

## OpenText™ Information Archive

### **Configuration Guide**

Configure SIP and table applications. Design searches and configure how users will export search results. Learn how to use declarative configuration. Perform compliance-related tasks. Improve system performance.

EARCORE250400-CGD-EN-01

---

## **OpenText™ Information Archive**

### **Configuration Guide**

EARCORE250400-CGD-EN-01

Rev.: 2025-Sept-11

This documentation has been created for OpenText™ Information Archive CE 25.4.

It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

#### **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>

#### **© 2025 Open Text**

Patents may cover this product, see <https://www.opentext.com/patents>.

#### **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

<b>1</b>	<b>Introduction .....</b>	<b>15</b>
1.1	Archiving requirements questionnaire .....	16
1.2	Best practices: working with a staging environment .....	17
1.3	Using the Global Settings tab .....	18
1.4	Configuring CORS headers .....	18
<b>2</b>	<b>Managing an OpenText Information Archive application ....</b>	<b>21</b>
2.1	Exporting configuration from staging and importing into a production environment .....	22
2.1.1	Preparing the ZIP file for exported SIP application into another deployment .....	22
2.1.2	Preparing the ZIP file for an exported table application into another deployment .....	22
2.2	User context .....	23
2.3	Creating an application with the IA Web App .....	24
2.4	Setting an application's status to active .....	26
2.5	Editing applications .....	27
2.6	Deleting application data .....	28
2.7	Deleting an application .....	29
2.8	Configuring application storage with spaces .....	30
2.8.1	Creating a space in IA Web App .....	30
2.8.2	Editing a space in IA Web App .....	31
<b>3</b>	<b>Configuring a table application .....</b>	<b>33</b>
3.1	Configuring how data is stored in PostgreSQL .....	33
3.2	Configuring databases, schemas and tables .....	34
3.2.1	Table and column constraints .....	36
3.2.2	Unstructured content .....	39
3.2.3	Storing extracted text .....	40
3.2.3.1	Storing extracted text metadata .....	42
3.2.4	Storing multi-language extracted text .....	43
3.3	Enabling full-text for unstructured content in a table application .....	45
3.4	Table archive indexing .....	47
3.4.1	The indexing process for table-based applications .....	47
3.4.2	Creating indexes .....	48
3.4.2.1	Single column value indexing vs. basic-text indexing vs. full-text indexing .....	49
3.4.2.2	Multi-column path value index .....	50
3.4.2.3	Enforcing unique column values .....	51
3.4.2.4	Custom indexes .....	51
3.4.2.5	Multi-language custom indexes .....	53

3.5	Backing up and restoring table databases .....	55
3.6	Editing the stores for a database .....	56
3.7	Using the JDBC driver to retrieve table data .....	56
3.7.1	Configuring a SQL client to use the JDBC driver .....	57
3.7.2	Example: configuring SQuirreL SQL client to use the JDBC driver to retrieve table data .....	59
3.7.3	Setting up TLS/SSL for the JDBC driver .....	60
3.7.4	Example: configuring SQuirreL SQL client to use TLS/SSL with the JDBC driver .....	62
<b>4</b>	<b>Configuring a SIP application .....</b>	<b>63</b>
4.1	SIP datamodel .....	63
4.2	Holding .....	64
4.2.1	Creating a holding with the holding wizard .....	65
4.2.1.1	Step 1: Start .....	66
4.2.1.2	Step 2: Data and Contents .....	67
4.2.1.3	Step 3: Ingestion .....	71
4.2.1.4	Step 4: Stores .....	74
4.2.1.5	Step 5: Confirmation .....	76
4.2.1.6	Step 6: Retention .....	77
4.2.1.7	Step 7: Review Configuration .....	79
4.2.1.8	Limitations of the holding wizard .....	79
4.2.2	Editing a holding .....	79
4.2.2.1	Editing a holding's name .....	80
4.2.2.2	Editing a holding's ingestion settings .....	80
4.2.2.3	Editing a holding's store settings .....	81
4.2.2.4	Editing a holding's confirmation settings .....	82
4.2.2.5	Editing a holding's retention settings .....	82
4.2.2.6	Exporting a holding .....	83
4.3	Holding configuration details .....	83
4.3.1	Library mode .....	83
4.3.2	Cache management .....	85
4.3.2.1	Frequently asked questions about caching .....	85
4.3.3	Structured data libraries management .....	86
4.3.3.1	Modes .....	87
4.3.3.2	Load balancing .....	87
4.3.3.3	Locking the library in the cache .....	88
4.3.4	Partition keys .....	88
4.3.4.1	Using partition keys .....	89
4.3.4.2	Types of partition keys .....	90
4.3.5	PDI indexes .....	90
4.3.6	Stores .....	91
4.3.6.1	About offline storage .....	92

4.3.7	Handling unstructured content (CI) .....	93
4.3.8	Using the confirmation .....	94
4.3.8.1	Confirmation .....	94
4.3.8.2	Delivery channel .....	98
4.3.9	Custom SIP format support .....	98
4.3.9.1	Goals .....	98
4.3.9.2	Reception .....	99
4.3.9.3	Ingestion .....	100
4.3.9.4	Content retrieval .....	100
4.4	AIC configuration details .....	101
4.4.1	Configuring the AIC .....	101
4.4.1.1	Using an AIC predicate .....	102
4.4.2	Configuring the query .....	103
4.4.3	Configuring quota for search .....	104
4.4.4	Configuring the ResultConfigurationHelper .....	104
4.5	Enabling full-text on unstructured content in a SIP archive .....	105
4.5.1	Language option .....	107
4.5.2	Additional settings .....	108
<b>5</b>	<b>Ingesting data for a SIP application .....</b>	<b>111</b>
5.1	SIP ZIP format .....	112
5.1.1	SIP SDK .....	113
5.1.2	Connectors .....	113
5.2	Delayed ingestion support .....	114
5.3	Receiving data and ingesting a SIP using the IA Web App .....	114
5.4	Limiting the size of files transferred through REST .....	115
<b>6</b>	<b>Setting up searches for an application .....</b>	<b>117</b>
6.1	Differences based on application archive type .....	118
6.2	Search CRUD .....	118
6.2.1	Search listing .....	119
6.2.2	Configuring one-click display for a search form .....	119
6.2.3	Creating a single application search .....	120
6.2.4	Creating a cross-application search .....	122
6.2.5	Search importing and exporting from/to ZIP .....	126
6.2.6	Copying a search .....	127
6.2.7	Editing a search .....	127
6.2.8	Deleting a search .....	127
6.2.9	Search workflow .....	128
6.3	Configuring search sets .....	128
6.3.1	Creating a search set .....	128
6.3.2	Managing permissions .....	129
6.3.3	Saving a search set .....	129

6.3.4	Editing search set options .....	129
6.3.4.1	Configuring a retention period for background searches .....	130
6.3.4.2	Configuring a search result quota .....	130
6.3.4.3	Configuring a search to run in parallel .....	130
6.3.4.4	Defining target languages at the search level .....	131
6.3.5	Renaming a search set .....	132
6.3.6	Deleting a search set .....	132
6.3.7	Duplicating a search set .....	132
6.4	Composing a search form using the XForms Editor .....	133
6.5	Composing a search form using IA Web App .....	134
6.5.1	Adding fields .....	135
6.5.2	Moving and resizing fields .....	135
6.5.3	Adding default values .....	136
6.5.3.1	Adding a default value to a text field .....	136
6.5.3.2	Adding default values to a number field .....	137
6.5.3.3	Adding default values to a number range field .....	137
6.5.3.4	Adding default date range values to a search form .....	137
6.5.3.5	Adding default number values to a search form .....	138
6.5.4	Searching for a value in multiple fields .....	138
6.5.5	Configuring an input element on a search form .....	139
6.5.5.1	Configuring number ranges for an input element .....	141
6.5.6	Configuring key and value pairs for form elements .....	141
6.5.7	Configuring a checkbox element on a search form .....	142
6.5.8	Configuring a radio group element on a search form .....	143
6.5.9	Configuring the select element on a search form .....	145
6.5.10	Configuring a date element on a search form .....	147
6.5.10.1	Adding date-time fields to a search form and results .....	149
6.5.10.2	Mapping a date range or datetime range control to two different columns .....	151
6.5.10.3	Restricting a date range field .....	152
6.5.11	Configuring a text element on a search form .....	152
6.5.12	Configuring a container element on a search form .....	153
6.5.12.1	Using containers to create search forms that contain conditional fields .....	153
6.5.13	Value lists .....	157
6.5.13.1	Creating a value list .....	157
6.5.13.2	Using value lists .....	158
6.5.13.3	XQueries for value lists .....	159
6.5.13.4	Samples .....	159
6.5.14	Configuring data resolution .....	160
6.5.14.1	Configuring data resolution for supporting search value expansion ...	161
6.5.14.2	Configuring data resolution to use XQuery .....	161

6.5.14.3	Configuring data resolution for form control .....	162
6.5.14.4	Configuring data resolution to use an external service .....	162
6.5.14.5	Configuring data resolution for form control .....	163
6.5.14.6	Unsupported form controls .....	164
6.5.14.7	Samples .....	164
6.5.15	Search form composition tips .....	164
6.6	Composing the search result .....	165
6.6.1	Adding columns .....	165
6.6.2	Moving columns and fields in a set of search results .....	167
6.6.3	Customizing the column type .....	170
6.6.4	Configuring a column of search results .....	170
6.6.5	Disabling local time zone in search results .....	172
6.6.6	Adding filters to a column of search results .....	173
6.6.7	Adding an external link to search results .....	174
6.6.8	Extending a nested search to another application .....	177
6.6.9	Configuring a result column to allow downloadable unstructured content .....	178
6.6.10	Configuring a result column for downloadable content .....	179
6.6.10.1	OpenText Information Archive Viewer .....	180
6.6.10.2	Native Browser .....	180
6.6.11	Configuring a result column when the Column Type is View Content or Viewable Export .....	181
6.6.12	Configuring search result export .....	186
6.6.12.1	Configuring a result column to allow search result export .....	188
6.6.12.2	Configuring the HTML Content View .....	190
6.6.12.3	Including repeating elements in exported data .....	191
6.6.12.4	Setting sort functionality on repeating elements .....	191
6.6.13	Composing result details .....	191
6.6.13.1	Adding details to a side panel .....	192
6.6.13.2	Adding an in-line panel .....	193
6.6.13.3	Configuring autoexpand of in-line panels .....	193
6.6.13.4	Adding tabs .....	193
6.6.13.5	Customizing panel fields .....	194
6.6.13.6	Configuring a custom view for an inline panel .....	200
6.6.13.7	Reordering fields .....	200
6.6.14	Configuring a horizontal scroll bar .....	200
6.6.15	Configuring search results page size .....	201
6.6.16	Custom view .....	201
6.6.17	Enabling a timeline view of results .....	202
6.6.18	Configuring advanced XQuery expressions on SIP search results ....	209
6.7	Creating a custom view using the IA Web App .....	210
6.7.1	Editing a custom view .....	211

