking to retrieve an invoice:

To use an operator, select it from the list and then enter the search criteria.

# Executing a Search

## Filling a Search Form

To execute a search, complete as many of the available fields as required and click **SEARCH**.

When the search results will be available depend on whether the search is run in the foreground or background.

When a search runs in the foreground, this is referred to as a synchronous search. Search results immediately appear in the **Record Search** tab.

If a search, however, takes longer than eight seconds to return results, the system will prompt you to either continue the search in the background or cancel the search. If you continue the search in background, another dialog presents you with a name for the asynchronous search, which can be updated. Once satisfied, click **OK**.

**Note:** Eight seconds is the default time limit set in the server. Your company's administrator may have changed the time limit to be longer or shorter.

The results of an background (or asynchronous) search can be picked up from the **Background Requests** page. Refer to [Background Requests Tab](#) for further information.

## Running a Search in the Background

After completing the required fields in a search form, you can opt to run a search in the background. This is also referred to as a asynchronous search.

To deliberately run a search in the background:

1. Click the search arrow and select **Run search in background**:

- A dialog appears with a default search name.
2. If desired, update the search name.
3. Click **START BACKGROUND REQUEST**.

The results of a background search can be picked up from the **Background Requests** tab once the search has completed. For more information, refer to [Background Requests Tab](#).

## Viewing Search Results

If a search is executed synchronously, the results are presented as a series of columns that the Developer selected during the search's composition.

The screenshot shows a search interface for 'Baseball'. At the top, there's a navigation bar with a magnifying glass icon, the word 'Baseball', 'Record Search' (which is underlined), and 'Application Info'. Below this is a button labeled 'Select all 88'. The main area displays a table of search results with columns: Last Name, Birth Year, Birth Month, and Birth Day. The first row is highlighted. To the right of the table is a 'Details' panel containing fields like 'Base Date', 'Projected Disposition Date', 'Retention Policy', 'On Hold', 'Given Name', 'Debut', and 'Final Game'. Navigation controls at the bottom include arrows for page navigation and a page number field set to '5 of 9'.

Last Name	Birth Year	Birth Month	Birth Day
Miller	1985	5	21
Miller	1875	10	28
Miller	1879	5	23
Miller	1894	8	30
Miller	1910	4	12
Miller	1927	7	26
...			

◀◀
◀
▶
▶▶
Page
5
of 9
▶
▶▶

Depending how the search developer configured the search, the results may contain text, a link to a nested search, as well as downloadable or viewable content.

To navigate to a specific page, enter the page number in the field and click **Enter**.



The user is also able to navigate the search results by using the arrows:

	Return to the first page of search results.
	Navigate backwards one page.
	Navigate forwards one page.
	Navigate to the last page of search results.

## Using a Filter to View Search Results

When a search developer adds filters to a search, it helps you user narrow down the search results. You can combine multiple search criteria and use logical operators (Match All and Match Any, or combine multiple filters).

The search developer can allow a filter to be used on a single column or multiple columns in the results of a search.

To use a filter to view search results:

- Once a filter is enabled for at least one column of search results, a switch to filter the results is displayed. If none of the columns include a filter, the switch will not be displayed. Turn the switch on:



An additional row under the table header is displayed that includes at least one filter icon. The symbol indicates that a particular result column can be filtered:

To navigate back to the original search results, turn the filter switch off .

2. Click the filter icon or anywhere in the cell under table header.  
A popup is displayed.
3. Choose operators to apply to the filter. These operators are available based on the data type associated with that column. For encrypted columns, only two operators are available (Equals and Not Equals). Enter the filter criteria and click **GO** or **ENTER**, depending on the widget being filtered.

The filter is also executed when you change the MATCH ALL/MATCH ANY selector value.

To clear the cell and execute the filter again with the remaining values, click **X**.

For example, you want to filter the column of search results that contain the last names of different customers. You are looking for the customer Paul Abbott. You can select the operator for **Exact Match** and type in 'Abbott' or select the operator for **Begins with** and type 'A'.

The filter operators that are displayed depend on the type of data that will appear in the column. These values may include:

- Number operators
- String operators
- Date operators

If the column contains date-related data, select the desired filter operator and click anywhere on the input field with the date calendar to launch the date picker. Date-related operators also include the operator **Between**, which allows you to type in or select a start and end date to filter the search results. The following data types have **Between** as operator:

- Date
- DATETIME
- NUMBER

Repeat this step if you want and are able to filter multiple columns.

For string comparison (when a the column data type is **String**), you can execute case-sensitive or insensitive exact matches. Some operators, such as **Begins With**, **Contains** and **Ends With** are not case-sensitive.

You can look for multiple values in one column, as well. For example, if you are not sure of a customer's first name, and if the **First Name** filter is enabled in the search result column, enter a comma-separated list of first names. For example, entering **Dan, Don, James** will display all entries whose first name equals Dan, Don or James.

4. Click **GO**
5. Repeat steps 2 and 3 if you want to apply multiple filters to the search results.

Once a filter is applied, click **X** to remove the filter from the set of search results.

## In-Line Panels

Depending how the search developer configured a particular set of search results, further details may be contained in an in-line panel of the search's results.

To view the in-line information for a set of search results, click the button indicated by the red arrow:

Invoice Date	Customer Number	Customer Name	Invoice Number	Ship Date	CID
1995-04-24	LKP49	Ahold	000020	2000-04-02	<a href="#">Download</a>
1995-07-24	X3L50	Coca-Cola	000009	2002-08-02	<a href="#">Download</a>
1996-09-01	D10T9	Mirrorsoft	000001	1997-09-16	<a href="#">Download</a>
1997-07-19	ITPH2	Sorcim	000003	1998-01-17	<a href="#">Download</a>
1997-09-07	9H5IH	SCR Adventures	000004	1995-07-08	<a href="#">Download</a>
1997-09-21	NQ5HM	Aegis Developments	000002	2001-06-25	<a href="#">Download</a>

**Details**

Base Date	Dec 12, 2013 11:00:53 AM
Projected Disposition Date	Mar 11, 2014 12:00:53 PM
Retention Policy	Yes
On Hold	No
Longest Retention Policy Name	Invoices-1-policy
Longest Retention Source Type	Package
Longest Retention Event Based	No

The in-line information is displayed. Use the scroll bar to view the entire set of in-line information:

Invoice Date	Customer Number	Customer Name	Invoice Number	Ship Date	CID
1995-04-24	LKP49	Ahold	000020	2000-04-02	<a href="#">Download</a>
<b>detail</b>					
Item Description	Item Number	Ordered Amount	Quantity	Unit Price	
16oz. Insulated Stainless ...	5426	75.	19	3.99	
Acropolis - Men's Watch	2167	2940.	84	35.00	
Avatar Vertical Compu Br...	6514	2929.	97	30.20	
1995-07-24	X3L50	Coca-Cola	000009	2002-08-02	<a href="#">Download</a>
1996-09-01	D10T9	Mirrorsoft	000001	1997-09-16	<a href="#">Download</a>

To close the in-line panel, click the button indicated by the red arrow:

## Viewing Side Panel Search Results

Depending how the search developer configured a particular set of search results, further details may be contained in a side panel of the search's results. The side panel information is indicated by the red circle in the following screen shot:

Customer ID	First name	Last name	Call start	Call end	Call from	Call to	Sent to
000467	Lucas	White	2001-11-02T2...	2001-11-02T2...	684478207	351244391	2001-12-01
000345	Tristan	Clark	2001-11-14T0...	2001-11-14T0...	1283285404	1531838353	2001-12-01
000549	Alexa	Cooper	2001-11-26T1...	2001-11-26T1...	802423364	1730802154	2001-12-01
000068	Mia	Turner	2002-11-18T1...	2002-11-18T1...	865703616	1008136841	2002-12-01
000388	Jack	Smith	2002-11-16T1...	2002-11-16T1...	892778754	33303518	2002-12-01
000479	Christian	Singh	2002-11-30T1...	2002-11-30T1...	1320476452	1453609275	2002-12-01
000435	Alexander	Patel	2003-11-19T0...	2003-11-19T0...	1930331924	1255846933	2003-12-01
000022	Allison	Green	2003-11-16T2...	2003-11-16T2...	1780922848	1637939699	2003-12-01
000564	Henry	Foster	2003-11-15T2...	2003-11-15T2...	2138815642	1255846933	2003-12-01
000528	Nathan	Robinson	2004-11-18T1...	2004-11-18T1...	1620243383	755504356	2004-12-01

Note how the side panel contains two tabs, which contain different fields about the search results. Click the tab to view its contents.

## Downloading Search Results

Depending how the search developer configured a particular set of search results, further information can be downloaded from a set of search results.

During search composition, the Developer configures whether the search user is able to download information in the following formats:

- .pdf
- .tiff
- .png
- .jpeg
- .gif

To download information from a set of search results, click the **Download** button, as seen in the CID column in the following screenshot:

Record Search    Retention Sets    Hold Sets    Purge Lists    Application Info

Invoice Date > Results  
Invoice Date: 1992-01-01 - 2017-03-05

Select all 20

Displaying 1 - 10 of 20    Page 1 of 2

	Invoice Date ▲	Customer Number ...	Customer Name ▲	Invoice Number ▲	Ship Date ▲	CID
<input type="checkbox"/>	1995-04-24	LKP49	Ahold	000020	2000-04-02	<a href="#">Download</a>
<input type="checkbox"/>	1995-07-24	X3L50	Coca-Cola	000009	2002-08-02	<a href="#">Download</a>
<input type="checkbox"/>	1996-09-01	D10T9	Mirrorsoft	000001	1997-09-16	<a href="#">Download</a>

## Previewing Search Results

Depending how the search developer configured a particular set of search results, further information can be accessed by clicking a **View** link.

You may also be able to view and download files in a set of search results. During search composition, the search developer configures whether you are able to view and download information in the following formats:

- .pdf
- .tiff
- .png
- .jpeg
- .gif

The following screen shot illustrates a set of search results in the Invoice sample application. Note the two columns that are circled:

A > Invoices      Record Search      Application Info

Invoice Number > Results

Select all 20

Displaying 1 - 10 of 20      Page 1 of 2

	Customer ...	Customer ...	Customer ...	Invoice Date	Invoice Nu...	Ship Date	CID	Browser V...
▶	D10T9	774ZH69	Mirrorsoft	1996-09-01	000001	1997-09-16		<a href="#">View</a>
▶	NQ5HM	V9LL27Y	Aegis Develop...	1997-09-21	000002	2001-06-25		<a href="#">View</a>
▶	ITPH2	3FZM125	Sorcim	1997-07-19	000003	1998-01-17		<a href="#">View</a>
▶	QW3WU	MEFLXK	SCORAL	1997-09-07	000004	1998-07-09		<a href="#">View</a>

Click the **View** link to display a preview of the information (in this case, a .pdf file). To see an enlarged version of the item, click the arrows highlighted by the red arrow in the following screenshot:

**Eureka! Promotions**

**INVOICE**

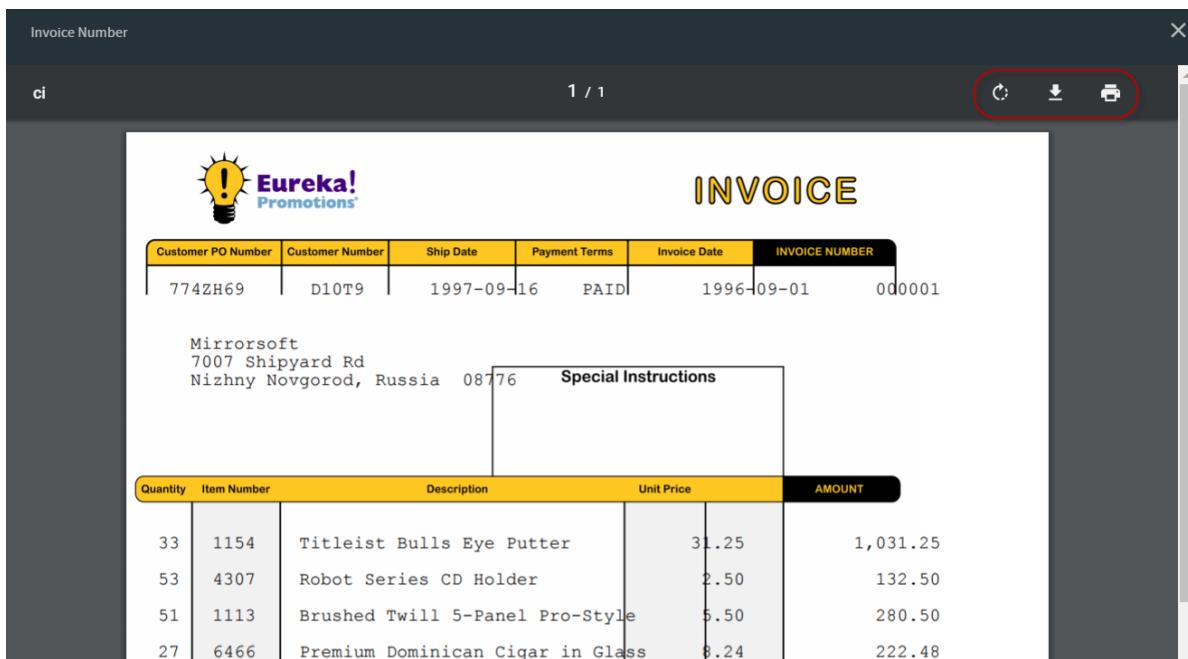
Customer PO Number	Customer Number	Ship Date	Payment Terms	Invoice Date	INVOICE NUMBER
7742H69	D10T9	1997-09-16	PAID	1996-09-01	000001

Mirrorsoft  
7007 Shipyard Rd  
Nizhny Novgorod, Russia 08776 Special Instructions

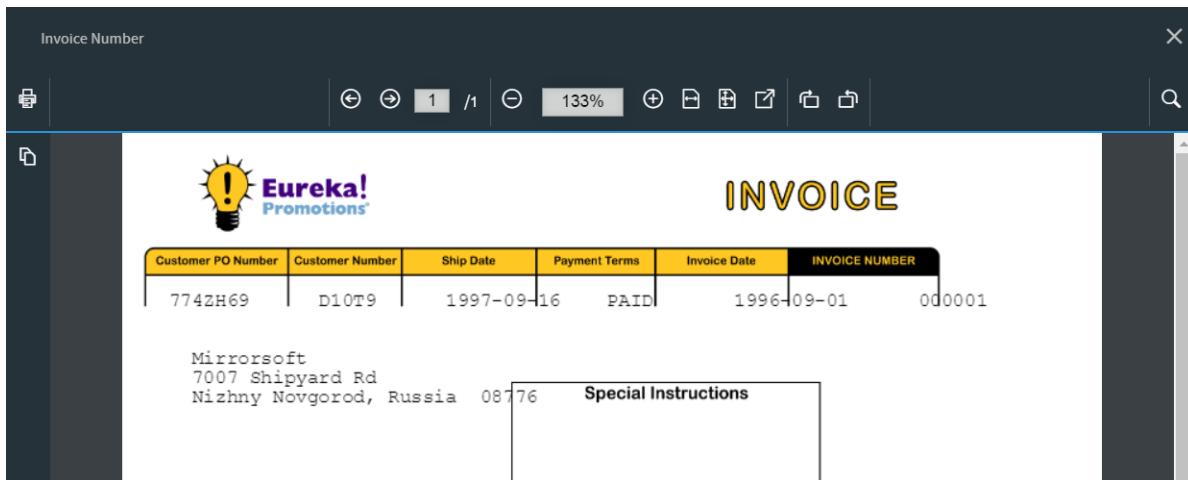
Quantity	Item Number	Description	Unit Price	AMOUNT
33	1154	Titleist Bulls Eye Putter	31.25	1,031.25
53	4307	Robot Series CD Holder	2.50	132.50
51	1113	Brushed Twill 5-Panel Pro-Style	5.50	280.50
27	6466	Premium Dominican Cigar in Glass	8.24	222.48
42	4335	Process Foam Wrist Rest	1.49	62.58
87	4320	Combo Wrist Rest/Fabric Mouse Pa	2.50	217.50
99	5423	Translucent Mouse Pad	12.25	1,212.75
40	6498	Safety Ridge Black Ash Tray	3.45	138.00
81	1150	Denim Cap with Micro-Suede Visor	1.40	112.00
60	3268	Pop-Up Note Dispenser	1.50	90.00

**Payment Terms PAID**

When enlarged, you can scroll through the information and rotate the item. Other actions may be permitted, such as printing or downloading the item. Again, the actions available depend on how the search developer configured the search results. The available actions in the following item are contained in the red circle, and include the ability to download or print the item:



The search developer may have configured the search results to allow even further options, such as the ability to search the item, jump to a specific page in the item, rotate and change the viewing settings for the item:



Click **X** or the **Esc** button on your keyboard to close the enlarged item.