6.7.2	Duplicating a custom view .....	211
6.7.3	Deleting a custom view .....	212
6.7.4	Custom views and Content Aviator .....	212
6.8	Advanced search configuration .....	213
6.8.1	Pre- and Post-Filtering .....	213
6.8.2	Nested searches .....	214
6.8.3	Setting up Smart URL .....	214
6.8.4	Entering categories to organize search forms .....	216
6.8.5	Searches and Glacier .....	216
6.8.6	Content retrieval from Glacier .....	217
6.8.7	HTML Content View .....	217
6.8.8	Search result queries .....	218
6.8.9	Form composition details .....	220
6.8.10	OpenText Intelligent Viewing integration .....	244
6.8.10.1	OTIV integration architecture .....	245
6.8.10.2	Example of using OpenText Intelligent Viewing in OpenText Information Archive .....	246
6.9	SIP specific search configuration .....	247
6.9.1	Cross holding .....	247
6.9.1.1	With a plain search .....	247
6.9.1.2	With creating primary and nested searches .....	250
6.9.2	Searching for records that are cached out .....	252
6.10	Table search configuration .....	253
6.10.1	Identifying the search criteria data type .....	253
6.10.2	Creating the SQL query .....	253
6.10.2.1	Creating a multi-language SQL query .....	256
6.10.3	Editing the SQL query for record lookups by ID .....	257
6.11	Setting the debug options for a search .....	258
6.12	Adding user context in custom search forms and search result presentations .....	260
6.12.1	Injecting user context XForm instance to a custom XForm .....	260
6.12.2	Accessing user context in a custom XForm .....	261
6.12.3	Adding user context APIs to IA Custom Presentation Service .....	261
6.12.4	Accessing user context in a custom presentation .....	261
6.13	Search troubleshooting .....	262
<b>7</b>	<b>Configuring the export of search results .....</b>	<b>271</b>
7.1	Introduction .....	271
7.1.1	Creating export configurations at the tenant and application levels ...	274
7.1.2	Default export configurations .....	276
7.2	Writing XSLT to export search results .....	280
7.2.1	Default collection .....	280
7.2.2	Get-Compliance .....	282

7.2.3	PDF support .....	282
7.3	Writing CUSTOM implementation to export search results .....	286
<b>8</b>	<b>Declarative configuration .....</b>	<b>293</b>
8.1	Introduction .....	293
8.2	YAML-based format .....	293
8.3	DC variants .....	294
8.3.1	ZIP archive .....	295
8.4	Import DC configuration .....	296
8.5	Export DC configuration .....	297
8.5.1	Export levels .....	297
8.5.2	Export content and formats .....	297
8.5.3	Security export aspects .....	299
8.5.4	Export API .....	299
8.6	Logging .....	300
8.7	Supporting old 'S3Glacier' declarative configurations .....	300
8.8	Example applications .....	300
8.8.1	Example application declarative configurations .....	301
8.8.2	Locating OpenText Information Archive applications outside of the examples folder structure .....	301
8.8.3	Example application data .....	302
8.8.4	IA Shell installation script .....	302
8.8.5	Learning tools .....	303
8.8.5.1	SIP .....	303
8.8.5.2	Table .....	303
8.8.5.3	Declarative configuration and table metadata .....	303
8.8.5.4	XML namespaces in table DC .....	304
8.8.5.5	Configuring a table search .....	304
8.8.6	Other .....	305
<b>9</b>	<b>Performing compliance tasks .....</b>	<b>307</b>
9.1	Viewing, creating, and deleting a retention policy .....	307
9.1.1	Viewing retention policies .....	307
9.1.2	Common use cases for retention .....	310
9.1.3	Creating a retention policy .....	317
9.1.4	Deleting a retention policy .....	319
9.2	Applying retention .....	319
9.2.1	Choosing where to apply retention and the consequences .....	319
9.2.2	Hardware retention .....	320
9.2.2.1	Retention and the different store types .....	320
9.2.2.2	When are dates pushed to the hardware? .....	322
9.2.2.3	Changing retention policies based on storage .....	324
9.2.3	Applying multiple retention policies .....	325

9.2.4	Table archiving – application .....	326
9.2.5	Table applications .....	327
9.2.6	SIP archiving .....	328
9.2.6.1	Record-based retention .....	329
9.3	Mechanisms for applying retention .....	329
9.3.1	Using the Application Info tab .....	329
9.3.1.1	Applying a retention policy to an application from the Application Info tab .....	330
9.3.2	Applying a hold to search results .....	330
9.3.3	Applying retention to records .....	332
9.3.4	Applying a retention policy to an AIP .....	333
9.3.5	Using jobs to apply retention .....	334
9.3.5.1	Using the Apply Retention Policy to Records Job .....	334
9.3.5.2	Using the Apply Retention Rule to Records Job .....	336
9.3.6	Restricting the apply retention and hold jobs to only evaluate records ingested since the last run for SIP applications .....	337
9.3.7	Using rules to apply retention .....	337
9.3.7.1	Viewing an application's rules .....	338
9.3.7.2	Running rules to apply retention for an application .....	338
9.3.7.3	Using the rule engine to apply a retention policy or hold to records ...	340
9.3.7.4	Getting started with rules .....	342
9.3.7.5	Jobs using rules .....	342
9.3.7.6	Example of granular retention .....	343
9.3.7.7	Rule files .....	345
9.3.7.8	Loading rules .....	346
9.3.7.9	Access to metadata .....	346
9.4	Applying retention to records for SIP .....	348
9.4.1	Applying granular retention using the holding wizard .....	348
9.4.2	Updating configuration to apply retention classes dynamically on records .....	349
9.5	Checking that data is protected .....	350
9.5.1	Viewing items in a retained set .....	350
9.5.2	Verifying that an item is protected .....	351
9.5.2.1	Verifying using the application's Tables tab .....	351
9.5.2.2	Verifying using the application's Packages tab .....	352
9.5.2.3	Verifying retention and holds in search results .....	353
9.6	Event-based retention .....	353
9.6.1	Applying an event-based retention policy .....	353
9.6.2	Fulfilling events with the Trigger Event Policy Job .....	353
9.6.2.1	Populating event dates for the Trigger Event Policy job .....	354
9.6.3	Choosing between event-based and mixed retention policies .....	355
9.6.4	Fulfilling events with rules .....	355

9.6.5	Process Retention Events Job .....	356
9.7	SIP retention .....	356
9.7.1	How does refactoring work? .....	356
9.7.2	Purge confirmations .....	356
9.8	Determining if a retention policy is in use .....	357
9.9	Changing a retention policy .....	357
9.9.1	Removing retention .....	359
9.9.1.1	Removing retention from an application from the Application Info tab	359
9.9.1.2	Running the Remove Policy job .....	359
9.9.2	Re-enabling disposition processing .....	360
9.9.3	Running the Requalification job .....	360
9.10	Disposition flow .....	361
9.10.1	How are items added to a purge candidate list? .....	361
9.10.2	Purge Lists .....	364
9.10.3	Performing actions to a purge list .....	366
9.10.3.1	Exporting purge lists .....	367
9.10.3.2	Exporting packages or tables from a purge list .....	368
9.10.3.3	Providing a reason and attaching approval documentation when approving a purge list .....	369
9.10.4	Application disposition .....	369
9.10.4.1	SIP application disposition .....	370
9.10.5	Running the Clean Up Purge Candidate Lists and Applications job ...	371
9.10.6	AIP disposition .....	372
9.10.6.1	Immediately disposing AIPs .....	372
9.10.6.2	AIU disposition .....	373
9.10.6.3	Disposition of AIPs and AIUs .....	373
9.10.7	Table disposition .....	374
9.10.7.1	Disposing a table with records under hold .....	375
9.10.7.2	Table record disposition .....	376
9.10.7.3	Disposition of tables and table rows .....	376
9.10.7.4	Table disposition and storage footprint .....	376
9.11	Retention, holds and disposition for tables with foreign key constraints and extracted text .....	377
9.12	Records Report .....	378
9.12.1	Running reports to check for overdue and upcoming dispositions .....	380
9.13	Holds .....	380
9.13.1	Creating a hold .....	382
9.13.2	Applying a hold .....	384
9.13.2.1	Applying a hold to an application from the Application Info tab .....	384
9.13.2.2	Applying a hold to an AIP .....	385
9.13.2.3	Applying a hold to a table .....	385
9.13.2.4	Applying a hold to saved search results .....	386

9.13.3	Using legal matters to apply holds .....	386
9.13.3.1	Creating legal matters .....	386
9.13.3.2	Working with legal matters .....	387
9.13.3.3	Sharing legal matters .....	389
9.13.3.4	Access and authorization rules for legal saved searches and matters .....	389
9.13.4	Using rules to apply holds .....	391
9.14	Removing holds .....	393
9.14.1	Viewing items in a hold set .....	393
9.14.2	Removing an item from a hold set .....	395
9.14.3	Removing a hold from an application from the Application Info tab ...	395
9.15	Running and viewing metrics with the compliance dashboard .....	396
9.15.1	Viewing application retention coverage .....	397
9.15.2	Exporting data from the compliance dashboard to CSV format .....	400
9.16	Enriching search results with compliance information .....	402
9.16.1	Example of how to update a sample application to use compliance columns .....	403
9.16.2	Enriching saved searches with compliance .....	403
9.17	Exporting compliance information .....	404
9.17.1	Configuring an export configuration for compliance information .....	404
9.17.2	Exporting retention and hold sets .....	405
9.18	Audits .....	405
9.18.1	Using audits for compliance .....	405
9.18.1.1	Audit event type .....	406
9.18.1.2	Application-specific audits .....	406
9.18.1.3	Searching for audits .....	407
9.18.1.4	Audit entries .....	407
9.18.2	Using the Archive Audit job .....	409
9.18.3	Configuring Apache Kafka Producer Client .....	409
9.18.3.1	Event format .....	409
9.18.3.2	Configuration .....	411
9.18.3.3	Enabling the Kafka profile .....	416
9.19	Caching in packages that contain records for retention sets and hold sets .....	417
9.20	Compliance troubleshooting .....	418
<b>10</b>	<b>Improving performance .....</b>	<b>421</b>
10.1	Specifying the number of batches .....	422
10.2	Improving the response time for a background request .....	422
10.3	Improving performance for a SIP archive .....	423
10.3.1	Improving ingestion speed .....	425
10.3.2	Prevent loading table data twice .....	426
10.3.3	Improving search speed for a SIP archive .....	426

10.4	Lucene libraries configuration cache .....	426
10.4.1	Viewing Lucene cache statistics .....	427
10.5	Improving performance for a table archive .....	428
10.6	Importance of the IO performance of the storage systems .....	428
<b>11</b>	<b>Appendix .....</b>	<b>431</b>
11.1	Supported languages .....	431
11.2	Supported file types for unstructured content text extraction .....	432
11.3	Mapping XSD data types .....	436
11.4	Mapping table SQL data types to PostgreSQL data types .....	439
11.5	AIP .....	440
11.6	Supplemental data per event type and event name .....	445
11.6.1	Partial dispose AIP event type .....	446
11.6.2	Dispose .....	446
11.6.3	AIC .....	446
11.6.4	Job instance .....	447
<b>12</b>	<b>Glossary and acronyms .....</b>	<b>449</b>



# Chapter 1

## Introduction

Configuring OpenText Information Archive consists of taking a newly installed instance and taking it through all of the stages necessary to ingest data from a source applications. At that point, you can perform day-to-day administration and optimization.