## Exporting Search Results

The ability to export search results is only available if the search developer added the export option when the search was created.

It is possible to select all or only some of the search results to export. The number of items returned during the search is displayed in the **Select All** button. Once one or all of the search result items are selected, the **Export** button is displayed.

To select search result items for export:

1. Select one or multiple items. To select all of the search results, click **Select All #**.
2. Click **EXPORT** and select the export option. The options available depend on what the search developer opted to add when the search was created.
3. Enter a **Name** for the exported file or use the default name provided and click **OK**.

No matter how many search result items have been selected, access the exported files in the **Background Requests** tab. Refer to [Background Requests Tab](#) for further information.

There is a known issue in which search results that contain content (for example, .mp3 files) are exported, only the metadata is exported for the PDF, not the content. This issue occurs because, unless you export the content as part of the query, only the metadata will be exported. The issue will even occur if you select the PDF or the format with metadata and content.

## Background Requests Tab

When certain tasks are initiated, they are processed asynchronously, and the results are displayed on the **Background Requests**. These tasks include:

- Searches run in the background.
- When search results are exported into a CSV file.

Each request is displayed in a table that contains the following information:

Column	Description
<b>Name</b>	Indicates the name of the background request. Unless you changed the name of the request, the default name for the request includes the date and time the request was initiated.
<b>Type</b>	Possible values include: <ul style="list-style-type: none"><li>• <b>Search:</b> Displayed when the results are from a background search</li><li>• <b>Export:</b> Displayed when a user exported search results into a CSV file</li></ul>
<b>Application</b>	Indicates the application the request was initiated from.
<b>Submission Date</b>	Indicates the date and time the request was initiated.

Column	Description
<b>Duration</b>	Indicates the time the operation took to finish. Is not updated until the status is complete.
<b>Status</b>	<p>Status has one of the following values:</p> <ul style="list-style-type: none"> <li>• <b>Submitted:</b> The request has been submitted. As the framework picks up tasks every 10 seconds, this status may be rare.</li> <li>• <b>In Progress:</b> The request is still running.</li> <li>• <b>Completed:</b> The task completed with no errors</li> <li>• <b>Error:</b> The task could not be completed.</li> </ul>

Click the link in the **Status** column to review batch information and log information.

The **Logs** tab includes information related to the selected request.

Access the **Find a Log Item** field to locate specific log information.

The following check boxes allow you turn on/off the following types of information:

- **Info:** Includes general information about the request.
- **Warn:** Includes any warnings about the request.
- **Error:** Includes any errors that were encountered during the request.
- **Debug:** Includes any debugging information about the request.

To download log information:

- For order item logs, click **DOWNLOAD DIAGNOSTIC LOGS**.

## Viewing Search Results from a Background Search

You can view the results of a background search once the Status is **Completed**. Click **VIEW** to review the search results for a particular order item.

## Deleting a Background Task

1. For the background request item being deleted, click .
2. When prompted to verify that you want to delete the background request item, click **DELETE**.

# Chapter 9

---

## Appendix A – XQuery Best Practices

InfoArchive stores all archived metadata in XML format in xDB. As a result, searches have to be executed by using XQuery. In most cases, this is rather straightforward and there is no need to have a deep understanding of XQuery or xDB. However, if the performance of an XQuery does not meet the requirements, it may be necessary to investigate the cause of this bad performance. This section aims to provide some knowledge about XQuery optimization and to provide guidance in the best way to debug an XQuery.

InfoArchive supports two types of archiving: table and SIP archiving. Both archiving types use XQueries and indexes in different ways. The table below illustrates the main differences:

	Table-Based	SIP-Based
Search XQueries have to be defined manually	Yes	No
Support for multipath indexes	Yes	No
Support for path value indexes	Only in exceptional cases, definitions have to be defined manually	Yes

Because of these differences, some parts of this section especially apply to table archiving while other parts apply to SIP archiving.

Readers of this section are expected to have basic knowledge of the XQuery language.

## XQuery Structure

This section provides a global overview of the most important parts of the XQuery and how all of the combined parts support a search.

For SIP-based archiving, search XQueries are generated. Therefore, this section will only focus on table XQueries.

An XQuery consists of the following main XQuery parts:

- [Prolog](#)
- [FLWOR expression](#)

## Prolog

The prolog of an XQuery contains options and declarations that apply to the complete XQuery.

Examples of declarations are:

- namespace declarations,
- function declarations,
- module declarations, and
- variable declarations.

Examples of options are: xhive debug options.

For a table search XQuery, the form input is passed to the XQuery by using external variables. Before a search XQuery is executed, InfoArchive sets these variables. For this reason, InfoArchive must know which external variable corresponds to a specific form field. For this purpose, the name of the external variable must match the Form Field Data Binding value.

In the screen shot below, you can see in the Form Field dialog, a field with label Assignment Record Id is created. The Data Binding of this form field has name the assignmentRecordId, so there must be a matching external variable with the name \$assignmentRecordId in the XQuery.

**Note:** In this XQuery, the external variable has a default value. This value will be overwritten when the variable value is set by the application.

SEARCH FORM   RESULT LIST   RESULT DETAIL   **QUERY EDITOR**   PERMISSIONS

Enter Xquery below.

```

1 declare namespace ia = "urn:x-emc:ia:schema:fn";
2 declare namespace table = "urn:x-emc:ia:schema:table";
3 declare variable $assignmentRecordId external
4 := <assignmentRecordId>1</assignmentRecordId>;
5 let $rows :=
6 for $elem in /PATENT/ASSIGNMENT_RECORDS/ROW
7 where $elem/ASSIGNMENT_RECORD_ID = xs:integer($assignmentRecordId)
8 return $elem
9 return <results total="{ count($rows) }">
10 {
11 for $elem in $rows
12 return
13 <row id="{ string($elem/@table:id) }">
14 <column name='assignmentRecordId'>
15 { $elem/ASSIGNMENT_RECORD_ID/text()
16 }
17 <column name='correspondentName'>
18 { $elem/CORRESPONDENT_NAME/text()
19 }
20 <column name='correspondentName_addr'>
21 { $elem/CORRESPONDENT_ADDRESS_1/text()
22 }
23 <column name='correspondentName_addr'>
24 { $elem/CORRESPONDENT_ADDRESS_2/text()
25 }
26 <column name='correspondentName_addr'>

```

Edit Field

UI Control: <input type="text"/> Data Binding*: <input style="border: 2px solid #000; border-radius: 15px; width: 150px; height: 20px;" type="text" value="assignmentRecordId"/> Field Label: Assignment Record Id	Required: <input type="checkbox"/> Hidden: <input type="checkbox"/>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------

## FLWOR Expression

A FLWOR expression consists of multiple clauses, the main clauses are:

- for,
- let,
- where,
- order by, and
- return.

By using these clauses, a very simple table search XQuery can be constructed.

The for clause is used for iteration of the XQuery context nodes. In a table XQuery, the for clause usually iterates all ROW elements of a specific table.

The where clause is used to filter the nodes found by matching the values inserted in the form with the corresponding field elements values.

The return clause returns constructed elements where the root element has name <row> containing a child element with name <column> for each Column in the RESULT LIST. Each <column> element has a name attribute with name which corresponds to the Column Name.

In the screen shot below you can see a Column dialog of the **Result List** tab. The name of the column is Correspondant Name and the name of the Column Name is correspondantName. This correspondantName must be used as attribute name in the column element which holds the value of element CORRESPONDENT\_NAME in the data.

SEARCH FORM    RESULT LIST    RESULT DETAIL    QUERY EDITOR    PERMISSIONS

Enter Xquery below.

```

1 it query with specific assignment record id (int)
2
3 namespace ia = "urn:x-emc:ia:schema:fn";
4 namespace table="urn:x-emc:ia:schema:table";
5
6 variable $assignmentRecordId external;
7
8 query-results :=
9 $elem in /PATENT/ASSIGNMENT_RECORDS/ROW
10 st to integer is needed for index usage :)
11 $elem/ASSIGNMENT_RECORD_ID = xs:integer($assignmentRecordId)
12 $elem
13
14
15 $total := count($query-results)
16
17 for $query-result in $query-results
18 return
19 <row id="string($query-result/@table:id)">
20 <column name='assignmentRecordId'>{ $query-result/ASSIGNMENT_RECORD_ID/text() }</column>
21 <column name='correspondentName'>{ $query-result/CORRESPONDENT_NAME/text() }</column>
22 <column name='reelNo'>{ $query-result/REEL_NO/text() }</column>
23 <column name='frameNo'>{ $query-result/FRAME_NO/text() }</column>

```

**Edit Column**

Properties	Linked Column
Column Label	Correspondent Name
Column Type	XQuery Reference
Column Name	correspondentName

XDB Schema    PATENT

FLWOR expressions can be nested.

The order by clause can be used to order returned XQuery results. However, there is no need to order the result of a search XQuery. The reason is that search results are cached-in InfoArchive.

Whenever a user would like to order the result by a specific field (for example, Correspondent Name), the cached result is displayed in this order. Therefore, there is no need to use an order by clause in the search XQuery.

## XQuery Modules

If a specific function declaration is used in multiple search XQueries, it is best to store the function declaration in an xquery-module. The xquery-module is stored in a separate location. Function or variable declarations stored in the XQuery module can be referenced and used from within an XQuery.

XQuery modules can be scoped on both the application and tenant levels. The Tickets sample application illustrates how to use the XQuery module at tenant level.

To change its scope to be on the application level:

- For the ANT version of the sample applications, change the `Tickets/xquery-modules/iau-utility-tenant.xml` file to have the context:  
`<context>application</context>`
- For the declarative configuration version of the sample applications, change the scope and set the application (now set to null) in `<INFOARCHIVE_ROOT>\examples\applications\Tickets\config\xquery-modules\configuration.yml`:  
`xqueryModule:  
 name: iau-utility-tenant  
 application: Tickets  
 context: application`

## XQuery Performance Considerations

This section gives a high-level overview of aspects that affect XQuery performance.

- Reduce Page Access: XQueries run on data stored in the xDB database. XQuery performance is best when page access can be reduced as much as possible. For this purpose, indexes are used and can traverse filtered data in the most efficient way.
- Reduce the Size of the (Intermediate) XQuery Result: The bigger the size of the XQuery result, the slower the XQuery. An unselective XQuery may potentially return many XQuery results. In the worst case scenario, all data is returned. Especially when using large data sets, it is important to try to minimize the size of the result. A way to reduce the chance of bad performance caused by a large result is to make the XQuery more restrictive. A way to achieve this is by making form fields required.

It is also recommended to keep the returned constructed elements as small as possible by only returning field values that are really required.

An intermediate XQuery result is the result of a xquery-part that is used as input to another xquery-part. The smaller the intermediate results, the less page access. The [Table Joins](#) section provides an example of the effect of the size of an intermediate result.

- Memory Usage: xDB stores XML natively. This allows xDB to execute XQueries on very large XML documents without running out of memory. There are, however, some cases where it is required to execute XQuery parts completely in memory. For example, the group by clause always groups nodes in memory. For this reason, it is also important to always ensure that the size of (intermediate) results is limited.
- Usage of an Unordered Declaration: XQuery has a setting to specify if expressions must return nodes in document-order.

For historical reasons, the default ordering of the xDB XQuery implementation is ordered. This setting can cause unexpected poor XQuery performance, for instance, when ordering is applied to an (intermediate) result retrieved from an index. As results retrieved from an index are not in document-order, there is a potential risk of unnecessary ordering.

To prevent this ordering, use the following declaration in the prolog of the xquery:

```
declare ordering unordered;
```

Starting from xDB 11.2, the default ordering is unordered.

- Limit Size of XQuery Context: The xDB library structure used for SIP archiving is very different from the library structure used for table archiving. To understand more about the effect of this library structure on XQuery performance, it is necessary to understand more about the selection of the XQuery context. Refer to [XQuery Context and Index Location](#) for more information.

From this section it can be concluded that the number of AIPs used as XQuery context of an XQuery must be limited to reduce the impact of index merges on XQuery performance.

Refer to the documentation of SIP archiving to use search criteria to limit the number of AIPs in an XQuery.

## XQuery Optimization

The main optimization of XQueries is achieved by the use of indexes. For this purpose, xDB has multiple types of indexes. This section discusses the different types of indexes and their uses in table and SIP archiving.

Refer to the *xDB Admin Client Guide* for more specific information on indexes.

## Index Types

### Multipath index

The multipath index is based on the Lucene full-text index. It has one single main-path and a set of sub-paths.

The main-path is the path to the elements to be stored in the index. For instance, for table archiving, the main-path is usually the path to the ROW element (for example, /BASEBALL/MASTER/ROW). The sub-paths are paths with the main-path as context. Examples include 'DEBUT', 'NAMELAST',

etc. These elements contain the actual values. You can add as many sub-paths as required. Each sub-path has options and a type. The default type is String. There are multiple sub-path options, such as full-text searching and value comparison. One single multipath index can be used for optimization of all possible table XQueries.

## Path Value Index

The path value index is a B-Tree index. xDB supports more B-Tree index types, such as the value index and full-text index, but these index types are a subset of the path value index.