The main tasks that you perform are as follows:

---

### Configure system-wide settings

Configure settings that apply to all of your archiving applications, such as storage and auditing, and to OpenText Information Archive as a whole, such as language and branding. For more information, see section 2 “Configuring system-wide settings” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

---

### Configure an application

Perform configuration tasks that apply to both table and SIP archiving applications. While later sections of this guide contain more information about this topic, to review conceptual information, see section 4 “Applications and data” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.

---

### Configure a table application

Perform configuration tasks that are specific to table archiving applications. While later sections of this guide contain more information about this topic, to review conceptual information, see section 5 “Table archiving fundamentals” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.

---

### Configure a SIP application

Perform configuration tasks that are specific to SIP archiving applications. While later sections of this guide contain more information about this topic, to review conceptual information, see section 6 “SIP archiving fundamentals” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.

---

### Set up searches for an application

Set up the searches that users will use to access an application’s data. While later sections of this guide contain more information about this topic, to review conceptual information, see section 4.3 “How data is searched” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.

---

### Configure which search formats can be used for search results

Configure how search results are exported and downloaded by users.

---

### Advanced application setup using a declarative configuration

Using YAML files, perform advanced configuration for an existing application or create an application.

---

#### **Administer OpenText Information Archive**

Perform day-to-day administrative tasks. For more information, see section 3 “Administering OpenText Information Archive” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

---

#### **Perform compliance tasks**

Perform tasks related to retention and disposition.

---

#### **Improve performance**

Optimize your deployment and troubleshoot performance problems.

---

## **1.1 Archiving requirements questionnaire**

Before you configure OpenText Information Archive, for each application that you want to archive information from, consider the following questions:

- What is your archiving use case? Decommissioning a legacy application or live archiving? For more information, see section 1 “What is OpenText Information Archive?” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.
- What kind of OpenText Information Archive application do you want to create? A table or SIP archive? To learn about the differences between these two types of archives, see section 2.3 “Table archiving and SIP archiving” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.
- What type of storage do you want to use to store unstructured content? For example, a local file system for non-production data, Dell EMC Elastic Cloud Storage, Amazon S3 Storage, or CAS?
- What is your desired deployment and performance?
  - How many users are going to access the system?
  - When is data going to be ingested? Per night or per hour?
  - What is your approximate database volume (in TB or PB)?
  - What kind of distributed topology do you want?
- For a SIP-based archive, consider the following:
  - How many SIPs per day are you expecting?
  - How many AIUs per SIP are you expecting?
  - What is the retention date range? How long should data be stored?
  - What is the total amount of AIPs that you are expecting?
  - What is the average volume (in MB) of the SIP package?
  - What is the average volume (in MB) of the `eas_pdi.xml` file inside the SIP package?
  - What is the average volume (in MB) of unstructured content in the SIP package?

- For a table-based archive, consider the following:
    - How many table schemas does the application have?
    - How many records per table?
    - Is there any unstructured content?
    - What is the volume (in MB) of the data file to be ingested?
  - For High Availability (HA)/disaster recovery (DR) of your system deployment, consider the following:
    - What are your expectations for:
      - Recovery Time Objective (RTO): Consider how long it takes OpenText Information Archive or some of its components to recover from a disaster (for example, a server crash),
      - Recovery Point Objective (RPO): Consider the level of loss of recent operations (ingest, retention, etc.) is acceptable after a disaster.
- Such questions are often related to how mission-critical the archive is for an organization.
- How will you manage backups and replication of the various types of data?
    - For RPO, what frequency should backups be stored for various types of content?
    - What are the inherent backup capabilities of your storage systems?

## 1.2 Best practices: working with a staging environment

You should have a staging environment set up as well as a production environment. For more information about setting up a staging environment, see section 5.6 “Setting up a staging environment” in *OpenText Information Archive - Installation Guide (EARCORE-IGD)*.

The following best practices will help you make the most of your staging environment. Throughout this guide, you can find more information about how to perform the tasks mentioned below.

- Your staging environment should have the same configuration as your production environment.
- In your development environment, design your configuration using test data. Then export the configuration from your development environment and import it into your staging environment.
- In your staging environment, test the configuration with data volumes that are similar to what you expect in your production environment.

- If you are using granular retention, run the Refresh Metrics job and look at the administration dashboard to see how much storage you are consuming, before you go into production.
- Run maintenance jobs regularly (for example, the Archive Audits job and Refresh Metrics job).
- Adjust your auditing of event types to make sure that you are auditing at the correct level. Turn off any audit event types that might be excessive.
- Once you have set up your applications in your production environment, set each application (including the Audit and Reports applications) to active to prevent the deletion of each application and its data. If an application is active, its data can only be disposed through the disposition process.
- Upgrade your staging environment before your production environment.

## 1.3 Using the Global Settings tab

The settings listed in the **Global Settings** tab used to be defined in the `application.yml` file, and are now persisted in the database so they can be implicitly shared between all OpenText Information Archive nodes and updated dynamically without necessitating a restart. Some of them have been slightly renamed/restructured but should still be recognizable. The purpose of these properties and how they affect behavior has not changed compared to their previous `application.yml` equivalents.

For more information about the **Global Settings** tab, see section 3.12 “Global Settings” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

There are many system configuration properties and four batch categories, each of which supports many different batch operations. The properties as well as the 'all' fallback for the four batch categories are mandatory and always have values. These properties use a hardcoded default value that is implicitly reverted to when unsetting them. There are also a few selected batch operation properties that use hardcoded default values, which is more appropriate than the 'all' fallback. All other batch operation properties can be optionally set with a value that overrides the 'all' fallback for that operation and can be unset to use the fallback value instead again.

## 1.4 Configuring CORS headers

Cross-Origin Resource Sharing (CORS) is a mechanism, based on HTTP headers, that allow a server to relax security constraints for a client so that it accepts additional HTML content from other hosts. By default, browsers (clients) restrict cross-origin HTTP requests coming from a host other than the origin host (the host that issued the main content). This restriction is due to security requirements to follow the same-origin policy.

However, most application servers allow for an additional configuration of the server, where you could specify a white-list of additional hosts for which HTTP requests should be allowed. This configuration allows you to hone it further by specifying which HTTP verbs/methods should be allowed and/or specific HTTP headers.

OpenText Information Archive allows for a similar configuration. By default, CORS is not enabled. You can enable CORS (headers) and specify which origins (hosts), HTTP methods, and headers should be allowed.

### Example 1-1: Helm-specific configuration for OpenText Information Archive

```
cors:
  enabled: false
  allowedOrigins:
  - "*"
  allowedMethods:
  - "HEAD"
  - "GET"
  - "PUT"
  - "POST"
  - "DELETE"
  allowedHeaders:
  - "*"
  exposedHeaders:
  - "content-disposition"
```

To enable CORS configuration, set `cors.enabled` to `true` and then set `allowedOrigins` (`Access-Control-Allow-Origin`) accordingly. For example, to allow content coming from `http://safe-site.example.com` and `http://also-safe-site.example.com`, set this list as follows:

```
allowedOrigins:
- "http://safe-site.example.com"
- "http://also-safe-site.example.com"
```

The following list outlines additional configuration options:

#### **allowedMethods**

Specifies the list of HTTP methods (one or more) allowed for CORS requests (`Access-Control-Allow-Methods`), when accessing a resource in response to a preflight request.

#### **allowedHeaders**

Specifies the list of allowed HTTP headers. Used in response to a preflight request to indicate which HTTP headers can be used when making the actual request (`Access-Control-Allow-Headers`).

#### **exposedHeaders**

Specifies the list of HTTP headers to indicate which response headers should be made available to scripts running in the browser in response to a cross origin request. This allows a server to indicate which response headers should be made available to scripts running in the browser in response to a cross-origin request.



**Note:** Adding a wildcard (see above) allows for all values. It may be used as a starting point in fine-tuning but, ultimately, you should provide a specific list of allowed values. For more details on these fields, refer to the CORS definition documentation on the internet.



## Chapter 2

# Managing an OpenText Information Archive application

As discussed in section 4 “Applications and data” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*, an OpenText Information Archive application provides access to archived data.

Creating an application requires an Administrator to create storage and, for table applications, to configure a data node.

**The following is an overview of the steps involved in the creation of a SIP application:**

1. Create the shell of an application via the IA Web App.
2. Configure the space for the new application.
3. Create stores for the application based on your storage type. Refer to section 2.2 “Setting up storage for unstructured data” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)* for more information.
4. Use the holding wizard to configure the application’s holdings.
5. Create searches to access the archived data.
6. Ingest data and test the searches. Refer to section 2.7.5 “ingest” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)* for more information.
7. Test the searches that were created.
8. Once satisfied, set the application to active to ensure data can only be deleted by disposition.

**The following is an overview of the steps involved in the creation of a table application:**

1. Create the application using Declarative Configuration (DC), including setting up the databases, schemas, and tables.
2. Create a relational database (RDB) database to store structured content and configure with the space used by the new application.
3. Configure the space for the new application.
4. Create stores for the application based on your storage type.
5. Create searches to access the archived data.

6. Ingest data and build the necessary indices. Refer to section 2.7.5 “ingest” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)* for more information.
7. Test the searches that were created.
8. Once satisfied, **set the application to active** to ensure data can only be deleted by disposition.

## 2.1 Exporting configuration from staging and importing into a production environment

OpenText Information Archive provides the ability to export the configuration for your application, a holding, or a search. If you plan to transfer an application from one environment to another, it is recommended that the user is in Developer, Administrator, and Retention Manager roles to ensure all parts for an application are installed.

Exporting the configuration for an application does not export any of the data, just the configuration. To export an application, from the **Applications** listing tab, select the **Export to Zip file** option and click the **Export** button.

When exporting an application, all `sqlFunctions` are also exported. You may want to clean up any unused `sqlFunctions` before exporting an application.

Export configuration definitions associated to a tenant only will be exported as a reference only unless the user is the Administrator.

### 2.1.1 Preparing the ZIP file for exported SIP application into another deployment

There are no additional steps required, as the exporting ZIP file can be used to import into your new system.

### 2.1.2 Preparing the ZIP file for an exported table application into another deployment

In the table application’s administration-config folder, make the following changes in the `configuration.yml` file:

1. Under the `rdbDataNodes` section:
  - a. Replace the `superUserPassword` with the value that matches the password for the username (used to login to PostgreSQL). If you wish to not have the password in clear text, encrypted, set the value to `${rdbDataNode.password:}` and ensure the encrypted password is set in the `default.properties` file under the `config/iashell` directory.
  - b. If you are using SSL, and the `sslkey` is password protected, add the `sslpassword` property and set it to the correct value.

2. Under the `rdbDatabases` section, replace the `adminPassword` with the value that matches the password for `adminUser`, which is used to login to PostgreSQL.

The following is an example of both set up correctly.

If you plan to use SSL, you need to fill in the `connectionProperties`.