In the rest of this section, we will mainly refer to path value indexes. A path value index is based on a path definition. An example of a Path value index definition is /BASEBALL/MASTER/ROW[NAMELAST]. The index stores ROW elements by key which, in this case, is the value of element NAMELAST. Index keys can have different data types and are always stored ordered. This way, the index lookup for node(s) by key is very efficient.

The disadvantage of a path value index is that it is not possible to let one single index cover a complete schema. You may require multiple path value indexes for the optimization of all XQueries for one particular schema. This makes the process of optimization of an XQuery more complex. As for table archiving, the multipath index is already used, path value indexes are only of use in exceptional cases. For SIP archiving, this is the only index type supported.

## Composite key index

Path value indexes support multiple index keys. A path value index with multiple index keys is called a composite key index. For instance, consider the following XQuery:

```
for $r in /BASEBALL/MASTER/ROW
 where $r/BIRTHSTATE = 'CA' and $r/DEBUT < xs:dateTime('1990-01-01T00:00:00')
 return $r
```

A composite key index to find the nodes by these values with one single index lookup is /BASEBALL/MASTER/ROW[BIRTHSTATE<STRING> + DEBUT<DATE\_TIME>]

An index lookup of a composite key index is very efficient because the keys can be more selective and it can cover a larger part of the path expression. It is important, however, to use this index in the proper way. If the first part of the key is very unselective and the second part is selective, then, in some cases, an XQuery using this composite key index may be much slower than an XQuery using a Path value index with only the selective part as key.

The disadvantage of a composite key index is that it can only be used when the XQuery has a condition for every key. The reason is that if the element of one of the keys does not exist, the element is not stored in the index. So, if you know the element of a key does exist, but you are not interested in the value, use an 'exists' condition for the element. For example, if only the BIRTHSTATE value is set and you know the DEBUT element exists, the following XQuery will use the index:

```
for $r in /BASEBALL/MASTER/ROW
 where $r/BIRTHSTATE = 'CA' and $r/DEBUT < xs:dateTime('1990-01-01T00:00:00')
 return $r
```

Composite key indexes also support full-text search, but only one single full-text index key is allowed.

## Full-Text Search Versus Value Comparison

This section is especially of use for table archiving as for SIP archiving.

Both multipath and path value index support full-text searches. In both XQuery and the index definition, the search type must match, otherwise the index will not be used.

For instance, the patent ASSIGNMENT\_RECORDS table has the following field element <CORRESPONDENT\_NAME>THOMAS J. ENGELLENNER</CORRESPONDENT\_NAME>.

To search for an exact match, do a value comparison by using condition \$row/CORRESPONDENT\_NAME = 'THOMAS J. ENGELLENNER'.

**Note:** A value comparison is always case-sensitive.

If, however, you allow full-text search for a part of the name, for instance 'thomas', then the condition must be \$row/CORRESPONDENT\_NAME contains text 'thomas'. To allow wildcards, the condition must be \$row/CORRESPONDENT\_NAME contains text 'thom.\*' using wildcards.

Value search is very useful to find records by matching field values like dates, IDs and numbers. When a record must be found by matching search terms in a field value, full-text search must be used.

Full-text search uses an analyzer to define the search terms. The analyzer is used to:

- Tokenize the search terms used in a full-text XQuery.
- Tokenize the field value before it is stored in the full-text index.

For search and index, exactly the same analyzer must be used.

## Full-Text Analyzer

At this moment, the default xDB analyzer is used for all full-text searches and indexes. This analyzer only tokenizes letters and digits. As a result, it is not possible to search full-text to find an e-mail address like 'john.doe1@opentext.com'. This is because this string would be tokenized as 'john', 'doe1', 'opentext', 'com', as the other characters are identified as separators.

The analyzer is partly configurable. For instance, it has an option named FTI\_SA\_ADJUST\_TO\_LOWER CASE. When this option is set, all terms are stored lowercase in the index. Search terms in the XQuery will be first made lowercase before they are matched against the index terms.

Another option is FTI\_SA\_FILTER\_ENGLISH\_STOP\_WORDS. If this option is set, all terms matching English stopwords like "a", "and", "are", "as" and "at" are filtered from the indexed search terms.

Refer to the xDB manual to find the complete set of full-text index options.

To ensure that both index and search use the same analyzer, the analyzer options of the default xDB analyzer are stored in the full-text index. When an XQuery uses a full-text condition, the XQuery optimizer will find out what full-text index is used for the condition. The XQuery will then use the analyzer and options stored on the index to tokenize the terms of the condition.

In some use cases, the default xDB full-text analyzer can return unexpected results due to the tokenization. For instance, if the value is an email address, john.smith@opentext.com, then the default xDB full-text analyzer will split the text in john, schmidt, opentext and com. If a search user uses the operator Contains with the value john.smith, the search will return no records.

To change this behavior, InfoArchive provides InfoArchiveAnalyzer, a new out-of-the-box full-text analyzer to remove the tokenization. This analyser is not enabled by default and must be configured manually.

To enable this analyzer, the Administrator must edit the PDI configuration file to have exported the holding configuration. All path value indexes with the full-text option need to be updated, as illustrated below. After the change, the new configuration must be imported again. If some AIPs have been already injected, the Post Ingest Processing job must be executed to refresh the indexes with the new settings.

### Before

```
<datas>
 ...
 <data>
 <id>pdi.index.creator</id>
 <key.document.name>xdb.pdi.name</key.document.name>
 <indexes>
 <path.value.index>
 <name>Email</name>
 <path>/ {urn: eas-samples:en:xsd:phonecalls.1.0}Calls/{urn: eas-samples:
 en:xsd:phonecalls.1.0}Call[{urn: eas-samples:en:xsd:phonecalls.
 1.0}Email<FULL_TEXT::SA_ADJUST_TO_LOWERCASE,SUPPORT_PHRASES>]
 </path>
 </indexes>
 </data>
 ...
<datas>
```

### After

```
<datas>
 ...
 <data>
 <id>pdi.index.creator</id>
 <key.document.name>xdb.pdi.name</key.document.name>
 <indexes>
 <path.value.index>
 <name>Email</name>
 <path>/ {urn: eas-samples:en:xsd:phonecalls.1.0}Calls/{urn: eas-samples:
 en:xsd:phonecalls.1.0}Call[{urn: eas-samples:en:xsd:phonecalls.
 1.0}Email<FULL_TEXT:com.emc.ia.common.xdb.analyze.
 InfoArchiveAnalyzer:SA_ADJUST_TO_LOWERCASE,SUPPORT_PHRASES>]
 </path>
 </indexes>
 </data>
 ...
<datas>
```

## Search for Typed Data

This section is especially of use for table archiving as for SIP archiving.

For all index types, it is important to use the proper data types. By default, this is String. If you want to search for xs:dateTime values, both index sub-path type and the XQuery condition must match.

For instance, if you want to search for String '1959', then your condition must be:

```
$row/BIRTHYEAR = '1959'
```

If you want to search for 1959 as number, then your condition must be:

```
$row/BIRTHYEAR = 1959
```

To search for a value of type xs:dateTime, you must cast the found value:

```
$row/DEBUT > xs:dateTime('1820-01-01T00:00:00')
```

However, as InfoArchive sets the external variables by using the complete form element, additional casting may be required for numbers.

For example, if the value 1 is used as input for the Record Id Search of the Patent application, and the Data Binding of the Form Field is assignmentRecordId, then the value of the corresponding external variable: \$assignmentRecordId is <assignmentRecordId>1</assignmentRecordId>

When using this form element in the value comparison in the following manner:

```
declare variable $assignmentRecordId external := <assignmentRecordId>1</assignmentRecordId>;
let $rows :=
 for $elem in /PATENT/ASSIGNMENT_RECORDS/ROW
 where $elem/ASSIGNMENT_RECORD_ID = $assignmentRecordId
 return $elem
```

Then the value \$assignmentRecordId, according to the XQuery spec, is cast to a String but, if ASSIGNMENT\_RECORD\_ID is indexed as an integer type field, then the index is not used for this condition. So, in this case, \$assignmentRecordId must be explicitly cast to an integer:

```
where $elem/ASSIGNMENT_RECORD_ID = xs:integer($assignmentRecordId)
```

## Table Joins

In table archiving, you may want to create an XQuery where two or more tables are joined. Unfortunately, xDB does not support optimization of table joins. As a result, these joins should be avoided if possible for large tables.

If table joins must be used, then it should take into account that the performance of the XQuery highly depends on the way the tables are joined. The XQuery should be constructed in a way that the number of intermediate results is reduced as soon as possible. If two tables are joined, and both tables have their own index, the outer loop should be the part with the most selective conditions.

To illustrate the impact of the order of for loops, see the following simple XQuery with table join executed on the BASEBALL dataset: get the last name and birth date of all players that have won the 'Comeback Player of the Year' award:

```
let $award := 'Comeback Player of the Year'
for $m in /BASEBALL/MASTER/ROW
for $a in /BASEBALL/AWARDSPLAYERS/ROW
where $a/PLAYERID = $m/PLAYERID and $a/AWARDID = $award
return $m
```

This XQuery will traverse all MASTER rows. Then, for every row, the corresponding AWARDSPLAYERS record with given PLAYERID is checked for the award. As there are 18589 rows in the MASTER table, there are 18589 index lookups to find the corresponding AWARDSPLAYERS rows to check if the PLAYERID and the award matches.

This XQuery takes 4450 milliseconds.

By switching the loops, the XQuery takes only 50 milliseconds. This XQuery will first find the PLAYERID with the award in the AWARDSPLAYERS table, then retrieves the corresponding record in the MASTER table to get the NAMELAST and BIRTHYEAR of the player.

As there are 21 different players with the award, only 21 index lookups are required to identify the MASTER records.

## Range Queries

When using the multipath index, ranges are optimized. When using path value indexes, the XQuery must be written in a certain format.

For SIP searches, the XQueries are already generated in the proper way.

Assume we have the following path value index: /BASEBALL/MASTER/ROW[DEBUT<DATE \_TIME>], and we use the following XQuery:

```
for $r in /BASEBALL/MASTER/ROW
where $r/DEBUT > xs:dateTime('1950-01-01T00:00:00') and $r/DEBUT
< xs:dateTime('1990-01-01T00:00:00')
return $r
```

In this case, only one condition is optimized by the index:

```
query:7:11:Using query plan:
query:7:11:index(DEBUT) [child::DEBUT[. < ...]]
```

In order to make the index optimize both conditions, the where clause has to be rewritten in the following format:

```
for $r in /BASEBALL/MASTER/ROW
where $r/DEBUT[. > xs:dateTime('1950-01-01T00:00:00') and . < xs:dateTime
('1990-01-01T00:00:00')]
return $r
```

The reason is that only this way the optimizer knows both conditions apply to the same DEBUT element.

The debug output shows that the index is used to optimizes both conditions:

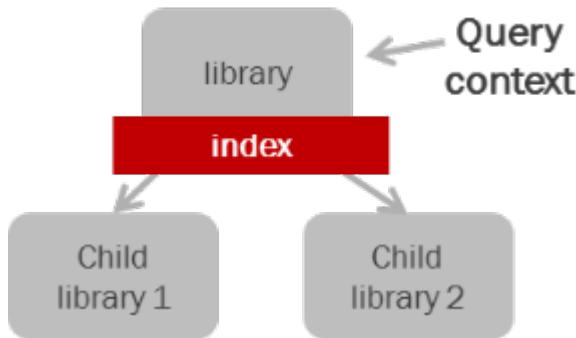
```
query:7:11:Using query plan:
query:7:11:index(DEBUT)
query:7:11:Looking up "(1950-01-01T00:00:00,1990-01-01T00:00:00)" in index "DEBUT"
```

## XQuery Context and Index Location

Whether or not an index is used for an XQuery, highly depends on where the index is located in relation to the XQuery context. The XQuery context is a library or document used as context for XQuery execution. The following use-cases can be identified:

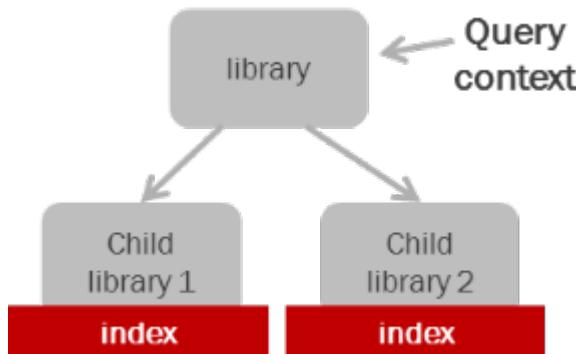
### The XQuery Context Contains the Index

If the XQuery context is one single library which also contains the index, the index is used. This is the most optimal usage of indexes, as only one single index has to be used. Table archiving uses this type of XQuery context.



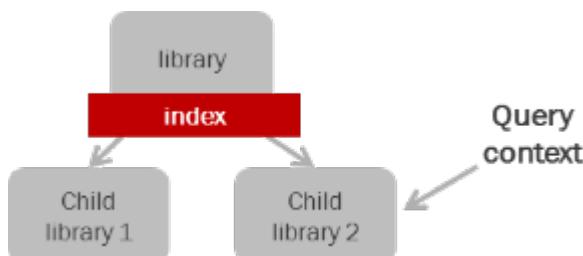
## The Children of the XQuery Context Contain the Indexes

This is equivalent to the situation where the XQuery context is a collection of libraries or documents that contain the indexes. The indexes are used for optimization, but the XQuery results must be retrieved from different indexes and must be merged. The higher the number of children, the more results have to be merged. This has a negative impact on XQuery performance.



## The Parent of the XQuery Context Contains the Indexes

If the parent of the XQuery context contains the index, the index is not used.



## XQuery Optimizer

The selection of the best possible indexes for an XQuery is done by the XQuery optimizer during XQuery execution. Based on the selected indexes an index plan is created.

This section aims to give a better understanding of the strategy used by the optimizer to create this index plan.

**Note:** This section is of importance to SIP archiving because multiple path value indexes can be used for a single search.

For table archiving, index selection is usually trivial because, in most cases, one single multipath index will cover the complete path expression. However, when a schema has hundreds of tables, all with a multipath index, a large amount of indexes is located on the same library. If no index selection options are set, the optimizer will have to inspect each of these indexes for every single Search. By setting Index Selection Options, the optimizer can skip this time-consuming index selection process.

[Debugging an XQuery](#) explains how to check which indexes are selected by the optimizer.

## Optimizer Strategy

While analyzing the indexes found in the XQuery context, the optimizer gives scores to the indexes. The index(es) with the highest score is (are) selected.

To select the best indexes, the XQuery optimizer executes the following steps:

1. Collect all indexes found in the XQuery context and filter the list depending on the index options.
2. Give scores to each index based on the following criteria:
  - The better the path expression matches the index paths, the higher the score.
  - The higher the number of conditions match the index keys, the higher the score.
  - The higher the number of ordering conditions, the higher the score.
3. If multiple indexes have the highest score, index [statistics](#) is used to select the best index.
4. If one single index can cover the complete path expression, this is considered the best possible index.  
If an index is found that does not cover the complete path expression, the analyzer will iterate over steps 2, 3 and 4 to find additional indexes for the rest of the expression.
5. When the optimizer is done, it will return an index plan if any suitable indexes are found.

### The Selection Done by the Optimizer is not Always the Best

XQuery optimization is a complex task. Unfortunately, the optimizer does not always select the best index plan. For this reason, it is important to test the performance of XQueries before going into production.

Examples of bad optimization selections:

1. The optimizer has selected an index plan with an intersection of indexes. In many cases, the XQuery is faster when only using one single index.
2. The optimizer found multiple indexes with the same score, but selected the least selective index. This may occur when index analytics is not available or not up-to-date.
3. The optimizer has selected a composite index with a very non-selective first key which is used for range XQueries.
4. The optimizer selects a composite key index as index, while a single key index is more selective.

## Influencing Index Selection

If it is detected that the XQuery optimizer selects the wrong index for an XQuery, the options `xhive:ignore-indexes` and `xhive:use-indexes` can be used to influence the index selection process. These options can also be used to improve the performance of the optimization process.

1. Ignore indexes:

Use the XQuery option `xhive:ignore-indexes` with a comma-separated list of index names to tell the optimizer to ignore the indexes. For example:

```
declare option xhive:ignore-indexes 'name1, name2';
```

2. Use indexes:

Use the XQuery option `xhive:use-indexes` with a comma-separated list of index names to tell the optimizer to only use these indexes. All other indexes will be ignored. If, for table archiving, a schema has hundreds of table definitions, a large number of indexes is located in the schema library (see Context and Index location). To prevent inspection of all these indexes by the optimizer, option `xhive:use-indexes` can be used to specify the indexes to be used by the XQuery.

The name of a multi-path index for a table always has the following pattern:

`SQL-INDEX/[schema]/[table]/ROW`. So, Table MASTER of Schema BASEBALL has a multi-path index with the name `SQL-INDEX/BASEBALL/MASTER/ROW`.

For example, if the search XQuery only uses BASEBALL Table MASTER, set the following option at the top of the XQuery:

```
declare option xhive:use-indexes 'SQL-INDEX/BASEBALL/MASTER/ROW';
```

If the search XQuery has a table join over BASEBALL Tables MASTER and AWARDSPLAYERS, use option:

```
declare option xhive:use-indexes 'SQL-INDEX/BASEBALL/MASTER/ROW,
'SQL-INDEX/BASEBALL/AWARDSPLAYERS/ROW';
```

**Note:** InfoArchive automatically sets the `xhive:use-indexes` option when a table is selected as the context of a search.

## Hidden Optimization for Table Archiving

InfoArchive automatically sets the `xhive:use-indexes` option when a table is selected as the Archival Collection of the search.

You will not see this option in your XQuery, as InfoArchive sets this option internally by using the API. A drawback of this optimization is that, if you select a table as the Archival Collection and, in the XQuery, join the table with another table, the join may not be optimized. Make sure to not set a table as the Archival Collection when you have to use a table join in your XQuery.

## Index Statistics

xDB has some index statistics that are used by the XQuery optimizer to select the most selective index when two indexes have the same score.

Index statistics is up-to-date for an index when it is created after the data is loaded. However, it is not updated automatically when data is loaded after index creation. As in InfoArchive, data is always loaded before index creation, and it is expected that index statistics is always up-to-date.

## The Optimizer Only Optimizes a Subset of XQuery Expressions

The extensive XQuery language supports a large variation of XQuery expressions. Unfortunately, the optimizer can only optimize a subset of these expressions. Some expressions cannot be supported by any index type, other expressions are only supported by one of the index types.

As both table and SIP archiving documents have a fixed or similar schema, the for clause part is usually not the reason why an XQuery cannot be optimized. One example is the lacking optimization support for [Table Joins](#). The part where the optimizer is most likely to fail unexpectedly is the where clause, which is where the query conditions are located.

Consider the following XQuery:

```
declare function local:getDate($year, $month, $day) {
 concat($year, $month, $day)
};
for $elem in /PATENT/ASSIGNMENT_RECORDS/ROW
where $elem/RECORDED_DATE = local:getDate("1979", "03", "05")
return $elem
```

As RECORDED\_DATE is indexed as String and the dates are compared as String, this condition is optimized.

However, the XQuery is not optimized by both index types when using the following negated condition:

```
for $elem in /PATENT/ASSIGNMENT_RECORDS/ROW
where not($elem/RECORDED_DATE = local:getDate("1979", "03", "05"))
return $elem
```

The following table provides some value comparisons and their index support:

Condition	Multipath Index	Path Value Index
not (\$elem/RECORDED_DATE = "19790305")	No	No
\$elem/RECORDED_DATE != "19790305"	Yes	No
\$elem/RECORDED_DATE ne "19790305"	Yes	No

Condition	Multipath Index	Path Value Index
<code>starts-with(\$elem/RECORDED_DATE, "1979")</code>	Yes	Yes
<code>contains(\$elem/RECORDED_DATE, "03")</code> <b>Note:</b> This is not the same as the full-text 'contains text'.	Yes	No
<code>ends-with(\$elem/RECORDED_DATE, "05")</code>	Yes	No

For a full-text search, the mult-ipath index also supports more comparison types than the path value index.

## Debugging an XQuery

For both SIP and table archiving, the XQuery execution may show bad performance, even if indexes are defined. In this case, it is necessary to investigate the optimization of the XQuery.

For table archiving, the customer has to create a search XQuery manually. This can be done by using the InfoArchive Query Editor. Typically, the best way to start a new XQuery is to copy an existing one and adapt it to the data, search form and result list of the search. If the XQuery does not perform, the user can decide to investigate the XQuery debug the output.

There are two ways to produce the debug output of an XQuery:

1. By setting the XQuery debug options on a search.
2. By running the XQuery in the xDB Admin tool.

Setting the debug options on a search may be the most convenient way when you want to get debug information fast.

Usage of the xDB Admin tool may be the most convenient way when your query does not compile, or when you want to modify and debug the XQuery at the same time. The problem is that the XQuery may have to be adjusted because it is running outside of InfoArchive.

## Making a Search Produce XQuery Debug Output

InfoArchive allows the user to set the following debug options on a search:

```
searchXqueryOptions :
 optimizerDebug : false
 indexDebug : true
 queryplanDebug : true
 pathexprDebug : true
sdxXqueryOptions :
 optimizerDebug : false
 indexDebug : false
 queryplanDebug : false
 pathexprDebug : false
```

The searchXQueryOptions are used for the actual Search, the sdxXqueryOptions are for the partition keys query. A partition key query is executed only for a SIP search.

The indexDebug, queryPlanDebug and pathExprDebug options produce useful debug output that provides a better understanding of how the XQuery is executed. The optimizerDebug should not be combined with these options. This option is best used separately when deeper knowledge of the internal decision process of index usage is required. For instance, to understand why another index than expected is used.

These debug options may produce so much output that it affects the performance of the actual search. For this reason, the debug options for a search must always be in-set or deleted when no longer required.

To set the debug options of a search, you must use IA Shell:

1. Connect and traverse to the search you want to debug.
2. Navigate to search-debug:

```
ia-shell>cd search-debug
```

3. Upload a configuration in a file and run the following command:

```
ia-shell>import --from C:/some/location/search-debug.yml
version: 1.0.0
tenant:
 name: INFOARCHIVE

application:
 name: Baseball
 configure: use existing
search:
 name: Player Search - Custom Search Form
 configure: use existing

searchDebug:
 search: Player Search - Custom Search Form
 searchXqueryOptions :
 optimizerDebug : false
 indexDebug : true
 queryplanDebug : true
 pathexprDebug : true
 sdxXqueryOptions :
 optimizerDebug : false
 indexDebug : false
 queryplanDebug : false
 pathexprDebug : false
```

After executing the search, the debug output appears in the logs/iaserver/ia.log file.

4. Ensure that the debug options are deleted when no longer required:

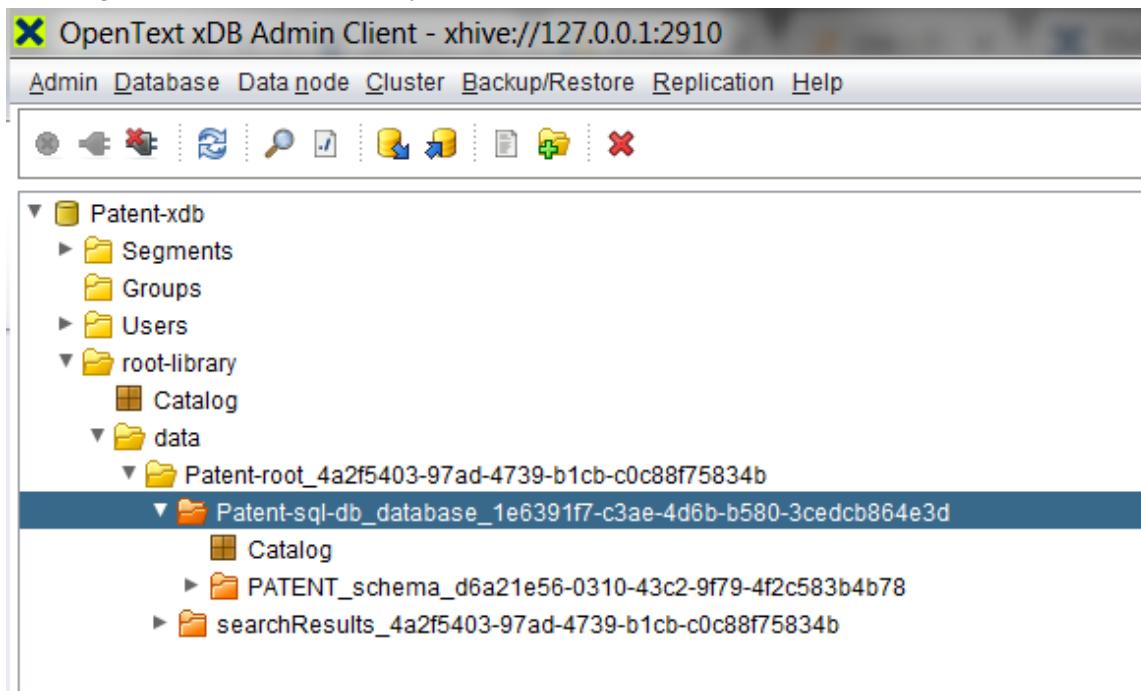
```
iashell> cat
79080b06-742-4630-8acd-606696adf1b
iashell> cd 79080b06-f742-4630-8acd-606696adf1b6
iashell> delete
```

## How to Use the xDB Admin Tool for Debugging

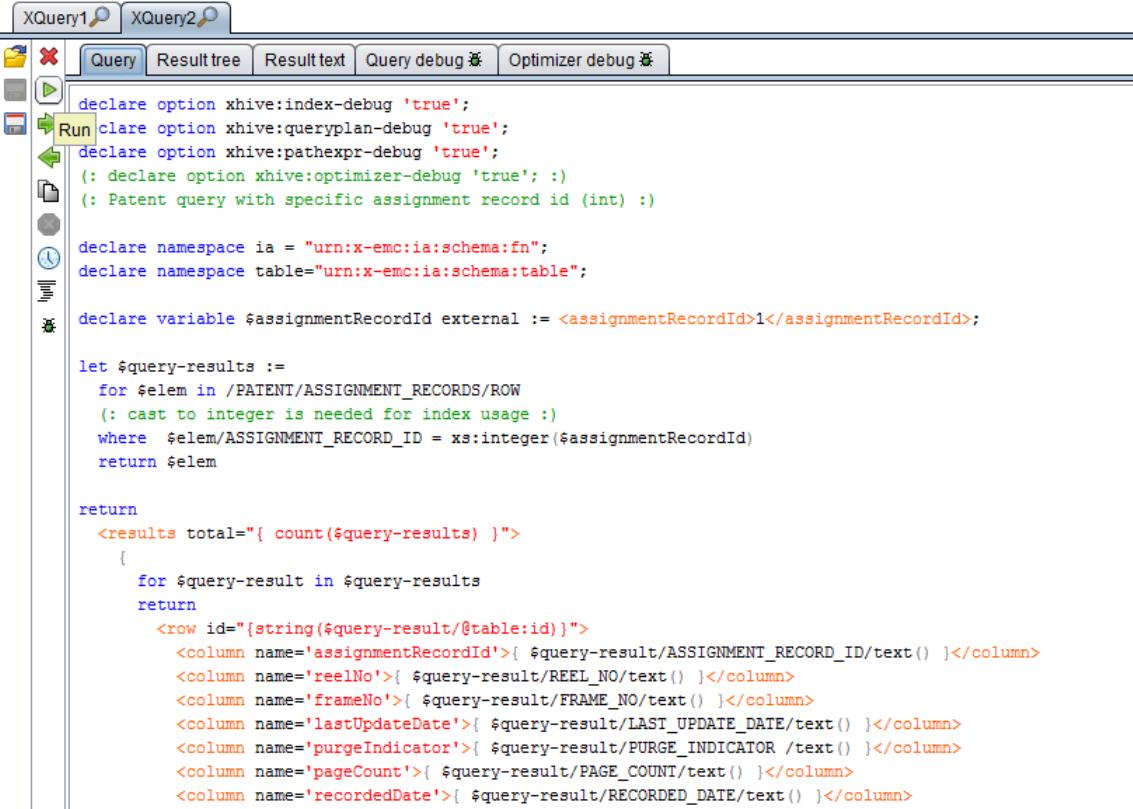
The xDB Admin tool is better suited for debugging purposes and, therefore, is the best tool to use. However, there are some drawbacks. For instance, InfoArchive-specific classes are not in the classpath of the xDB Admin tool. For this reason you cannot use the encryption functions from within the xDB Admin tool. Also, the InfoArchive URI resolver, which is internally called when SIP XQueries use the collection('my uri') function, cannot be supported. This section provides tips to work around usage of these classes.

To run an XQuery with the xDB Admin tool:

1. Start the xDB Admin tool.
2. Select the database that contains the data on which your XQuery will run.
3. Select the library or document to be used as context of the XQuery. Instructions can be found in [Selecting the Context of an XQuery](#).