The changes are indicated in **bold** in the following:

```
rdbDataNode:
  name: structuredData
  configure: create
  default: true
  bootstrap: jdbc:postgresql://<Replace with the IP address of structured data node>:5432/
  userName: <Replace with the user name for data node>
  superUserPassword: <Replace with the password for data node>
  connectionProperties:
    ...
    ...

rdbDatabases:
- name: ${application.name}-rdb
  configure: create
  default: true
  adminUser: <Replace with the user name for database>
  adminPassword: <Replace with the password for database>
  rdbDataNode: structuredData
```

## 2.2 User context

Many administration tasks will be performed using the OpenText Information Archive UI, which allows you to navigate, search, select, and edit administration objects, and invoke related administration functionality.

Before we start listing and explaining these functions and objects, here is a note on the UI's user context.

OpenText Information Archive remembers your selections for the secondary navigation toolbar, any filter values you have entered, as well as any column filter values you have entered. For example, you access the **Compliance > Holds** tab, set some filters and navigate to the Applications tab. When you return to the **Holds** tab, the filter values are preserved.

There is no way to deactivate this functionality. You can, however, click the following button to clear any filter information you have entered:



## 2.3 Creating an application with the IA Web App

The **Applications** tab in the IA Web App lists the available applications. From here, you can create or edit applications, or select an application and run a record search. The create and edit operations are available for the Developer role only.

You can quickly find a specific application by using the **Find an Application** filter located in the navigation panel. The filter functionality matches the names and descriptions of each application to input text in the filter field.

When an application is created, it has a status of **In Test**, which allows you to test the application, typically with fake data.

For both SIP and table applications, the Developer needs to create at least one space, one file system, and two stores:

- One space with one filesystem root declared as a storage system,
- One export store (on any storage system)
- One search result store (on a file system only)
- In addition for the table application, a database should be defined

For a SIP application, use the Holding Wizard, which will guide you through the process of configuring the holding.

For a table application, you can create the stores and space, but must use declarative configuration to define the database, schema, and tables for your application.



**Note:** The search result and library stores must be created on a FILESYSTEM storage.

To view the available applications in IA Web App, click the **Applications** tab.

### To create a new application in IA Web App:

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

---

#### Application Name

A name that identifies the application.

---

#### Description

A description of the application.

---

#### Application Category

Select a previously used category or enter the name for a new category.

---

#### Default retention policy

The default retention policy for the application. This field is used only for SIP applications that have a holding configured for package-based retention and the retention policy to be applied is associated with the application.

See [Creating a holding with the holding wizard](#) and [Editing a holding's retention settings](#) for options on how to configure package retention on ingestion.

#### Application Type

An option that specifies a type of archive. It can be one of the following:

- **Application Decommissioning**
- **Active Archiving:** If selected, the **Archive Type** is restricted to **Based on packages** and the **Cache Size** and **Preload** fields are displayed.

#### Archive Type

The available options depend on the selected **Application Type**.

If the application type is **Application Decommissioning**, then one of the following options is available:

- **Based on Table Schema** for a table archive.
- **Based on packages** for a SIP archive.

If the **Archive Type** is based on packages, the **Cache Size** and **Preload** fields are displayed.

Once an application is created, **Archive Type** becomes read only.

#### Cache Size

This field is only available if the **Application Type** field is set to **Active Archiving**.

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 the database.

Select one of the following options:

- **All data is in cache:** All AIPs archived in the application will not be cached out.
- **Limit data in cache. Search may take longer for data not in cache. The CacheOut job must be scheduled to apply the limit:** If selected, enter a cache size and choose whether the limit is in megabytes, gigabytes, or terabytes. If selected, searches may take longer for data that is not cached in.

If this limit is exceeded, the Cache-Out job removes the less-used AIPs from the metadata repository (cache-out), thereby decreasing its size.

#### Preload



**Note:** Preload configuration is applicable only if the Global Setting `search.sip.preloadEnabled` is set to `true` and the YAML server

configuration key `infoarchive.server.preload.enabled` is set to true.

This field is only available if the **Archive Type** field is set to **Based on packages**.

Preloading reduces the overhead of first queries after a start up by opening libraries according to the selected policy.

Select one of the following options:

- Latest created: Lucene cache is populated with all libraries sorted by creation date descending from newest to oldest.
- Latest used: Lucene cache is populated with all libraries depending on the search statistics sorted by the last access date (from the most recently to least recently used).
- Maximum: The maximum number of libraries created or used.
- Period: Only include libraries created or used in the last few days.

Since the Lucene cache size is limited, the priority field is used to set a priority value. The higher the value, the more cache preloading is prioritized for this application.

- 
3. Click **Create**.

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

## 2.4 Setting an application's status to active

Whether an application was created via the IA Web App or DC, the application is automatically set to **In Test**.



### Important

When an application is imported from a staging environment with DC, the imported application will have an In Test status, even if it was previously marked as Active in the staging environment.

The following operations can only be done while an application's status is In Test:

- Use IA Shell to remove individual packages not under retention or a hold.
- **Delete the application's data.**
- **Delete the application.**
- Make changes to the table schemas.

The In Test status is meant to allow you to test the application with sample data and make the necessary changes before real data is ingested. Setting the application status to Active is a permanent change. The status cannot be changed back to In Test. Once an application is active:

- Deleting the application can only be done by putting a retention policy on the application and running disposition.
- It is no longer possible to change the schema for table applications.

**To set an application to active:**

1. For the application being updated, click ... and select **Edit Application**.
2. Set the **Status** to **Active**.
3. Click **Save**.

## 2.5 Editing applications

A user with the role of Developer or Administrator can edit an application.

**To edit an application:**

1. On the **Applications** tab, click the context menu for the application that you want to edit, and select **Edit Application**.
2. Edit the application information. For more information about most of the fields available, see [Creating an application in IA Web App](#). For more information about the remaining fields, see the steps below.
3. The **Primary Language** field allows you to define additional languages to improve searches on unstructured contents with various languages. Defining additional languages has an impact on the library size and the search performance.

For an application, select the primary language from the list. Optionally, select the **Additional Languages** box to add additional languages and then select the desired languages.



**Tip:** Whether you are defining the Primary Language or Additional Languages, use the search field to find the desired language(s).

**Important**

Defining a new language at the application-level has the following impacts:

- For a SIP application, defining additional languages impacts library size and search performance.
  - For a table application, defining additional languages requires more storage space. Furthermore, after changing the primary language or defining additional language settings, the application's unstructured content must be re-extracted and re-indexed.
4. If required, edit the **Status** of the application. A user in the Developer role can only delete all of the data from an application if the application's **Status** is set to

**In Test.** If the application's **Status** is set to **Active**, the following is no longer allowed:

- Deleting the application's data
- Deleting the application

The application and its data can be disposed only through the disposition process. See [Application disposition](#) for more details.



**Note:** Once an application's **Status** is set to **Active**, it can no longer be set back to **In Test**.

5. If required, update the **Cache Size** field. Setting a cache is only available for SIP applications. This field refers to the cache that the libraries are consuming. The main reason for setting this cache is to control the size of the database for the AIPs. Use this mechanism to limit the caching of older historical data that will not likely be searched.

The CacheOut job needs to be scheduled to control the size of the cache. Refer to [Cache management](#) for further information.

6. Click **Save**.

## 2.6 Deleting application data

There are two main reasons a Developer or Business Owner might want to delete application data:

- You are testing an application with sample data, and you want to remove the sample data.
- Your application in a production deployment is no longer needed, so you want to remove the application. The first step is to remove all the data.

If you delete data, searches run against the application no longer work because schemas are deleted upon completion of this procedure. If you have created any rules, they are also deleted.

If the application contains a significant amount of data, the delete data process runs as a background task.

A Developer or Business Owner can only delete all the data from an application if the application has a status of **In Test**. If the application's status is **Active**, data can only be disposed through the disposition process. For more information, see [Application disposition](#). An Administrator cannot delete an application's data.

The Developer and Business Owner can also remove any retention or holds from the data being deleted without relying on the Retention Manager.

**To delete application data:**

1. On the **Applications** tab, click  and select **Delete Data** for the desired application.
2. If retention or holds have been applied to the application data, select the **Delete Compliance data** box.
3. When prompted to confirm that you want to delete all the application's data, click **Delete**.

## 2.7 Deleting an application

A Developer and Business Owner can delete an archived application in **In Test** status (an Administrator can no longer perform this action). The delete operation typically occurs in the development environment, especially during the ETL and data validation processes.

An application can be deleted if there are no running jobs scoped to the application

If the application's status is **Active**, the application can only be disposed through the disposition process. For more information, see [Application disposition](#).

*Before you begin:* You must delete the application's data before you can delete the application. For more information, see [Deleting application data](#).

If an application is in test, it is possible to delete the application's data. If this is done, the DELETE audit event will have an additional entry in the supplemental data where the key is Deleted\_data\_only and the value is the name of the application.

**To delete an application:**

1. On the **Applications** tab, click the context menu for the application and select **Delete Application**.
2. When prompted to confirm that you want to delete the application, click **Delete**.

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

## 2.8 Configuring application storage with spaces

A space is a configuration object that ties part of the configured storage (storage systems, and data nodes and databases) to a particular application.

### 2.8.1 Creating a space in IA Web App

In IA Web App, an Administrator can create a space, which holds records and content for an application.

While a SIP application can have more than one space, a table application can only have one space. If you try to add a second space to a table application, an error message appears.



**Note:** Under normal circumstances, an application should only have one space as it is possible to configure multiple storage types to a single space.

Before you create a space, do the following:

- **Create an application:**
  - Add a storage system (refer to the specific storage system in this guide for further information); at least one file system is mandatory.
  - For table applications only:
    - Register a data node
    - Create a dedicated database to store the application's structured data



**Tip:** Most applications only require one space, as it is possible to associate multiple storage types to a single space.

In IA Web App, on the application's **Spaces** tab, spaces are displayed in a table that contains the following information:

---

#### Space Name

The name of the space.

---

#### Storage System

The storage system associated with the space.

---

#### In Use

Indicates whether the space is currently being referenced.

---

#### Side panel

The details of the selected space, including:

- Description
- The storage system (only if using Cloud storage)

- The file system root path

---

**To create a space in IA Web App:**

1. On the application's **Spaces** tab, click +.
2. Specify the following information:
  - **Space Name:** Enter a name for the space.
  - **Database:** Select the database to be associated with the application (this section appears only for table applications).
  - **Storage Type:** Select a type of storage for the space (for example, file storage). Customers can associate multiple storage types to the space.
  - **Storage Name:** Select the name of the storage being used for the space.
3. Click **Create**.

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

## 2.8.2 Editing a space in IA Web App

If a space is not in use, you can edit it or delete it. When you edit a space, you can update its space name and add or delete a database library or storage system.

If a space is in use, you can only edit it.

**To edit a space in IA Web App:**

1. In IA Web App, click the application's **Spaces** tab.
2. In the **Spaces** table, click the down arrow next to the **Space Name** of the space you want to edit and click **Edit**.
3. Do any of the following:
  - Change any of the settings, such as the name of the space
  - Add a storage system by clicking the add button (+)
  - Remove a storage system by clicking the remove button  This only removes the reference of the application to the storage.
4. Click **Save**.