4. Right-click on this library or document and select option 'Execute XQuery'. At the bottom there will be a tab with name 'Query'.
5. Enter your XQuery in the **Query** tab, as illustrated in the following screen shot:



```
declare option xhive:index-debug 'true';
declare option xhive:queryplan-debug 'true';
declare option xhive:pathexpr-debug 'true';
(: declare option xhive:optimizer-debug 'true'; :)
(: Patent query with specific assignment record id (int :))

declare namespace ia = "urn:x-emc:ia:schema:fn";
declare namespace table="urn:x-emc:ia:schema:table";

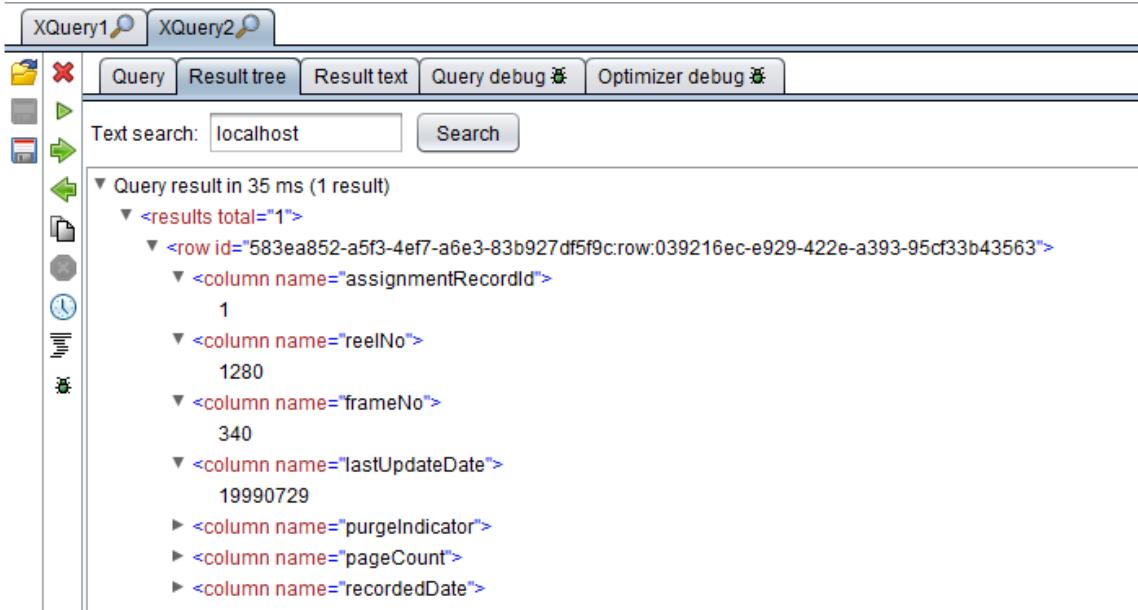
declare variable $assignmentRecordId external := <assignmentRecordId>1</assignmentRecordId>

let $query-results :=
 for $elem in /PATENT/ASSIGNMENT_RECORDS/ROW
 (: cast to integer is needed for index usage :)
 where $elem/ASSIGNMENT_RECORD_ID = xs:integer($assignmentRecordId)
 return $elem

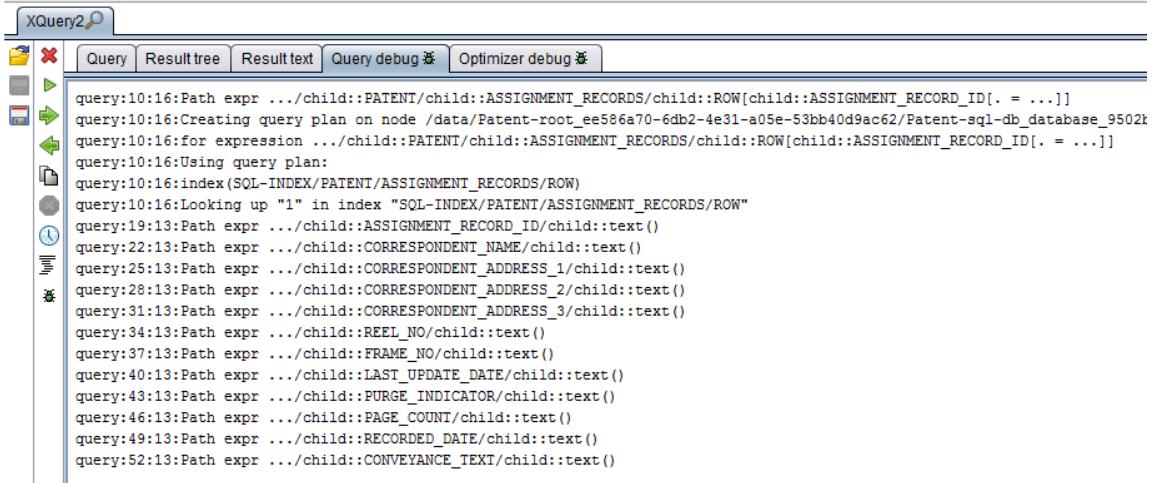
return
<results total="{ count($query-results) }">
{
 for $query-result in $query-results
 return
 <row id="{string($query-result/@table:id)}">
 <column name='assignmentRecordId'>{ $query-result/ASSIGNMENT_RECORD_ID/text() }</column>
 <column name='reelNo'>{ $query-result/REEL_NO/text() }</column>
 <column name='frameNo'>{ $query-result/FRAME_NO/text() }</column>
 <column name='lastUpdateDate'>{ $query-result/LAST_UPDATE_DATE/text() }</column>
 <column name='purgeIndicator'>{ $query-result/PURGE_INDICATOR /text() }</column>
 <column name='pageCount'>{ $query-result/PAGE_COUNT/text() }</column>
 <column name='recordedDate'>{ $query-result/RECORDED_DATE/text() }</column>
```

There are some icons on the left side that represent the following actions:

- To format your XQuery, press the 'Format the xquery' icon, the second item from the bottom.
  - To add debug options to the XQuery, press the 'Toggle debug output' icon, the item at the bottom. The following declarations are added to the top of the XQuery:
    - declare option xhive:index-debug 'true';
    - declare option xhive:queryplan-debug 'true';
    - declare option xhive:pathexpr-debug 'true';
  - To add debug options to the XQuery, press the 'Toggle debug output' icon, the item at the bottom.
  - To run the XQuery, press the 'Run' icon, the second icon from the top.
6. Run the XQuery. When the XQuery is executed, the result is shown in the **Result tree** tab, as illustrated in the following screen shot:



The XQuery debug output can be seen in the **Query debug** tab, as illustrated in the following screen shot:



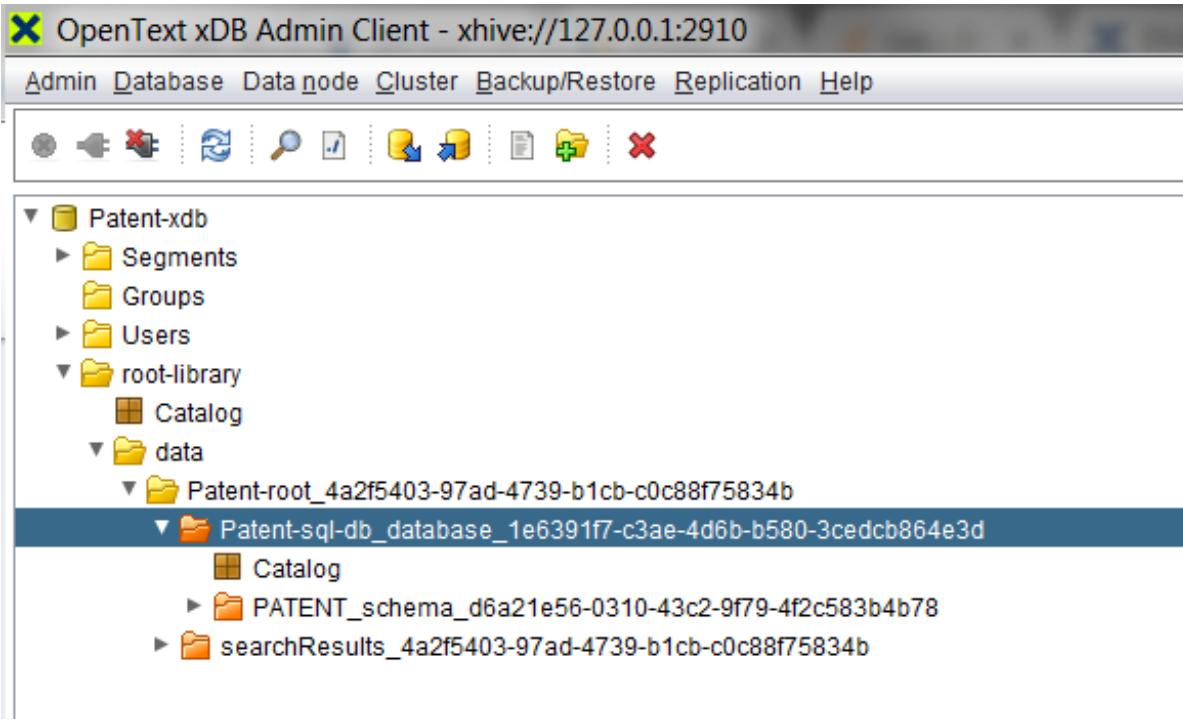
## Finding the XQuery that is Run for a Search

For table archiving, you can find the XQuery of a search by opening it in the Query Editor. For both table and SIP archiving, you can find the XQuery of the executed search in the IA Server logs, which, by default, can be found in <INFOARCHIVE\_ROOT>/logs/\*.log.

## Selecting the Context of an XQuery

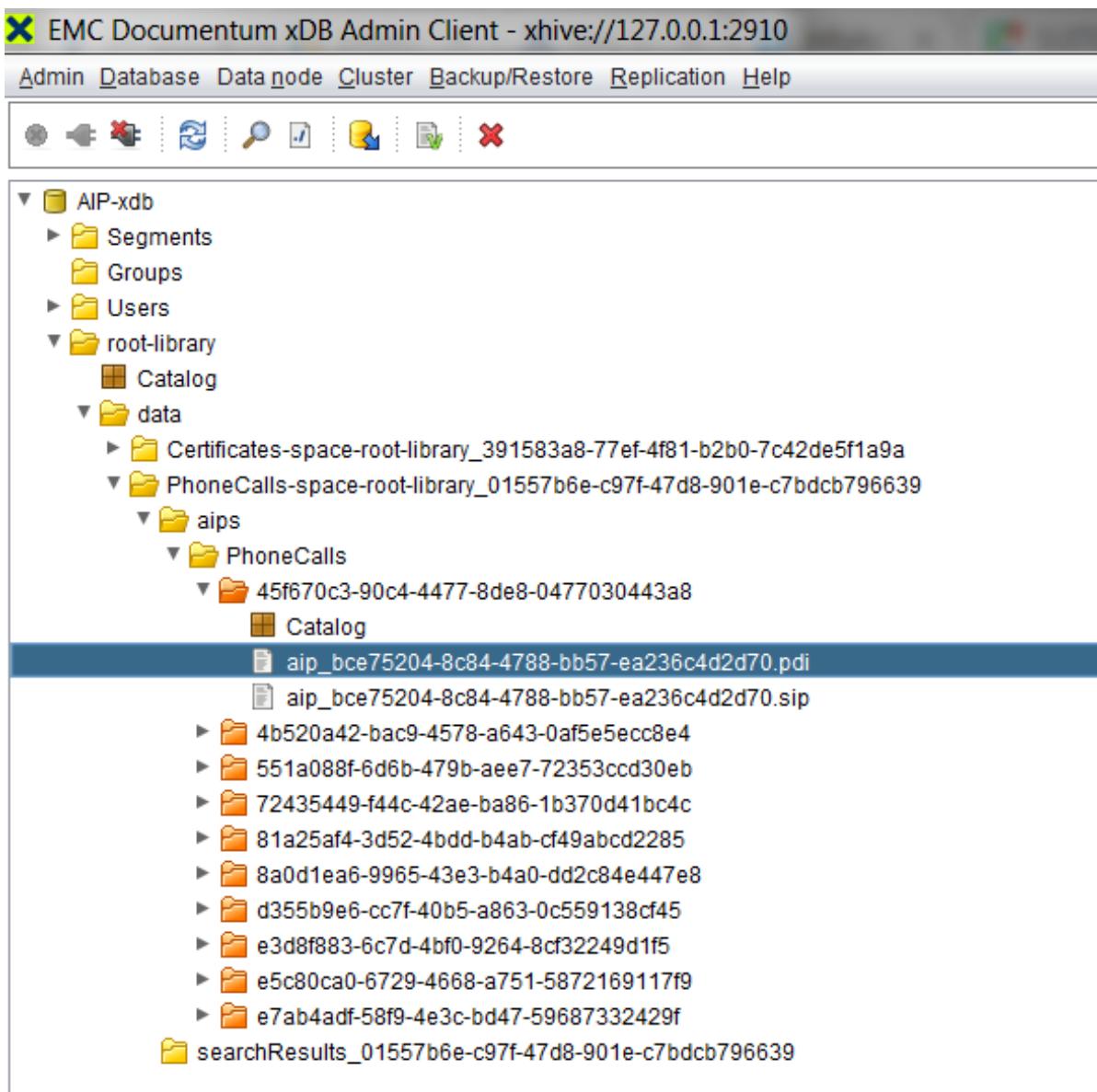
For table archiving, the library where the indexes are located is the schema library. This is the library with 'schema' in the name. For instance, when using the Patent application, this is the library that starts with 'PATENT\_schema\_'. To test index usage, this library can be used.

When a search is executed in InfoArchive, the parent library of the schema library is used as context. This is the library with 'database' in the name. This library is used as context to enable searches over multiple schemas. To test the performance of an XQuery executed with IA Web App, this library should be used.



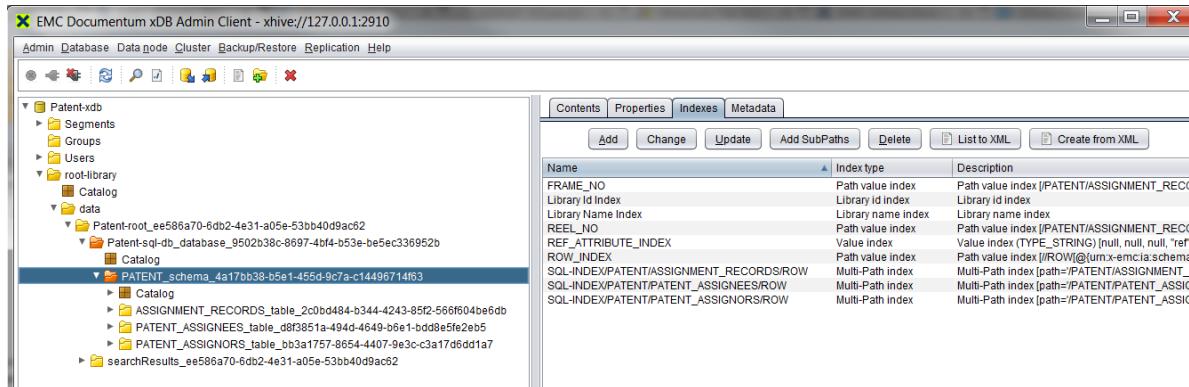
For SIP archiving, the XQuery uses a URI to identify the context (for instance, for the PhoneCalls example, this is collection('urn:eas-samples:en:xsd:phonecalls.1.0')). During XQuery execution, an InfoArchive resolver class is used to resolve this URI, which is not available in the xDB Admin environment. For this reason, remove this collection part from the XQuery.