## Chapter 3

# Configuring a table application

*OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)* explains the conceptual differences between table-based and SIP-based archiving (for more information, see section 5 “Table archiving fundamentals” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)* and section 6 “SIP archiving fundamentals” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*). There are also significant differences in the way these types of archiving are configured. This chapter explains how to configure a table application.

## 3.1 Configuring how data is stored in PostgreSQL

Table application data needs to be stored in PostgreSQL. When configuring a table application, you need to specify the data node and database that is to be used. We recommend that a separate structured data node be used and not use the system data node. Depending on how many table applications you have, you may want to consider multiple structured data nodes.

The RDB data node represents an installation of PostgreSQL, including information on how to connect.

The RDB database represents a PostgreSQL database that is stored in RDB data node.

You will, most likely, define the RBD data node and RBD database through declarative configuration (the IA Web App interface allows you to create the RBD data node and RBD database), which is explained later in this guide.

When defining the RBD data node, you can choose to either provide an RBD data node user that can create databases or you need to manually create the databases yourself in PostgreSQL.

For table applications, we recommend using a separate database for each table application.

Refer to the example applications on how to define the RDB data nodes and RDB databases in YAML files.

## 3.2 Configuring databases, schemas and tables

Before you can ingest data into your table-based archiving application, you need to let OpenText Information Archive know about the structure and types of data, and how the data needs to be processed. Basically, this is data about data (henceforth, the remainder of this section refers to it in the context of table-based archiving as metadata).

Typically, for a database, compile all the metadata in one XML file, which can be uploaded to the application through IA Shell (see section 1.5 “Configuration files” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*). Before uploading this file, you need to at least have created the application with a database object in it. You will, most likely, complete this using declarative configuration (the IA Web App interface allows you to create the application, but not the database object under it), which is explained later in this guide. The following assumes that both the application and database object are there.

The following is a fragment of metadata configuring the TICKET\_ATTACHMENT table from our Tickets example application:

```
<metadata>
  <defaultSchema>TICKETS</defaultSchema>
  <locale>en-US</locale>
  <schemaMetadataList>
    <schemaMetadata>
      <name>TICKETS</name>
      <tableCount>3</tableCount>
      <tableMetadataList>
        <tableMetadata>
          <name>TICKET_ATTACHMENT</name>
          <recordCount>5</recordCount>
          <columnList>
            <column>
              <name>TICKET_NUMBER</name>
              <ordinal>1</ordinal>
              <type>INT</type>
              <typeLength>10</typeLength>
              <indexing>VALUE</indexing>
              <encrypt>false</encrypt>
            </column>
            <column>
              <name>ATTACHMENT</name>
              <ordinal>2</ordinal>
              <type>VARCHAR</type>
              <typeLength>64</typeLength>
              <indexing>VALUE</indexing>
              <encrypt>false</encrypt>
            </column>
          </columnList>
        </tableMetadata>
      </tableMetadataList>
    </schemaMetadata>
  </schemaMetadataList>
</metadata>
```

The `metadata` element has a `schemaMetadataList` element under which are the definitions of all the schemas in the database. Each `schemaMetadata` element has a `tableMetadata` element under which are the definitions of all the tables in its schema.

The `tableCount` and `recordCount` fields indicate how many tables are expected in the schema, and how many records are expected to be in the table, respectively. It is also possible to set these values to 0 if the actual number is not known. Setting these

fields can be used for so-called chain of custody tests to ensure that the expected number of tables for a schema have been created, and the expected number of records for tables have been ingested. See the following for further information about chain-of-custody:

- The command descriptions of the `chain-of-custody` and `view-chain-of-custody` commands (see section 2.8.2 “chain-of-custody” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)* and section 2.8.12 “view-chain-of-custody” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*),
- Section 2.8.2.1 “chain-of-custody test file format” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*, and
- Section 2.8.2.2 “Running chain-of-custody tests for table archives” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*.

It is possible to divide table definitions for a schema over multiple files, each with a separate set of table definitions. It should be noted that in that case, the `tableCount` field for the schema should be set in only one of these files. It represents the number of table definitions across all files for that schema (so not only for the tables defined in that particular file).

The `defaultSchema` element is needed by the JDBC driver. The `locale` element defines a collation for the columns of `VARCHAR`. The element can be defined for the entire metadata XML file, a schema, or a table element.



**Note:** The collation for the `locale` must be defined in PostgreSQL server when the metadata is uploaded to OpenText Information Archive, otherwise the upload will fail. There is a wide range of collations for various locales defined by PostgreSQL automatically during installation. If the `locale` element is not defined, the default location of PostgreSQL server is used.

The column definitions under a table should look familiar to anyone dealing with relational/SQL databases, except perhaps for:

- `index-related` elements, which are explained below; and
- `encrypt` (boolean). This element indicates whether or not the ingested values in the corresponding column should be encrypted upon ingestion, to facilitate encryption at rest.

### 3.2.1 Table and column constraints

SQL allows you to specify constraints on tables and columns, enabling the relational database management system (RDBMS) (PostgreSQL) to enforce those constraints on your data. OpenText Information Archive table archiving supports (the most important aspects of) the following constraints:

- Foreign key constraints
- Unique key constraints
- Primary key constraints
- Non-null (required) columns

These have an XML representation in our metadata XML schema, enabling you to specify constraints on tables and columns, which, on ingestion/processing, OpenText Information Archive translates into SQL statements for the creation of those constraints in the corresponding PostgreSQL database.

Examples of constraints can be found in the OrderManagement sample application from the OpenText Information Archive distribution. If you navigate to tables (for example, ORDERS, LINEITEMS) of the OrderManagement application in the IA Web App, you can view their constraints in the details panel.

#### Foreign key constraints

A foreign key constraint has the following properties:

**name**

Name of the constraint.

**referenceTable**

Name of the table whose records are referred by (primary) key.

**referenceSchema**

Name of the schema of the table in which records are referred by (primary) key.

**onDeleteAction**

(SET NULL, SET DEFAULT, RESTRICT, NO ACTION, CASCADE)

In our `metadata.xml`, we specify the top four fields directly under the element for the table, and the mapping under the element(s) of the column(s). The following XML fragment from the OrderManagement schema XML illustrates this:

```
<tableMetadata>
  <name>LINEITEMS</name>
  <recordCount>60175</recordCount>
  <foreignKeyConstraints>
    <foreignKey>
      <name>Order_ForeignKey</name>
      <referenceTable>ORDERS</referenceTable>
      <onDeleteAction>CASCADE</onDeleteAction>
    </foreignKey>
  </foreignKeyConstraints>
  <columnList>
    <column>
```

```

<name>L_ORDERKEY</name>
<required>true</required>
<ordinal>1</ordinal>
<type>INTEGER</type>
<typeLength>10</typeLength>
<indexing>VALUE</indexing>
<foreignKey>O_ORDERKEY</foreignKey>
<foreignKeyConstraintName>Order_ForeignKey</foreignKeyConstraintName>
</column>
...

```

In an order management application and database, an order consists, among others, of a number of line items. This is modeled in our schema, by having LINEITEMS records refer to ORDERS records by foreign key. For this, we need to add a `foreignKeyConstraints` element to LINEITEMS tableMetadata with one `foreignKey` in it. We named it Order\_ForeignKey, and through the `referenceTable` element, we make it refer to the ORDERS table in the same schema (we do not specify the `referenceSchema` element, so we default to the current schema). Finally, we specify CASCADE as the `onDeleteAction` to indicate cascading deletes.

```

<foreignKeyConstraints>
  <foreignKey>
    <name>Order_ForeignKey</name>
    <referenceTable>ORDERS</referenceTable>
    <onDeleteAction>CASCADE</onDeleteAction>
  </foreignKey>
</foreignKeyConstraints>

```

All we need to do now is indicate which columns from the LINEITEMS refer to which columns of ORDERS. In our example, the `L_ORDERKEY` column of the LINEITEMS table contains the references to ORDERS table's `O_ORDERKEY` column value (`O_ORDERKEY` is the primary key of the ORDERS table).

To this end we specify that column `L_ORDERKEY` is part of foreign key constraint `Order_ForeignKey`:

```
<foreignKeyConstraintName>Order_ForeignKey</foreignKeyConstraintName>
```

And that the name of the referred column is `O_ORDERKEY`:

```
<foreignKey>O_ORDERKEY</foreignKey>
```

On install of the OrderManagement application, when OpenText Information Archive sees all this, it will make the appropriate SQL calls to PostgreSQL to add this constraint to the LINEITEMS table.

## Unique key constraints

A unique key constraint in PostgreSQL also has a name, and then just a list of columns that constitute the unique key. The following XML fragment from the OrderManagement schema XML illustrates this:

```

<tableMetadata>
  <name>CUSTOMERS</name>
  <recordCount>1500</recordCount>
  <uniqueKeyConstraints>
    <uniqueKey>
      <name>Customer_UniqueKey</name>
    </uniqueKey>
  </uniqueKeyConstraints>

```

```
<columnList>
  <column>
    <name>C_CUSTKEY</name>
    <required>true</required>
    <ordinal>1</ordinal>
    <type>INTEGER</type>
    <typeLength>10</typeLength>
    <indexing>VALUE</indexing>
    <primaryKey>true</primaryKey>
  </column>
  <column>
    <name>C_NAME</name>
    <required>true</required>
    <ordinal>2</ordinal>
    <type>VARCHAR</type>
    <typeLength>50</typeLength>
    <indexing>VALUE</indexing>
    <uniqueKeyConstraintName>Customer_UniqueKey</uniqueKeyConstraintName>
  </column>
  <column>
    <name>C_ADDRESS</name>
    <ordinal>3</ordinal>
    <type>VARCHAR</type>
    <typeLength>100</typeLength>
    <indexing>VALUE</indexing>
    <uniqueKeyConstraintName>Customer_UniqueKey</uniqueKeyConstraintName>
  </column>
...

```

The unique key constraint itself is specified directly under the `tableMetadata` element, listing its name as the only property:

```
<uniqueKeyConstraints>
  <uniqueKey>
    <name>Customer_UniqueKey</name>
  </uniqueKey>
</uniqueKeyConstraints>
```

Each column that is part of the unique key then gets a reference to it:

```
<uniqueKeyConstraintName>Customer_UniqueKey</uniqueKeyConstraintName>
```

In our example, a CUSTOMER's name (`C_NAME`) and address (`C_ADDRESS`) together constitute a unique key for the customer.

## Primary key constraints

In the unique key example, we already encountered the element:

```
<primaryKey>true</primaryKey>
```

in the `C_CUSTKEY` column specification of the CUSTOMERS table. Specifying true indicates that the column belongs to the table's primary key.

## Non-null columns

A non-null column indicates that, for every record, the column must not be null (must have a value).

In both the foreign key and unique key examples, we already encountered the:

```
<required>true</required>
```

element (for example in the L\_ORDERKEY column specification of the LINEITEMS table). Specifying true indicates that we have a non-null column.

### 3.2.2 Unstructured content

Ingesting unstructured content into a table archive along with structured data requires no specific configuration in the application's metadata file. All you need is a varchar column (VARCHAR). If in the XML files with structured table data (records) the corresponding element has a ref attribute, then the system interprets that as a reference to a blob object (unstructured content), which should be part of the ingest request. OpenText Information Archive will process the unstructured data by storing it in the appropriate store and replacing the configured column value of the corresponding row to the content id (an ID of the content object/blob by which OpenText Information Archive can find it back for viewing, downloading, exporting, etc.).

The OrderManagement application includes an example of this. Consider the following configuration for the P\_FACTSHEET column of the PARTS table:

```
<column>
<name>P_FACTSHEET</name>
<ordinal>10</ordinal>
<type>VARCHAR</type>
<typeLength>64</typeLength>
<indexing>VALUE</indexing>
<encrypt>true</encrypt>
...
<ROW>
<P_PARTKEY>20</P_PARTKEY>
<P_NAME>bisque salmon dark blanched linen</P_NAME>
<P_MFGR>Manufacturer#1</P_MFGR>
<P_BRAND>Brand#12</P_BRAND>
<P_TYPE>LARGE POLISHED NICKEL</P_TYPE>
<P_SIZE>48</P_SIZE>
<P_CONTAINER>MED BAG</P_CONTAINER>
<P_RETAILPRICE>920.02</P_RETAILPRICE>
<P_COMMENT>slyly ironi</P_COMMENT>
<P_FACTSHEET ref=".../blobs/20.docx" />
</ROW>
```

Notice that it is just another VARCHAR column. However, if we ingest data like this into the PARTS table, the system processes the referred blob, and in the P\_FACTSHEET column of the PARTS table in PostgreSQL, you will see a value like 55f252fc-8811-45f7-a1ca-807c997a2f24:c1:1 in the corresponding row, which is the ID of the 20.docx blob that was ingested along with the row/record.

For columns representing unstructured content references, the <encrypt> element value is used to determine whether or not the associated content will be encrypted before being stored in unstructured content storage.

### 3.2.3 Storing extracted text

As mentioned in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*, OpenText Information Archive supports (configuration for) the extraction, storage, and indexing of text from unstructured content as part of the ingestion process.

The OrderManagement application includes an example of how to do this. Consider this continuation of the P\_FACTSHEET column of the PARTS table:

```
<column>
  <name>P_FACTSHEET</name>
  <ordinal>10</ordinal>
  <type>VARCHAR</type>
  <typeLength>64</typeLength>
  <indexing>VALUE</indexing>
  <refText>
    <table>FACTSHEETS</table>
    <textColumn>FACTSHEET_TEXT</textColumn>
    <textChunkColumn>FACTSHEET_TEXT_CHUNK</textChunkColumn>
    <overlap>100</overlap>
    <cidColumn>CID</cidColumn>
    <metadataColumn>METADATA</metadataColumn>
  </refText>
</column>
```

With this configuration, OpenText Information Archive extracts the text from any referred blob in the P\_FACTSHEET column data (for example, 20.docx from the previous section).

The system also stores the resulting text in the FACTSHEETS table, in the FACTSHEET\_TEXT column, with its content ID copied in the CID column, and its identified language copied in the LANGUAGE column, and its metadata (`lastModified`, `createdBY`, *etc.*) in the METADATA column.

Unfortunately, PostgreSQL has some limitations on the size of text fields if you want to be able to search for phrases, and create indexes to support phrase search (see below). You can find those limitations in the PostgreSQL documentation on `tsVector`, which is what PostgreSQL uses to store all the lexemes in a text and their positions.

To overcome these limitations, OpenText Information Archive lets you configure a `textChunkColumn` that breaks up text beyond 12,000 words into multiple rows. The chunks will be stored in the column configured in the `textChunkColumn` element (FACTSHEET\_TEXT\_CHUNK in our example). In addition, you can specify how many words of each consecutive pair of chunks should overlap. This is to facilitate searches for phrases that would cross the line between two chunks.

You can set the names of the columns of the `refText` element, and of the target table itself, to anything you want, provided that it is a valid name in PostgreSQL. The following restrictions apply to the `refText` element:

- All column names in a `refText` element must be different.
- The data types of the `textColumn`, `textChunkColumn`, and `cidColumn` must be `varchar`.

- The data type of the `metadataColumn` must be `jsonb`.
- The targetable and all the above columns must be configured in the database's metadata configuration too (see below).
- The value for overlap cannot exceed 1000.
- You must configure a `textColumn` or a `textChunkColumn` element (both is also allowed).
- You must configure a `cidColumn` element.

Storing the content ID in the target table provides a relation between the referring (`PARTS`) and referred (`FACTSHEETS`) table. This can be used in searches to establish a join when searching with predicates in both the extracted text and the other columns of the `PARTS` table.

To complete the configuration for storing extracted text from unstructured content, we must provide an actual configuration of the `FACTSHEETS` table:

```
<tableMetadata>
  <name>FACTSHEETS</name>
  <columnList>
    <column>
      <name>UID</name>
      <ordinal>1</ordinal>
      <type>SERIAL</type>
      <typeLength>64</typeLength>
      <indexing>VALUE</indexing>
    </column>
    <column>
      <name>CID</name>
      <ordinal>2</ordinal>
      <type>VARCHAR</type>
      <typeLength>64</typeLength>
      <indexing>VALUE</indexing>
    </column>
    <column>
      <name>FACTSHEET_TEXT</name>
      <ordinal>3</ordinal>
      <type>VARCHAR</type>
    </column>
    <column>
      <name>FACTSHEET_TEXT_CHUNK</name>
      <ordinal>4</ordinal>
      <type>VARCHAR</type>
    </column>
    <column>
      <name>METADATA</name>
      <ordinal>54</ordinal>
      <type>JSONB</type>
    </column>
  </columnList>
```

A `UID` column was added to the `FACTSHEETS` table configuration. This ensures that each chunk of extracted text gets a unique key, which is convenient for full-text SQL queries into the text chunks.

If you configure text extraction on one or more tables in your application, the actual text extraction is done after ingestion, as part of the Table Indexing job (see [The indexing process for table-based applications](#) for more information).



**Note:** If the unstructured content is encrypted, the Table Indexing job will still be able to extract text from it. However, be aware that the extracted text will not be (re-)encrypted in PostgreSQL. In other words, extracted text from unstructured content always lives as plain text in PostgreSQL.

### 3.2.3.1 Storing extracted text metadata

Some metadata fields are collected during text extraction of the unstructured content. This metadata is stored as JSONB in the column defined by the configuration. In the sample application, the location where the metadata is stored is configured in the `refText/metadataColumn` element of the `P_FACTSHEET` column. As a result, the metadata value is stored in column `METADATA` of the `FACTSHEETS` table.

The following is an example of the `METADATA` column value:

```
{
  "TITLE": "sky burnished salmon navajo hot",
  "AUTHOR": "Jack Hall",
  "CREATED": "2024-02-22T17:22:00Z",
  "MODIFIED": "2024-02-26T10:57:00Z",
  "KV_FORMAT_NAME": "MS_Word_2000_Fmt"
}
```

Not all file types have metadata support. For instance, text files have no metadata support. For file types with metadata support, fields `TITLE`, `AUTHOR`, `CREATED`, and `MODIFIED` are stored. The format field `KV_FORMAT_NAME` contains the file type as detected by the OpenText KeyView text extraction tool. It will be added as metadata field for every file type. For more information about supported file types and metadata, see [Supported file types for unstructured content text extraction](#).

A container file can have multiple subfiles with metadata. The metadata of a container file contains the metadata of these subfiles and reflects its structure. A field with name `ATTACHMENTS` was introduced to store the name and metadata of the subfiles. As a result, it is possible to query the metadata of subfiles.

**Example 3-1: Metadata of a PowerPoint file with embedded Office documents:**

```
{
  "TITLE": "Presentation with embedded Office files",
  "AUTHOR": "Jeff Blank",
  "CREATED": "2022-07-08T21:37:21Z",
  "MODIFIED": "2023-05-09T17:00:49Z",
  "ATTACHMENTS": {
    "Microsoft_Word_Document.docx": {
      "TITLE": "2022 calendar",
      "CREATED": "2022-07-08T21:05:00Z",
      "MODIFIED": "2022-07-08T21:26:00Z",
      "KV_FORMAT_NAME": "MS_Word_2007_Fmt"
    },
    "Microsoft_Excel_Worksheet.xlsx": {
      "TITLE": "Simple meal planner1",
      "CREATED": "2020-08-04T04:11:57Z",
      "MODIFIED": "2022-07-29T14:58:43Z",
      "KV_FORMAT_NAME": "MS_Excel_2007_Fmt"
    }
  }
}
```

```

    "KV_FORMAT_NAME": "MS_PPT_2007_Fmt"
}

```



The sample contains a search with name FactSheet Terms and Metadata to search information based on metadata field values. The search has form fields for full-text search on TITLE and AUTHOR and date range search on CREATED and MODIFIED.

As JSON values are always text, the values of CREATED and MODIFIED must be converted to date values before they can be used in SQL query comparisons. This conversion requires an SQL function, which is defined in the Order\_Management search configuration.yml:

```

sqlFunction:
  schema: ORDER_MANAGEMENT
  sql: |
    CREATE FUNCTION ORDER_MANAGEMENT.jsonfieldtodate (value jsonb, name text) RETURNS
    date as $$ SELECT (value ->> name)::date; $$ language sql immutable;

```

Condition in the search query where clause, using the function for field CREATED:

```

ORDER_MANAGEMENT.jsonfieldtodate(FACTSHEETS.METADATA, 'CREATED') > TO_DATE( $created/
from, 'YYYY-MM-DD' )

```

For optimization by an index, the same function must be used in the search and the index. For more information about construction of the index, see [Custom Indexes](#).

### 3.2.4 Storing multi-language extracted text

OpenText Information Archive supports storage of the extracted content by language separately. This is useful when it is required that searches on extracted text must be executed by language. The extracted text is stored and indexed only once, thereby ensuring searches remain fast.

Normally, the extracted text is stored in a column with a name (for example, FACTSHEET\_TEXT). With multi-language support, however, the text is stored in a separate column with a language code extension (for example, FACTSHEET\_TEXT\_en, FACTSHEET\_TEXT\_fr, FACTSHEET\_TEXT\_es).

To enable this feature, the table data must follow these rules:

- For each reference to unstructured content in the table data, the language of the content must be defined in the same table row.
- The language in the table data must be specified by using ISO 639 standard code.

The OrderManagementMultilanguage application includes an example of how to do this. This application is similar to the OrderManagement application, but it is adjusted to enable multi-language storage. In the OrderManagementMultilanguage application, the references to unstructured content are located in the ATTACHMENTS table where the language of each referenced content can be found:

```

<ORDER_MANAGEMENT>
  <ATTACHMENTS>

```

```
<ROW>
  <A_PARTKEY>100</A_PARTKEY>
  <A_LANGUAGE>en</A_LANGUAGE>
  <A_FACTSHEET ref=".../blobs/en/100.txt"/>
</ROW>
<ROW>
  <A_PARTKEY>100</A_PARTKEY>
  <A_LANGUAGE>fr</A_LANGUAGE>
  <A_FACTSHEET ref=".../blobs/fr/100.txt"/>
</ROW>
<ROW>
  <A_PARTKEY>100</A_PARTKEY>
  <A_LANGUAGE>de</A_LANGUAGE>
  <A_FACTSHEET ref=".../blobs/de/100.txt"/>
</ROW>
```

The `<A_PARTKEY>` represents the foreign key to the PARTS table.