To test index usage, you can use any pdi document under the aips/[applicationName] library as context. The indexes are stored on the pdi documents. To test the performance of an XQuery executed with IA Web App, use the aips/[applicationName] library.



## How Can I See the Indexes of My XQuery Context?

To view the indexes, select the library or document where the indexes are located in the tree-view on the left side and click the indexes tab in the panel on the right side. The [select the XQuery](#) section explains which library or document contains the indexes for table and SIP archives.



## Testing Form Input in the xDB Admin Tool

SIP XQueries are generated with form input.

For table archiving XQueries, the Form input is set by using external variables. To run these XQueries in the xDB Admin client, the external variables must be set to the corresponding form input.

The value of an external variable corresponding to form input is always an element with the name of the external variable containing the actual value as inserted into the form by the user.

So, if the user has inserted value 1 into the form of the Record Id Search of the Patent application, the actual value set in the XQuery matches the default value of this external variable:

```
declare variable $assignmentRecordId external := <assignmentRecordId>1</assignmentRecordId>;
```

## Interpreting XQuery Debug Output

The XQuery debug output contains a lot of information on how an XQuery is executed. The most important information is the index usage of the XQuery.

### How can I see if my XQuery is using an index?

You can see if an index is used by looking for the query plan in the debug output. For example, the following debug output shows that an index with name 'SQL-INDEX/BASEBALL/MASTER/ROW' is used:

```
query:1:14:Using query plan:
query:1:14:index(SQL-INDEX/BASEBALL/MASTER/ROW)
query:1:14:Looking up "(Miller, 1820-01-01T00:00:00, 1994-12-31T00:00:
00)" in index "SQL-INDEX/BASEBALL/MASTER/ROW"
```

## How can I see if an XQuery is using an index in the most optimal way?

Except in some unusual cases, the most optimal index usage is when the complete path expression is covered by one single index. Unfortunately, especially when using Path value indexes, this optimal usage is not always possible.

The following illustrates the debug output of an XQuery:

```
declare option xhive:index-debug 'true';
declare option xhive:queryplan-debug 'true';
declare option xhive:pathexpr-debug 'true';

declare namespace n="urn:acme-corp:xsd:certificates.1.0";
declare namespace ia-pdi="urn:x-emc:ia:schema:pdi";
declare namespace ia-fun="urn:x-emc:ia:functions";

let $aius_0 :=
 for $aiu in /n:Certificates/n:Certificate
 where
 $aiu/n:HeaderData/n:CreatedDate[. >= xs:date('2011-01-01T00:00:00Z') and
 . < xs:date('2013-01-01T00:00:00Z')]
 and $aiu/n:BusinessAddressDetails/n:TradingName[. = ('JRAMICK')]
 return $aiu
return (<results resultSetCount="{}(count($aius_0))" />, for $aiu in
($aius_0) return $aiu)
```

It is optimized with Path value indexes with the following definition:

```
Index name: CreatedDate
Index path: /{urn:acme-corp:xsd:certificates.1.0}Certificates/{urn:acme-corp:xsd:certificates.1.0}Certificate[{urn:acme-corp:xsd:certificates.1.0}HeaderData/{urn:acme-corp:xsd:certificates.1.0}CreatedDate<DATE_TIME>]

Index name: TradingName
Index path: /{urn:acme-corp:xsd:certificates.1.0}Certificates/{urn:acme-corp:xsd:certificates.1.0}Certificate[{urn:acme-corp:xsd:certificates.1.0}BusinessAddressDetails/{urn:acme-corp:xsd:certificates.1.0}TradingName<STRING>]
```

When executing the XQuery with these indexes, the following debug output is shown:

```
query:13:14:Found index "CreatedDate"
query:13:14:Found index "TradingName"
query:13:14:Using query plan:
query:13:14:index(TradingName)[. in index(CreatedDate)]
query:13:14:Looking up "JRAMICK" in index "TradingName"
query:13:14:Looking up "[2011-01-01T00:00:00,2013-01-01T00:00:00]" in index "CreatedDate"
```

The debug output shows the optimizer has chosen a query plan where both indexes are intersected.

In this case, every result of index(TradingName) must be matched with the index lookups of index(CreatedDate). If you see this kind of debug output and the XQuery does not perform well, it is recommended to run the XQuery again by disabling one of the indexes. Very often, the performance is best when index intersection is prevented, particularly if the first index is very selective and the second index is not very selective.

To disable the usage of index 'CreatedDate', the following declaration can be placed at the top of my XQuery:

```
declare option xhive:ignore-indexes 'CreatedDate';
```

After running the XQuery again, the debug output now shows a query plan where one condition of the path expression is optimized and the second part is not:

```
query:14:14:Found index "TradingName"
query:14:14:No index found
query:14:14:Using query plan:
query:14:14:index(TradingName) [child:::{urn:acme-corp:xsd:certificates.1.0}HeaderData/child:::{urn:acme-corp:xsd:certificates.1.0}CreatedDate[(. >= ... and . < ...)]]
query:14:14:Looking up "JRAMICK" in index "TradingName"
```

Behind `index(TradingName)`, the expression is shown that is executed in a non-optimized way. The query plan optimized by indexes always has an index part and, optionally, additional steps: `index(name)[additional steps]`. The query plan with only the index part has the most optimal index usage.

## How Can I Learn More About the Decisions of the XQuery Optimizer?

By putting the following debug option at the top of your XQuery, debug output of the XQuery optimizer is gathered:

```
declare option xhive:optimizer-debug 'true';
You can view this debug output in the Optimizer Debug tab.
```

## Building Up an XQuery Step-By-Step

If your XQuery does not give the expected results, the xDB Admin tool can be used to break down your XQuery step-by-step. Usually XQueries can become very complex and, consequently, become hard to debug. The best way to investigate the problem is by first creating a simple XQuery that works, then extend your XQuery step-by-step until you find the problem.

In the xDB Admin, you can easily try new XQueries in a separate tab. This allows you to preserve multiple versions of your XQuery without overwriting them. For instance, if you are debugging the following table XQuery:

```
declare namespace table="urn:x-emc:ia:schema:table";

declare variable $lastName external := "Miller";
declare variable $bats external := "";
declare variable $debut external := <debut><from>1820-01-01</from><to>1994-12-31</to></debut>

declare function local:addClause($whereClause as xs:string, $var as xs:string*, $expr as xs:string) as xs:string
{
 if (empty($var) or $var = "") then $whereClause
 else if ($whereClause = "") then $expr else concat($whereClause, " and ", $expr)
};

declare function local:getDateTime($date as xs:string) as xs:string {
 if (contains($date, 'T')) then $date else concat($date, 'T00:00:00')
};
```

```

let $whereClause := local:addClause("", $lastName, concat("$elem/NAMELAST = ''",
$lastName, ""))
let $whereClause := local:addClause($whereClause, $bats, concat("$elem/BATS = ''",
$bats, ""))
let $whereClause := local:addClause($whereClause, $debut/from, concat("$elem/DEBUT >
xs:dateTime('', local:getDateTime($debut/from), '')"))
let $whereClause := local:addClause($whereClause, $debut/to, concat("$elem/DEBUT
< xs:dateTime('', local:getDateTime($debut/to), '')"))

let $whereClause := if ($whereClause != "") then concat("where ", $whereClause)
else $whereClause

let $query-str := concat("for $elem in /BASEBALL/MASTER/ROW ", $whereClause, "
return $elem")

let $main-query := xhive:evaluate($query-str)

for $elem in $main-query

let $fname := $elem/NAMEFIRST/text()
let $lname := $elem/NAMELAST/text()
let $debut := $elem/DEBUT/text()
let $pBats := $elem/BATS/text()

return
<row id="{string($elem/@table:id)}">
<column name="lastName">{ $lname }</column>
<column name="firstName">{ $fname }</column>
<column name="debut">{ $debut }</column>
<column name="bats">{ $pBats }</column>
</row>

```

If the construction of the XQuery string is giving problems, then it is best to first print this xXQuery string and try to run it separately. You can do this by using this part of the XQuery:

```

declare variable $lastName external := "Miller";
declare variable $bats external := "";
declare variable $debut external := <debut><from>1820-01-01</from><to>1994-12-31</to></debut>;
....
let $whereClause := local:addClause("", $lastName, concat("$elem/NAMELAST = ''",
$lastName, ""))
....
let $whereClause := if ($whereClause != "") then concat("where ", $whereClause) else
$whereClause

return concat("for $elem in /BASEBALL/MASTER/ROW ", $whereClause, " return $elem")

```

It returns the following XQuery string:

```

for $elem in /BASEBALL/MASTER/ROW where $elem/NAMELAST = 'Miller' and
$elem/DEBUT > xs:dateTime('1820-01-01T00:00:00') and $elem/DEBUT
< xs:dateTime('1994-12-31T00:00:00')
return $elem

```

## What to do with a Slow XQuery

A very common situation is:

- The data set is very large.
- It is unclear how big the expected size of the XQuery result is.
- The XQuery is very slow and I do not want to wait until it finishes to check if indexes are used.

The best way to find out if the XQuery is slow because of the big result set is by wrapping your XQuery in a fn:subsequence function to reduce the result set. For instance, if your XQuery is:

```
for $r in /BASEBALL/MASTER/ROW
where $r/DEBUT > xs:dateTime('1950-01-01T00:00:00') and $r/DEBUT
< xs:dateTime('1990-01-01T00:00:00')
return $r
```

You can make this XQuery return only one result by using XQuery:

```
fn:subsequence(for $r in /BASEBALL/MASTER/ROW where $r/DEBUT >
xs:dateTime('1950-01-01T00:00:00') and $r/DEBUT < xs:dateTime('1990-01-
01T00:00:00') return $r, 1, 1)
```

The debug output should show if indexes are used.

## Limitations of xDB Admin Tool

InfoArchive-specific classes, such as analyzers, uri resolvers or encryption functions, are not supported. It is not possible to run an XQuery in parallel with the xDB Admin client.

## Troubleshooting

Issue	Possible Resolution
My search is running too long and consuming a lot of CPU time.	Refer to the <a href="#">Using Partition Keys</a> section of this guide in case you have not defined the partition keys.
Why does my XQuery not compile in the xDB Admin client?	<p>Usage of InfoArchive classes not in the xDB Admin classpath.</p> <p>Usage of the collection('uri') part in a SIP archiving XQuery (refer to <a href="#">Selecting the Context for an XQuery</a> for more information).</p> <p>Usage of encryption functions in both SIP and table archiving (refer to <a href="#">Limitations of xDB Admin Tool</a> for more information).</p>

Issue	Possible Resolution
Why is my XQuery unexpectedly slow?	<p>An index is not used.</p> <p>Index usage is not optimal (refer to <a href="#">The Selection Done by the Optimizer is not Always the Best</a> for more information):</p> <ul style="list-style-type: none"> <li>• only part of the conditions is optimized</li> <li>• index intersections</li> <li>• bad index selection by optimizer</li> </ul> <p>The XQuery result is very large. Refer to <a href="#">Reduce the Size of the (Intermediate) XQuery Result</a> for more information.</p> <p>An unordered declaration was used. Refer to <a href="#">Usage of an Unordered Declaration</a> for more information.</p> <p>Debug options produce a lot of output.</p> <p>For SIP archiving, there may be too many AIPs. Refer to <a href="#">Limit Size of XQuery Context</a> for more information.</p> <p>For table archiving, the index creation is still in process (refer to <a href="#">How Can I See the Indexes of My XQuery Context?</a> for more information).</p> <p>Usage of the wrong XQuery context (refer to <a href="#">Selecting the Context of an XQuery</a> for more information).</p> <p>For table archiving, as the schema has hundreds of tables, the optimizer has to inspect a very large number of indexes in the search context.</p> <p>For table archiving, : the XQuery contains a table join while the Archival Collection of the search is a table.</p>

Issue	Possible Resolution
Why is my index not used?	<p>Typed index field and search field do not match (refer to <a href="#">Search for Typed Data</a> for more information).</p> <p>Using Value comparison index for Full-Text condition/Using Full-Text index for Value comparison condition (refer to the <a href="#">Full-Text Search Versus Value Comparison</a> for more information).</p> <p>Usage of the wrong XQuery context (refer to see <a href="#">Selecting the Context of an XQuery</a> for more information).</p> <p>Namespace errors.</p>
Why does my Full-Text XQuery produce unexpected results?	Full-text does not work for special characters (refer to <a href="#">Full-Text Search Versus Value Comparison</a> for more information).

## Further Reading

Refer to the following documentation for more information about XQuery:

- <https://www.w3.org/TR/2017/PR-xquery-31-20170117/>
- <https://www.w3.org/TR/xpath-functions-31/>
- <https://www.w3.org/TR/xpath-full-text-30/>
- A [white paper](#) about XQuery optimization in xDB
- *Documentum xDB [Version Number] Manual*

# Chapter 10

---

## Appendix B – Using Metrics to Improve Performance

If a holding is properly configured, it will be able to handle a lot of data. If performance is poor, it is most likely because of a bad design decision in the holding configuration.

Most performance limitations actually derive from external factors rather than InfoArchive architecture. External factors include the operating system being used, the file transfer method, JRE, network, storage, etc.). Obviously, there are also differences between structured and unstructured data. For instance, an application dedicated to active archiving, which generally has an hourly SLA, is different from an application dedicated to a decommissioned application, in which injection will be a “one-shot”.

These types of business factors need to be included when configuring a holding.

Customer performance issues are either:

- The ingestion speed is slow; or
- The search speed is slow.

Follow these important rules to avoid performance issues:

- Capture customer use case before creating of application:
  - What is archiving type: active archiving or application decommissioning?
  - What is the application type: table-or SIP-based?
  - How many SIPs per day do you expect to have?
  - How many AIUs per SIP do you expect to have?
  - What is the retention date range? How long data should be stored?
  - What is the average volume of the SIP package? (Mb)
  - What is the average volume of the pdi .xml inside the SIP package? (Mb)
- If not required, do not use encryption/decryption in an application's configuration.
- Perform performance testing before going to production.
- Performance adaptation is an iterative process. In other words, you need to create an application configuration and test the performance on a large sample SIP. Then, based on the log's metrics, modify the application configuration to improve performance.

# Improving Performance for a SIP Archive

In general, the ingestion and search performance depends on the application configuration, holding configuration and the following external IT factors:

- The number of AIUs per SIP
- Ingestion mode
- Indexes
- Partitioning keys
- Load balancing topology
- External IT factors

It is recommended to choose the size of the SIP by always trying to get the maximum in it.

These are not, however, technical limitations. These guidelines create the correct balance between trying to store everything in one SIP and storing each AIU in its own SIP.

Although InfoArchive's multi-thread ingestion and partition mechanism aims to have multiple SIPS, the larger the SIP in AIUs and total volume, the better the performance.

The following table includes recommendations and performance testing metrics:

Name	Comments
PDI file size	<ol style="list-style-type: none"><li>1. Might be the limit of the size of a document in xDB.</li><li>2. Configure cache-out to keep only the amount of data you want into xDB.</li></ol>
SIP file size	<ol style="list-style-type: none"><li>1. Can be limited by the customer environment, as the SIP might have to be copied through its network.</li><li>2. The SIP file with structured content is less than 1Gb is a common case that uses batch ingestion.</li><li>3. Ingestion time is not a criteria when using batch ingestion as long as it does not run for more than 1-2 hours for a large file. The work done at ingestion is partially done to improve search and retrieval times.</li><li>4. 10 Mb SIP is a small-sized SIP. For batch ingestion, 1Gb SIP for unstructured data and 100Mb SIP for structured data are not uncommon.</li></ol>

Name	Comments
Number of AIU in a SIP	<ol style="list-style-type: none"> <li>1. Having one million records is not uncommon. On such a use case, the attention is put on being able to search over this data. As partitioning is done at the AIP level, one query will at least be run on an AIP, meaning one millions records. That means the holding must be designed to use strong partitioning and the search form must have some mandatory fields that ensure that synchronous search will use portioning and indexes.</li> <li>2. Having one item can slow down the search, and having 10 to 40 is still too few. Several tens of thousands of AIUs is adequate for structured content. Hundreds of thousands is not uncommon for pure unstructured data. This is only acceptable when doing synchrnous ingestion and using aggregate mode, when the AIPs will be aggregated into larger ones overtime.</li> </ol>
Number of SIP files in a holding	The only limitation is how long it will take to run the partitioning query. If indexes are positioned on the partitioning keys, 100,000 SIP files can work. For example, a standard use case is to have 5-10 SIPs by day on a holding. The retention period is 10 years. In this case, the holding could have 36,000 AIPs.

To improve SIP application search performance, try the following:

- Ensure that the indexes and partitioning keys for the search fields are defined
- Review the number of AIUs in an AIP. The application and holding configuration depends on number of AIUs per AIP. A low number (1 to 10,000) of AIUs per AIP can slow down the search if the PRIVATE ingestion mode was selected. Consider using the AGGREGATE mode if this is critical for your data.
- Ensure you use the optimal ingestion mode (private, aggregate or pooled).
- Check the partitioning key strategy. InfoArchive search is a two-tiered search. The first tier selects a subset of AIPs based on the defined partitioning keys. The second tier executes XQuery on the selected AIPs. It is optimal if the first tier selects around 200 to 300 AIP items in a synchronous search so that the second tier is executed quickly. In case there are more items that are supposed to be returned, consider an asynchronous (background) search.

In other words, when defining the partitioning keys, it worth having estimated the amount of AIPs returned by the first tier for the given search criteria.

- Ensure that structured data encryption is not used without need.
- Avoid using full-text search unless it is required, as it is slow by definition.
- Enable DEBUG log level for all components (services level and UI level) and repeat the ingestion or the search. Investigate the timing. The logs contain timing metrics. Based on the timing, you can pinpoint the part of application configuration that must be improved.

- Take XQuery generated by InfoArchive for the search and try to execute it with xDB Admin directly. See if indexes are taken into account. Use the xDB profiling tool.
- The AIPs are offline so it takes longer to bring them online.

## Improving Ingestion Speed

Ingestion speed depends on:

- The SIP size, including the percentages of structured and unstructured data;
- xDB indexes. Many XQueries are executed by different ingestion processors so xDB indexes have to be set up;
- Amount of AIUs per SIP;
- Network speed;
- Target storage; and
- Working directory storage.

The IA Server logs, which, by default, can be found in <INFOARCHIVE\_ROOT>/logs/\*.log., help you to determine how best to improve performance. For example:

- If customer does not require a crypto staff, then remove/disable all the encryption configurations.
- If customer does not require CI hashing, then remove/disable CI hashing configuration in the pdi.xml file or use a MD5 hash instead of a SHA-1.

```
<data>
 <id>ci.hash</id>
 <select.query>
 <![CDATA[
 return
]]>
 </select.query>
</data>
```

- If you have XQuery in some processors, ensure that the xPath is indexed.
- Use a Solid State Drive (SSD) for the working directory (change the path in the application.yml file).

## Improving Search Speed for a SIP Archive

The best method to improve search speed is to use the correct ingestion mode. Also, ensure that partitioning keys and indexes are used.

Refer to [Search Troubleshooting](#) for tips on how to improve the speed of a SIP search.

# Improving Performance for a Table Archive

Typically, a table is more “traditional”. More data means slower injection and, therefore, a slower search response with an impact on indexes.

It is necessary to understand search is performed in two steps:

1. Accessing the actual DOM element and retrieving the values with XQuery from xDB.
2. Appending the result node to the resulting DOM document where we cache our search results.

If searches conducted in a table archive are slow, complete the following to improve table application search performance:

- Make sure indexes are used;
- Try to reduce the amount of search result set (the bigger result set, the slower the search speed because of number 2 step above).
- Try to reduce the amount of result columns (the more columns in the result list, the slower the search will be because of the number 2 step in search).
- Review load balancing topology.
- Review external IT factors.



# Chapter 11

## Appendix C – Mapping XSD Data Types

The XML Schema Definition (XSD) specifies how to describe the elements in an XML document.

Use an advanced XML editor to validate an XML file against XSD.

The first column of the following table outlines the Java types that IA Server supports while the second column outlines the schema types that xDB supports:

Java Types Supported by the IA Server	Schema Types Supported by xDB
STRING, INTEGER, LONG, DOUBLE, DATE, DATE_TIME, BIGINTEGER, FLOAT, BIGDECIMAL	STRING, INT, LONG, DOUBLE, FLOAT, DATE, DATE_TIME, INTEGER and DECIMAL

When the XSD defines another type, we need to switch to one of them.

The following table outlines the mapping between xDB type support and InfoArchive type support:

xDB Type	IA type	Description
STRING	STRING	Any set of character values, including Boolean values: true, false, 0, 1
INT	INTEGER	32 bit signed type: between -2^31 and 2^31 - 1 (from -2 147 483 648 to 2 147 483 647)
LONG	LONG	64-bit signed type: between -2^64 and 2^64 - 1 (from -9 223 372 036 854 775 808 to 9 223 372 036 854 775 807)
INTEGER	BIGINTEGER	No theoretical limit.
FLOAT	FLOAT	A single-precision 32-bit IEEE 754 floating point.
DOUBLE	DOUBLE	A double-precision 64-bit IEEE 754 floating point.

xDB Type	IA type	Description
DECIMAL	BIGDECIMAL	Arbitrary-precision signed decimal numbers. A BigDecimal consists of an arbitrary precision integer unscaled value and a 32-bit integer scale (unscaledValue × 10-scale).
DATE	DATE	Defines a date value. e.g.: 2017/12/12
DATE_TIME	DATETIME	Defines a date and a time value e.g.: 2002-11-18T16:35:42.104+01:00

The following table outlines the mapping between XSD types and InfoArchive types:

Schema Type	IA Type	xDB Type	Description
ENTITIES	STRING	STRING	
ENTITY	STRING	STRING	
ID	STRING	STRING	A string that represents the ID attribute in XML (only used with schema attributes)
IDREF	STRING	STRING	A string that represents the IDREF attribute in XML (only used with schema attributes)
IDREFS	STRING	STRING	
language	STRING	STRING	A string that contains a valid language id
Name	STRING	STRING	A string that contains a valid XML name
NCName	STRING	STRING	
NMTOKEN	STRING	STRING	A string that represents the NMTOKEN attribute in XML (only used with schema attributes)
NMTOKENS	STRING	STRING	
normalizedString	STRING	STRING	A string that does not contain line feeds, carriage returns, or tabs
QName	STRING	STRING	
string	STRING	STRING	A string
token	STRING	STRING	A string that does not contain line feeds, carriage returns, tabs, leading or trailing spaces, or multiple spaces
date	DATE	DATE	Defines a date value

Schema Type	IA Type	xDB Type	Description
dateTime	DATETIME	DATE_TIME	Defines a date and time value
duration	STRING	STRING	Defines a time interval
gDay	INTEGER	INT	Defines a part of a date - the day (DD)
gMonth	INTEGER	INT	Defines a part of a date - the month (MM)
gMonthDay	STRING	STRING	Defines a part of a date - the month and day (MM-DD)
gYear	INTEGER	INT	Defines a part of a date - the year (YYYY)
gYearMonth	STRING	STRING	Defines a part of a date - the year and month (YYYY-MM)
time	STRING	STRING	Defines a time value
byte	INTEGER	INT	A signed 8-bit integer
decimal	BIGDECIMAL	DECIMAL	A decimal value
int	INTEGER	INT	A signed 32-bit integer
integer	BIGINTEGER	INTEGER	An integer value
long	LONG	LONG	A signed 64-bit integer
negativeInteger	BIGINTEGER	INTEGER	An integer containing only negative values (..,-2,-1)
nonNegativeInteger	BIGINTEGER	INTEGER	An integer containing only non-negative values (0,1,2,..)
nonPositiveInteger	BIGINTEGER	INTEGER	An integer containing only non-positive values (..,-2,-1,0)
positiveInteger	BIGINTEGER	INTEGER	An integer containing only positive values (1,2,..)
short	INTEGER	INT	A signed 16-bit integer
unsignedLong	LONG	LONG	An unsigned 64-bit integer
unsignedInt	INTEGER	INT	An unsigned 32-bit integer
unsignedShort	INTEGER	INT	An unsigned 16-bit integer
unsignedByte	INTEGER	INT	An unsigned 8-bit integer
anyURI	STRING	STRING	The anyURI data type is used to specify a URI
base64Binary	STRING	STRING	Binary data types are used to express binary-formatted data
boolean	STRING	STRING	The boolean data type is used to specify a true or false value
double	DOUBLE	DOUBLE	A double-precision 64-bit IEEE 754 floating point.

Schema Type	IA Type	xDB Type	Description
float	FLOAT	FLOAT	A single-precision 32-bit IEEE 754 floating point.
hexBinary	STRING	STRING	Binary data types are used to express binary-formatted data
NOTATION	STRING	STRING	
QName	STRING	STRING	

# Chapter 12

---

## Appendix D – Custom SIP Format Support

### Goals

By default, InfoArchive is able to receive and to ingest a SIP corresponding to the SIP ZIP format. With EP4, it's now possible to accept custom SIP formats by adding a custom implementations to support them.

With EP4, three new public APIs have been introduced to handle custom formats during the **reception**, the **ingestion** and the **content retrieval**.

### Reception

To receive a custom SIP package, a new class based on the public interface `SipReceptionHandler` needs to be implemented. When it was done, the jar containing the implementation must be added in the server class path and registered at the Receiver Node level.

### Create a new implementation based on the interface `SipReceptionHandler`

The contract is simple. From the SIP file, it's require to generate a SIP object. The SIP object is a JAXB object based on the `ia_sip` XML Schema. A context is set to provide the **working directory** and a **logger** before to call the extract method. It's possible to associate one or more contents to the receiver node, these contents are accessible during the extraction with the method `Context.downloadConfigurationContent`.

### Configure the reception to handle the new SIP format

- Edit the `configuration.yml`
- Declare the new SIP format (`sip_custom` for example) with the custom implementation in the receiver node configuration object
- It's possible to attach one or more contents (optional)

```
receiverNode:
 name: receiver_node_01
 sips:
```

```
- extractorImpl: com.emc.ia.reception.sip.extractor.impl.LegacyZipSipExtractor
 format: eas_sip_zip
- extractorImpl: com.emc.ia.reception.sip.extractor.impl.ZipSipExtractor
 format: sip_zip
- extractorImpl: com.acme.ia.reception.SipCustomReceptionHandler
 format: sip_custom
content:
- format: myformat
 mimeType: application/octet-stream
 resource: myconfigurationfile.myformat
```

**Note:** If the custom SIP format is a ZIP file containing a standard SIP descriptor, a custom implementation is not required. It is just necessary to declare the new format with the default handler : ZipSipExtractor.

## How to receive/ingest a SIP with a specific format with the CLI

An extra parameter - format - must be provided to specify the SIP format. If the format is not provided, the default format **sip\_zip** is used.

```
connect --u sue@iacustomer.com --p password
cd applications/MyApplication
ingest --from /tmp/mycustomsipzip.zip --format sip_custom
```

## Ingestion

To ingest a custom SIP package, a new class based on the public interface **SipIngestionHandler** needs to be implemented. When it was done, the jar containing the implementation must be added in the server class path and registered at the PDI XML configuration file level.

## Create a new implementation based on the interface **SipIngestionHandler**

When the method **extract** is called the original SIP file is provided. To be valid, the implementation needs to register the **PDI XML** file and the **folder** containing the unstructured contents. To help the extraction, a specific configuration can be retrieved from the **ConfigElement** object and/or from a content attached to the PDI configuration object. A context is set to provide the *working directory* and a *logger* before to call other methods.

## Configure the ingestion to handle the new SIP format

- Need to declare the custom implementation to use for the new format in the PDI configuration
- Edit **data-model-config/pdi-InvoicesPdi.xml**
- Add a new data element with the id **sip.extract** and include custom configuration to drive the transformation

```
<datas>
 <data>
 <id>sip.extractor</id>
 <pdi_name>eas_pdi.xml</pdi_name>
 <sip_name>eas_sip.xml</sip_name>
 <handler format="sip_custom" handler="com.acme.ia.ingestion.SipCustomIngestionHandler">
 <what_you_want/>
 <configuration_resource>custom_resource_name</configuration_resource>
 </handler>
 <data>
</datas>
```

- Edit the YML script to import external resources

```
pdi:
 name: Invoices-pdi
 content:
 - format: xml
 resource: data-model-config/pdi-Invoices.xml
 - format: custom_resource_name
 mimeType: application/octet-stream
 resource: data-model-config/custom_resource_name.zip
```

NB: The optional external resources are accessible on demand with the **context** object and the method **downloadConfigurationContent**

## Content Retrieval

When the content needs to be transformed before to be returned to Emma, it's possible to ask InfoArchive to intercept the input stream and to pass it to a custom handler. The custom handler needs to be registered in an Access Node object.

## Create a new implementation based on the interface ContentDownloadHandler

This interface allows to intercept the record content before returning it to Emma. This is the perfect place to transform the content at the runtime. The InputStream provided is seekable. The new mime-type, filename and/or size needs to be set via the Context object.

During the process, it's possible to access to a configuration file previously attached to the **AccessNode** object with the method **Context.downloadConfigurationContent()**.