As the ATTACHMENTS.A\_FACTSHEET column now contains the reference, the `<refText>` definition must be on this column in the table metadata:

```
<column>
  <name>A_FACTSHEET</name>
  <ordinal>3</ordinal>
  <type>VARCHAR</type>
  <indexing>VALUE</indexing>
  <refText>
    <table>FACTSHEETS</table>
    <textColumn>FACTSHEET_TEXT</textColumn>
    <textChunkColumn>FACTSHEET_TEXT_CHUNK</textChunkColumn>
    <overlap>100</overlap>
    <idColumn>CID</idColumn>
    <languageColumn>LANGUAGE</languageColumn>
    <languageSource>A_LANGUAGE</languageSource>
    <metadataColumn>METADATA</metadataColumn>
  </refText>
</column>
```

To support multi-language storage, the `<refText>` element must contain the element `<languageSource>`. This element contains the name of the column where the language is found. During text extraction, OpenText Information Archive uses this column name to find the language of the extracted text.

If `<languageSource>` is set, the extracted text is stored in separate language columns.

Only the content of columns with names defined by `<refText>` elements `<textColumn>`, `<textChunkColumn>`, and `<metadataColumn>` can depend on language. In case of multi-language storage, for these columns only, separate columns per language are created.

You can specify a primary language and additional languages in the IA Web App by editing an application (see [Editing applications](#) for more information). After changing an application's language settings, re-extraction and re-indexing is required.

If the language of referenced content is not found in the table data, the primary language is used as fall-back language.

By selecting additional languages, specify which languages the text must be extracted for and indexed next to the primary language.

[Multi-language custom indexes](#) contains more information on how to define indexes when the extracted text is stored in language specific columns. [Creating a multi-language SQL query](#) contains information on how to define multi-language SQL queries.

### 3.3 Enabling full-text for unstructured content in a table application

You can configure an application to extract and index the text of unstructured content during ingestion, search on them with full-text capabilities, and highlight them in the search results. For a list of supported languages, see [Supported languages](#).

You can specify a primary language and additional languages in the IA Web App by editing an application (see [Editing applications](#) for more information).

Select additional languages from the list of supported languages. For table archiving, however, support for a language is only possible when the PostgreSQL application database also has a matching search configuration. Therefore, the selection of languages without a matching search configuration is disabled.

Customers can add a missing search configuration to PostgreSQL manually. The name of a search configuration must match the language name.

#### To enable this feature for a table application:

1. In the **Global Settings** tab, set the `ingestion.ci.text.enabled` to true. For more information, see Section 3.12 “Global Settings” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

This step enables the full-text indexing for both SIP and table applications. To learn more about how the feature interacts with a SIP application, refer to [Enabling full-text on unstructured content in a SIP archive](#).

2. Prior to ingestion, update the metadata XML file to configure that the content in the attachment will be indexed. This way, when the user runs a search, the content will be associated with the correct record:

```
<column>
  <name>ATTACHMENT</name>
  <ordinal>2</ordinal>
  <type>VARCHAR</type>
  <typeLength>64</typeLength>
  <indexing>VALUE</indexing>
  <refContent>
    <ciTextTable>ATTACHMENTS</ciTextTable>
    <ciTextColumn>CI_TEXT</ciTextColumn>
    <iaRowIdsColumn>ia_rowIds</iaRowIdsColumn>
    <cidColumn>CID</cidColumn>
    ...
  </refContent>
</column>
```

For an example of a multi-language application, see [Storing multi-language extracted text](#). For an example of a single language application, see [Storing extracted text](#).



**Note:** In the current implementation it is not possible to display snippets into a dedicated column.

3. For each application, specify `defaultLanguage` and `additionalLanguages` settings. By default, `defaultLanguage` is set to English. These language settings are only used for a multi-language application.
4. Create the search SQL queries.

For table archiving, a full-text search is executed on PostgreSQL. Therefore, the Search SQL queries and indexes must use full-text expressions supported by PostgreSQL.

These expressions usually require a language-specific text search configuration, which specifies the proper dictionary, spelling (stop words), stemming, thesaurus, and synonyms for the language. The name of the search configuration matches the language name.

OpenText Information Archive cannot control the text search configurations delivered with a PostgreSQL distribution. You can check the available text search configurations on your PostgreSQL installation by using the following query:

```
SELECT cfgname FROM pg_ts_config;
```

For more information on multi-language SQL queries, see [Creating a multi-language SQL query](#).

5. Create the full text indexes to optimize the SQL queries.

For examples of single language and multi-language indexes, see sections [Custom Indexes](#) and [3.3.2.5 Multi-language custom indexes](#).

Support for container files is optional. By default, this type of file is not supported.

The OrderManagement application supports container files, and the data contains examples of container files:

- A ZIP file,
- An email message, and
- A PowerPoint document with embedded documents.

For more information on these types of files, see [Supported file types for unstructured content text extraction](#).

When the container files option must be enabled for a database, make sure the database IA Shell commands `index-build`, `index-rebuild`, and `index-extract-text` are executed with the additional `contained-file` boolean option.



### Example 3-2:

```
index-build --d applications/Order_Management/databases/Order_Management-sql-db --  
contained-files true
```



For more information about these IA Shell commands, see Section 2.8 “Commands specific to table applications” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*.

## 3.4 Table archive indexing

Typically, with table-based archiving, there is a fixed set of data (tables) to be ingested “in one go”. Once the table data has been ingested, the user tells OpenText Information Archive to start indexing the data. Contrary to SIP-based archiving, where ingested packages are indexed separately as part of the ingestion process, with table-based archiving, the indexing process indexes all ingested data under a database. To review conceptual information about indexing, see Section 5.1 “Metadata file” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.

If you plan to build indexes for a table application, ensure that there is data for the tables. For example, if you look at the sample application for Baseball, the following appears in the `install.iashell` file:

```
connect
import config
ingest --d applications/Baseball/databases/Baseball-sql-db/schemas/BASEBALL --from data/
BASEBALL
ingest --d applications/Baseball/databases/Baseball-sql-db/schemas/TICKETS --from data/
TICKETS
chain-of-custody --d applications/Baseball/databases/Baseball-sql-db/schemas/BASEBALL
chain-of-custody --d applications/Baseball/databases/Baseball-sql-db/schemas/TICKETS
index-build --d applications/Baseball/databases/Baseball-sql-db
quit
```

As you can see, the `index-build` command only needs to be entered once for the database, and not for each schema, and after ingestion is completed (for more information, see Section 2.8.4 “index-build” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*).

If the `index-build` command runs prior to ingestion, the Table Indexing job stops and a warning is issued. If the database specifies more than one schema, data must be available for each schema. You can ingest additional data after indexing, but indexing will be done upon ingestion and, consequently, impact performance.

### 3.4.1 The indexing process for table-based 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. For more information about the commands listed in the following procedure, see Section 2 “IA Shell commands” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*:

1. In the `<IA_ROOT>/config/iashell/application.yml` file, set the `defaultSettings.table.enableIndexBuildInBatch` parameter to `false`.
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 applications/<application name>/databases/<database name>
```

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. The table index expects that data has already been ingested.

The actual indexing on the PostgreSQL server is triggered by a job and runs in phases. First it creates the indexes on individual columns, and all the `pathValue` indexes, and any custom indexes that do not have the `unstructuredContent` element configured. - After that, it starts to extract text from unstructured content where this is configured and stores this text in the configured target tables and columns. The last step is the creation of indexes that do have the `unstructuredContent` element configured.

### 3.4.2 Creating indexes

To improve search performance, the Developer can use indexes for specific search queries in table archiving and specify the columns in the configuration (`metadata.xml`) for selected fields during or after ingestion. Then, after the ingestion of data, the Developer:

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

The following subsections discuss the various types of indexes you can configure.

### 3.4.2.1 Single column value indexing vs. basic-text indexing vs. full-text indexing

Value index is a B-tree-based index that supports generic value comparison predicates like ‘equal’, ‘greater’, ‘less’, etc. This index improves performance when you have a query that compares a column’s content to a specific value.

A basic text index is a GIN (Generalized Inverted) based index that is designed for string and substring queries. This index is a good choice when you have a query with substring similarity or a ‘regular expression’ pattern lookup.

A full text index is also a GIN index, but built on tsvector lexemes derived from a column’s content. This index is designed for a natural-language full text search where a query requires word distances information, lexemes weight and scoring.

You can configure applications to use text indexing only, 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>
```

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

<b>Only index basic-text search</b>	<indexing>BASIC_TEXT</indexing>
<b>Only index full-text search</b>	<indexing>FULL_TEXT</indexing>
<b>Only index value search</b>	<indexing>VALUE</indexing>
<b>Index both full-text and value search</b>	<indexing>FULL_TEXT_AND_VALUE</indexing>
<b>Index both basic-text and value search</b>	<indexing>BASIC_TEXT_AND_VALUE</indexing>

Basic-text and full-text indexing require much more storage space than value indexing. Only use these options when absolutely required.

By default, the following example applications use basic-text and 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 a basic-text search.

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

### 3.4.2.2 Multi-column path value index

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 will create a multi-column index on three columns: ACTOR\_ID, FIRST\_NAME, and LAST\_NAME

Due to limitations of PostgreSQL, the element <fulltext> is no longer supported. Full-text and basic-text indexes can only be created on a single column.

### 3.4.2.3 Enforcing unique column values

Path value indexes are used to speed up a search, but they can also be used to enforce unique table column values. For instance, if a table has a primary key, then a path value index can be used to ensure that the values of the primary key column remain unique.

An example of a path value index definition to enforce uniqueness of column values of TICKET field TICKET\_NUMBER is:

```
<pathValueIndexList>
    <pathValueIndex>
        <name>TICKET_NUMBER unique key check</name>
        <uniqueKey>true</uniqueKey>
        <column>TICKET_NUMBER</column>
    </pathValueIndex>
</pathValueIndexList>
```

After the sample index is created, an exception will be thrown when a record is added with a non-unique value for TICKET\_NUMBER. If the data is already loaded and the index is added later, however, an exception is thrown while creating the index.

Creating the indexes first avoids this problem and only crash when ingesting data with duplicate values at the expense of ingestion speed. In either case, this is a problem with the data. Either the data must be fixed or the uniqueness of the index must be dropped. Once invalid data has already been ingested, it is not possible to add such a unique key index afterwards.

### 3.4.2.4 Custom indexes

In addition to the above ways of specifying the most common indexes, you can configure just about any index you need in PostgreSQL using the `<indexList>` element that lives after the `<columnList>` element under `<tableMetadata>`. This feature was introduced to support gin indexes with expressions, which in turn is more or less a requirement to make full-text SQL queries performant.