**Caution:** The previous interface **ContentHandlerCustom** is deprecated and the new one needs to be used in priority.

## Configure the server to perform the transformation

- A new object **AccessNode** has been introduced to declare the implementation to use during the content retrieval.
- It's possible to attach one or more configuration files to read them during the content retrieval.

```
accessNode:
 name: default-access-node
 ciHandlers:
 - mimeType: application/pdf
 handlerImpl: com.acme.ia.content.handler.PdfContentDownloadHandler
 content:
 - format: custom_resource_zip
 mimeType: application/octet-stream
 resource: custom_resource.zip
```

# Chapter 13

---

## Appendix E – Glossary and Acronyms

The following table contains short definitions for InfoArchive resources, processes and mechanisms:

Name	Acronyms and Related Terms	Description
Active Archiving		An archiving process whereby data is ingested into an archive on a time-related basis (for example, per hour, per day, per week, etc.).
Active Directory	AD	One of the authentication mechanisms supported by InfoArchive.
Application		A logical configuration object in an archive system that presents the customer business item for preserving and storing the data. The following is the logical archive object hierarchy: Tenant -> Application -> Holding. An application can be one of the following types: SIP or table.
Application Decommissioning		An archiving process whereby data is extracted from a legacy application and ingested into an archive system to reduce the cost of supporting the legacy application.
Archival Information Package	AIP (Package)	An archive resource that represents the package of business information inside InfoArchive. It may contain from zero to multiple structured data elements, which are represented as XML, and/or zero to multiple unstructured data elements.
Archival Information Unit	AIU (Record)	An information atom. The smallest archival unit of an information package. An AIP contains AIUs.
Archive Information Collection	AIC	Search configuration resource that contains a set of criteria to be used during a search. It organizes a set of AIPs that support flexible and efficient data access.

Name	Acronyms and Related Terms	Description
Audit		<p>An audit indicates that a particular action has occurred. A particular audit is associated to an event type that defines the action that occurred (for example AIP / DISPOSE indicates that an AIP was disposed). Audits can be searched from the audit application after running the Ingest Audits job.</p> <p>Audits can be associated to the system, a tenant, or a particular application,</p>
Audit Event Type		An audit event type is a pairing of a type and name that act as a mechanism for enabling an audit. For example, it is possible to enable a DISPOSE event for a TABLE and an AIP, as each one is an audit event type.
Background Search		A background search is a search that runs asynchronously, and is associated to a background task. A search may need to run in the background if the number of results returned is large, or if the content is offline. A user can request that a search be run in the background.
Backup		Copying the data in a recovery mechanism that allows data to be restored in the event of data loss or corruption. This copy of the data is also called as "backup".
Batch Ingestion		A SIP ingestion approach that consists of three steps: Reception, Enumeration and Ingestion. When ingesting SIPs in a batch, it is required to first receive all SIP packages and then ingest them into the application. Use the batch ingestion approach when there is a large set of SIPs that need to be archived. See also <a href="#">Direct Ingestion</a> .
Batch Processing		An approach used to improve the performance of the long-running operations and jobs, such as applying/removing holds and retention policies. With batch functionality, an operation is broken into smaller chunks. Even if there is a small number of items to process, at least one batch is created.
Binary Large OBject	BLOB	The xDB store approach that uses an xDB feature to call a BLOB node to save binary content.

Name	Acronyms and Related Terms	Description
Bucket		A storage configuration resource used within ECS and S3 storage systems.
Cache-In		An action on an AIP object to restore the object in an xDB library from backup. The opposite action is referred to as a Cache-out. Cache-in and cache-out provide the ability for SIP archiving applications to improve performance by reducing the number of libraries in xDB.
Cache-Out		An action on an AIP object to detach the object from an xDB library. The opposite action is referred to as a Cache-in. Cache-in and cache-out provide the ability for SIP archiving applications to improve performance by reducing the number of libraries in xDB.
Chain of Custody		A set of tests that check the integrity of the ingested tables in InfoArchive.
Collection		A resource that an e-Discovery Administrator can run a search against and associate a collection with the search result. Typically, the collection will be associated with a legal matter.
Confirmation		A message generated in reaction to an AIP event (lifecycle transition).
Content Information	CI	A piece of unstructured content that is associated with a record. For each CI, there is corresponding RI entry in the table of contents (RI.xml).
Content Item		Internally, unstructured content that is related to an InfoArchive configuration object is represented as a Content Item.
Custom Search		Allows a search designer more control on how to show the results of a search.
Data Submission Session	DSS	A delivery of media or a single telecommunications session that provides data to a consumer.
Database		Archive information resource that presents a database for table-based applications.
Declarative Configuration	DC	The way to describe set of configuration objects in YAML format. Declarative configuration is used to present either entire system information and/or particular application and/or holding configuration resources. The goal of declarative configuration is to replace the ANT configuration format of applications.

Name	Acronyms and Related Terms	Description
Delivery Channel		The configuration resource that defines a destination where to send search results.
Direct Ingestion		A SIP ingestion process that allows the archival of a single SIP in one request, simultaneously, avoiding "receive-enumerate-ingest" steps. Direct ingestion is used in case a single SIP is not archived frequently. Selection of the ingestion approach is done based on performance requirements. See also <a href="#">Batch Ingestion</a> .
Disposition		<p>The controlled process of removing data from the archive after the required aging period has elapsed (defined by the retention policies applied). Only items that are under retention go through the disposition process. The disposition process has the following steps:</p> <ol style="list-style-type: none"> <li>1. Put information into purge lists.</li> <li>2. Obtain approval to dispose list.</li> <li>3. Run the Disposition job, and then run the Clean job to clean up the space.</li> </ol> <p>Depending on the resource being disposed, additional steps may be required (for example, disposing AIPs require that the Confirmation job is executed).</p>
Dissemination Information Package	DIP	An archive resource that presents an information package that is returned to the user via search or some other retrieving operation.
Export Pipeline		An "xProc Pipeline" that is specifically intended to export InfoArchive search results.
File System Folder		The storage configuration object that represents a file system folder in which unstructured content is stored.
File System Root		Storage configuration object that represents the storage of a Local File System or an Isilon type, and indicates a root location on a disk.
Group		A group defines a set of users. Groups are used to restrict access to applications and searches.
Hold		A compliance configuration object. A hold is applied to an object to block deletion or disposition, either temporarily or indefinitely.

Name	Acronyms and Related Terms	Description
Holding		A logical configuration destination archive in which to ingest and store data that shares common characteristics. For example, you can create a holding to archive data from the same source application (such as ERP data), or of the same format (such as audio recordings), or belonging to the same business entity. The following is the logical archive object hierarchy: Tenant > Application > Holding.
Holding Composition		A configuration object used by the holding wizard to define a new holding.
Hold Set		A logical container that references items with a hold applied against it. A hold set is created when a hold is applied to one or more items.
IA Shell	CLI	The Command Line Interface for InfoArchive. It is a tool that provides the set of commands for the administrator to manage the InfoArchive product and its resources.
InfoArchive	IA	InfoArchive is an integrated product suite designed for application agnostic information management and archiving. It is an information management system that preserves, maintains, and controls continuing access to valuable enterprise information assets.
Ingest		Ingestion is the process that actually stores an AIP (for SIP-based applications) or table data (for table-based applications) into an archive so the data can become searchable.
Ingestion Mode	xDB Mode	The way business data is preserved in xDB. There are three types of ingestion modes supported by InfoArchive: PRIVATE, POOLED and AGGREGATE.
Ingestion Node		An archive process configuration resource that defines the parameters for the ingestion and enumeration processes.
Invalidation		"Invalidate" is an action on an AIP package. Invalidation is required in case an incorrect SIP was submitted, and the correct SIP must be submitted with the same identifier. After invalidating an AIP, it is immediately removed from a search's scope.

Name	Acronyms and Related Terms	Description
Job		A job defines a type of operation that can be done asynchronously. Jobs can be configured to either run on a schedule or run manually. Examples of jobs include the Clean job, Disposition job, and the Ingest Audits job.
Job Instance		Represents either a scheduled or running instance of a job. Some jobs may spawn order items.
Lightweight Directory Access Protocol	LDAP	One of the authentication mechanisms supported by InfoArchive.
Matter	Legal matter	A matter is associated with a legal hold. Matters allow an e-Discovery Administrator the ability to associate a collection to a matter so that the records cannot be deleted. Matters have a state that can control whether or not collections can be added to the matter, which effectively prevents all disposition of records in the search result.  It is also possible to define a matter string for a hold.
Nested Search		A search within a search that allow more flexibility for creating searches.
Open Archival Information System	OAIS (ISO 14721)	A reference model that a wide variety of organizations use for archiving digital information for long-term preservation. It specifies the format that data is ingested into, stored in, and retrieved from InfoArchive throughout the information's lifecycle.
OpenText Directory Services	OTDS	One of the authentication mechanisms supported by InfoArchive.
Order		A configuration object used by SIP applications to set permissions and retention duration when executing an Order Item (search).
Order Item	Background task	An order item represents an operation that is being processed asynchronously. Order items can be initiated by a user and can be viewed from the Background Tasks tab. Order items can also be created by the system for jobs. Order items may also be split into chunks called batch items (for very large operations, such as applying hold to search results).

Name	Acronyms and Related Terms	Description
Partitioning Keys	PKEY	The key to be used during the search. A partition key is used in the first tier of the query process to limit the data returned when a search is executed. Partition keys are created during ingestion using XQuery and are stored in the AIP object.
PDI Schema		A holding configuration resource that contains a reference to the XML schema to be used for validation of the PDI XML file during the ingestion.
Permission		Indicates which groups have access to either an application or a search set.
Preservation Description Information	PDI	A OAIS standard and related to the mandatory SIP package resource with business data to be put into archive. There is eas_pdi.xml file in the SIP package, which is a PDI descriptor with business data.
Purge Candidate List		A list that contains items that qualify for disposition. Purge candidate lists need to be approved before the items can be disposed. Purge candidate lists store a state and the Clean purge list jobs can be used to remove cancelled or disposed purge lists.
Query		A search configuration resource that defines options for building a query to retrieve records.
Query Quota		A search configuration resource that defines the maximum number of records to be returned in a search. Query quotas are specific to searches in SIP-based applications.
Reception		The process of transferring SIP packages to IA Server with preparation of SIP packages for ingestion. Usually in part of ingestion process with sequence "receive","enumerate","ingest". The Reception stage is used in "batch" ingestion process.
Reception Node		An archive process configuration resource that defines the parameters for the receiver process.
Record		A record is either an row in a table or an AIU in an AIP.

Name	Acronyms and Related Terms	Description
Reference Information	RI	Identifies and, if necessary, describes one or more mechanisms used to provide assigned identifiers for the CI. It also provides those identifiers that allow outside systems to refer to this particular CI.
Rejection		"Reject" is an action on the AIP package. Rejecting of an AIP is required to invalidate all AIPs that belong to the same collection. After rejecting an AIP, it is immediately removed from a search's scope.
Result Configuration Helper		A search configuration resource that defines a set of columns that can be used by the Result Master to help configuring the search result page.
Result Master		A search configuration resource that represents the result search page columns and tabs (in-line panels and side panels). Result masters are part of a search set.
Retention		A general term that indicates how long content should be kept for compliance. Retention is associated to items via applying a retention policy and for many types of the retention policy, a base date is specified.
Retention Policy		A compliance configuration object that specifies the rules for how long to retain the data.
Retention Set		A logical container that references data under retention, including whether or not items in the set are aging together and the type of items in set (for example, application, package, table or record).
Role		A role is mapped to set of actions that can be done in InfoArchive, such as run search, apply retention, or ingest content. Groups are mapped to roles. Roles are fixed by InfoArchive. Some examples include End User, Retention Manager, Administrator, Developer, and E-Discovery Administrator.

Name	Acronyms and Related Terms	Description
Rule		Rules can be defined to apply retention or holds to records or packages. Rules can be evaluated by running the associated jobs (Apply Retention via Rule, Apply Hold via Rule). A rule contains one or more individual rules using DROOLS. When defining a rule, the type of action the rule is for is defined. Rules are defined per application, and can be defined using ANT or declarative configuration.
Schema		An archive information resource that presents a schema for table-based applications. Schemas are associated with a database and contain one or more tables.
Search		The primary method used to access data that has been ingested by InfoArchive. The process of searching an archive to retrieve the set of AIUs that satisfy the search criteria.  At the same time there is a search configuration object in InfoArchive that is used to represent the search item for either a SIP or table application. A search contains one or more search compositions.
Search Composition	Search Set	A search configuration resource that manages the search components (XForm, result master, search, permissions) and is used to execute a search.
Space		A storage configuration object that represents the relation between storage and application. It is used by: Space Root Folder, Space Root Object and Space Root xDB Library.
Space Root Folder		A storage configuration object that represents the relation between a Space and a Local File System/Isilon storage systems.
Space Root Object		A storage configuration object that represents the relation between a Space and ECS/S3 storage systems.
Space Root xDB Library		A storage configuration object that represents the relation between a Space and an xDB Database.

Name	Acronyms and Related Terms	Description
Storage		A storage configuration object that contains a list of properties for target storage configuration. Storage systems can be one of the following types: Local File System, Isilon, ECS, S3, Centera and Custom Storage. A storage system holds data, such as unstructured content for records, library backups, raw XMLs, ingestion logs, etc.
Storage End Point Credentials		A storage configuration object that represents the user credentials used when establishing a connection to the target storage system. The object is used by storages of following types: ECS and S3.
Store		A storage configuration object that contains properties for linking a space with a File System Folder or Bucket. Stores holds records in a context of an application.
Submission Information Package	SIP	An archive resource that presents an information package before the ingestion process. It is a term from OAIS standard for input archive package.
Table		An archive information resource that presents a table for table-based applications. Tables are contained within schemas.
Tenant		A logical configuration object in archive system that presents a customer business item for preserving and storing the data. Tenants store zero or more applications.
Transformation		A configuration resource that defines the XQuery/XSLT to use to perform a transformation. Transformations are associated with holdings.
Universally Unique Identifier	UUID	An identifier that is used to identify InfoArchive resources. Every resource has its own and unique UUID.
Value List		A search configuration resource that identifies the list of possible values in an XML document. Value lists are used by search forms to externalize the values to avoid the search forms storing the information. Value lists are per application and can be used by multiple search forms.

Name	Acronyms and Related Terms	Description
xDB Database		A storage configuration resource that represents a database in xDB. It contains a set of properties to access physical database.
xDB Federation		A storage configuration resource that contains a set of properties to establish a connection with a physical xDB Federation. A federation is a container for Databases.
xDB Library		A storage configuration resource that represents a container stored within a database. xDB libraries are detachable and can be taken offline (cached out).
xDB Library Policy		An archive process configuration object that contains a set of properties that define when an xDB Library can be closed when ingestion in POOLED and AGGREGATE modes.
xForm		The search resource that represents a user form that contains search criteria on the UI level.
xProc	<a href="#">W3C Recomendation</a>	A W3C Recommendation that defines an XML transformation language to define XML Pipelines.
xProc Pipeline		An XML Pipeline specifies a sequence of operations to be performed on zero or more XML documents. Pipelines generally accept zero or more XML documents as input and produce zero or more XML documents as output. Pipelines are made up of simple steps which perform atomic operations on XML documents and constructs similar to conditionals, iteration, and exception handlers which control which steps are executed.  See also <a href="#">Export Pipeline</a> .
xQuery		A search configuration resource that contains the query to be used during the search for table-based applications.
	CID	An internal identifier for content information that can be associated to one or more records.