Hence, in our example applications, we include an `indexList` configuration for supporting a full-text query on the text extracted from unstructured content. This is a continuation of the example in [Storing extracted text](#). Here's the snippet from the metadata file of OrderManagement (with the `tableMetadata` element repeated for reference):

```
<tableMetadata>
    <name>FACTSHEETS</name>
    <columnList>
        <column>
            <name>UID</name>
            <ordinal>1</ordinal>
            <type>SERIAL</type>
            <typeLength>64</typeLength>
            <indexing>VALUE</indexing>
        </column>
        <column>
            <name>CID</name>
            <ordinal>2</ordinal>
            <type>VARCHAR</type>
        </column>
    </columnList>
</tableMetadata>
```

```
<typeLength>64</typeLength>
<indexing>VALUE</indexing>
</column>
<column>
  <name>FACTSHEET_TEXT</name>
  <ordinal>3</ordinal>
  <type>VARCHAR</type>
</column>
<column>
  <name>FACTSHEET_TEXT_CHUNK</name>
  <ordinal>4</ordinal>
  <type>VARCHAR</type>
</column>
<column>
  <name>METADATA</name>
  <ordinal>5</ordinal>
  <type>JSONB</type>
</column>
</columnList>
<indexList>
  <index>
    <name>ginindex</name>
    <unstructuredContent>true</unstructuredContent>
    <method>gin</method>
    <fields>
      <field>
        <column>CID</column>
      </field>
      <field>
        <toTSVector>
          <config>english</config>
          <column>FACTSHEET_TEXT</column>
        </toTSVector>
      </field>
    </fields>
  </index>
</indexList>
</tableMetadata>
```

The `indexList` configures one index, which is a gin index for two fields of the `FACTSHEETS` table: `CID` and `FACTSHEET_TEXT` (you can add a similar gin index for the `FACTSHEET_TEXT_CHUNK` column). On the latter, the index uses an expression. It actually indexes the tsvector of the extracted text, keeping track not only of the lexemes, but also their positions. For more on indexes, refer to the PostgreSQL documentation. The XML-based index configuration schema is just a layer on top of the SQL way of index creation. For the complete schema of `indexList` and all other elements of the metadata file, see `sql-metadata.xsd` in the doc part of the OpenText Information Archive distribution.

Below are some notable entries from the schema:

- **unstructuredContent:** This element tells OpenText Information Archive that the index is on extracted text from unstructured content. This is important for determining the moment that the index should be created (i.e., after extracting and storing text from unstructured content).
- **method:** PostgreSQL supports many methods of indexing (gin, gist, btree, etc.). For full-text support, gin is advised.
- **storageParameters:** Each method has its own set of supported storage settings. See the PostgreSQL documentation for valid combinations.



**Note:** The schema can only restrict you to a limited extent. You can produce XML that is valid with regard to the schema, which results in, for example, an index definition that is still not accepted by PostgreSQL. The error logging following such misconfiguration should be sufficient to get you back on track.

The following is the definition of the index used by the Order\_Management sample search FactSheet Terms and Meta:

```
<index>
  <name>ginindex_text_and_metadata</name>
  <unstructuredContent>true</unstructuredContent>
  <method>gin</method>
  <fields>
    <field>
      <column>CID</column>
    </field>
    <field>
      <toTVector>
        <config>english</config>
        <column>FACTSHEET_TEXT</column>
      </toTVector>
    </field>
    <field>
      <expression>(to_tsvector('english', metadata -> 'AUTHOR'))</expression>
    </field>
    <field>
      <expression>(to_tsvector('english', metadata -> 'TITLE'))</expression>
    </field>
    <field>
      <expression>(ORDER_MANAGEMENT.jsonfieldtodate(metadata, 'MODIFIED'))</expression>
    </field>
    <field>
      <expression>(ORDER_MANAGEMENT.jsonfieldtodate(metadata, 'CREATED'))</expression>
    </field>
    <field>
      <expression>(metadata -> 'KV_FORMAT_NAME')</expression>
    </field>
  </fields>
</index>
```

The ORDER\_MANAGEMENT.jsonfieldtodate function is used to store the JSONB field values for MODIFIED and CREATED as date in the index.

### 3.4.2.5 Multi-language custom indexes

This is a continuation of the example in [Storing multi-language extracted text](#).

When the extracted text is stored in language-specific columns, the custom gin indexes must also use column names with language extension (for example, FACTSHEET\_TEXT\_en, FACTSHEET\_TEXT\_fr, and FACTSHEET\_TEXT\_es). This requires the definition of one index per language.

For convenience, OpenText Information Archive supports metadata index definitions that support multiple languages. One single metadata definition in the metadata lead to one index per language. As index names must be unique, language-specific index names are created with the language code as a name extension. For instance, when languages en, fr, and es are used, an index name ginindex leads to indexes with the names ginindex\_en, ginindex\_fr, and ginindex\_es.

To make the index definition support multiple languages, leave out the `config` definition.

In the multi-language use case, you can leave out the language-specific search configuration (`<config>` in an `<IndexField>` or `<toTSVector>`):

```
<field>
  <toTSVector>
    <column>FACTSHEET_TEXT</column>
  </toTSVector>
</field>
```

Internally, OpenText Information Archive creates the index with this field index for every language, using the language-specific search configuration and language-specific column name.

Use the `<metadataField>` element as child element of `<toTSVector>`.

With this `<metadataField>` element, you can specify a JSON field in the metadata that is stored as JSON. Metadata field names are `TITLE`, `AUTHOR`, `KV_FORMAT_NAME`, `CREATED`, and `MODIFIED`.

This `<metadataField>` element is only supported to be used within a `<toTSVector>` and it can only be applied to the column that is referred to as `<metadataColumn>` in the `<refText>` element:

```
<field>
  <toTSVector>
    <metadataField>TITLE</metadataField>
  </toTSVector>
</field>
```

Internally, the index with this field is created for every language, where the language-specific search configuration and language specific column name for metadata is used.

In an index field definition using `<expression>`, use the language prefix to specify a column has language-specific columns:

```
<field>
  <expression>(ORDER_MANAGEMENT.jsonfieldtodate(language:metadata, 'MODIFIED'))</expression>
</field>
```

Internally, the index is created for ever language, where the language-specific column name is used for this field.

If the index field definition already uses a language-specific column name, the column name is not adjusted.

If the column name does not have to be adjusted by for any index field in the index definition, the index is only created once.

If a `<config>` element is specified under `<field>` or `<toTSVector>`, this search configuration overrides the language-specific configuration. This may be of use when you do not want to use a language-specific search configuration for a full text

search condition. For instance, when searching full text on a name, you do not want to filter out stop words. In that case, you can use the search configuration with name simple:

```
<toTSVector>
  <config>simple</config>
  <metadataField>AUTHOR</metadataField>
</toTSVector>
```

**Creating a multi-language SQL query** will have information on how to define an SQL query that can be used with multiple languages.

## 3.5 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. Use IA Shell to backup the database:

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

For more information, see Section 2.8.1 “backup-database” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*.

2. Verify that the back up was completed.



**Tip:** Use, for example, 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.



### Caution

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

- a. Drop the database that you backed up directly from its data node using pgAdmin or any other tool of your choice
- b. Verify that the database was deleted by running an OpenText Information Archive search on it. You should see error messages that the search fails.
- c. Restore the content with IA Shell. Access IA Shell and enter:

```
iashell> connect
Connected to "http://localhost:8765/services" as adam@iacustomer.com
iashell> cd applications/Baseball/databases/Baseball-sql-db
iashell> restore-database
OK
```

**Caution**

If the data node owner does not have CREATEDB privileges, an empty database needs to be pre-created in PostgreSQL before running the `restore-database` command.

- d. Login as a Developer and do a search to verify that you get the results back for the search.

This operation is confirmed.

## 3.6 Editing the stores for a database

If a table application has not specified a retention store, edit the store to use the correct type.

Go to the database and select **Edit Stores**. Associate the retention data store used for granular retention.

The default for the retention store is set to a default store, which is a regular store, and must be changed to use a library store. If you plan on either applying retention or holds on records in an application, you must have a library store to store the information.

It is also recommended that you configure a regular store to store the backups of the retention information. If you chose to not do this, then OpenText Information Archive has no way of recovering from a disaster on your retention information.

The configuration of these stores is on the database for a table archive. For more information, see [Configuring a table application](#).

## 3.7 Using the JDBC driver to retrieve table data

The OpenText Information Archive Java Database Connectivity (JDBC) driver provides the ability to retrieve table-based archived data using SQL queries. JDBC is an API for Java that allows client access to databases. The JDBC driver can be configured with any open source SQL client (for example, SQuirreL SQL Client).

For table archiving, the JDBC driver provides a subset of PostgreSQL standard, read-only access to relational table data stored in OpenText Information Archive.

JDBC driver features include:

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

### 3.7.1 Configuring a SQL client to use the JDBC driver

For a SQL client to use the OpenText Information Archive JDBC driver to retrieve table data, you must specify the following parameters to make the connection and authenticate the client. For more information about authentication tokens, see *OpenText Information Archive REST API Developer's Guide* on support.opentext.com (<https://support.opentext.com/>).

**host**

The hostname of an instance of IA Web App or IA Server. In most cases, you should point to IA Web App because IA Web App is more likely to be accessible to clients (for example, if IA Web App is in a DMZ).

Mandatory: Yes

**port**

The port number for the IA Web App or IA Server instance (for example, 8080).

Mandatory: Yes

**tenant**

The name of the tenant (for example, INFOARCHIVE).

Mandatory: Yes

**application**

The name of the application (for example, Baseball).

Mandatory: Yes

**database**

The name of the database (for example, Baseball-sql-db).

Mandatory: Yes

**user**

The name of the OpenText Information Archive user you want to use to access the database (for example, sue@iacustomer.com, if you are using example user accounts).

Mandatory: Yes

**password**

The password for the OpenText Information Archive user.

Mandatory: Yes

**clientId**

The client ID for the OpenText Information Archive JDBC driver. This value must correspond with the `clientId` parameter for the JDBC driver in IA Web App's `<IA_ROOT>/config-default/iawa/application-CLIENTS.yml` file. By default, the `clientId` parameter in this file is set to `infoarchive.jdbc`.

If, however, you are using OTDS SSO, use the value for the JDBC client ID provided during setup.

Mandatory: Yes

**clientSecret**

The client secret for the OpenText Information Archive JDBC driver. This value must correspond with the `clientSecret` parameter for the JDBC driver in IA Web App's `<IA_ROOT>/config-default/iawa/application-CLIENTS.yml` file.



**Note:** If, however, you are using OTDS SSO, use the client secret provided during setup.

Mandatory: Yes

**authEndpoint**

The URL to the authentication endpoint for Gateway, which is the same host as IA Web App. The URL points to a host, optionally a port, and a path to the authentication endpoint.

For example, `authEndpoint=localhost:8080/infoarchive-webapp/oauth/token`.

Mandatory: Yes

**ssl**

Whether the client connects over TLS/SSL. Valid values are `true` and `false`. If you do not include this parameter in the connection string, then `false` is assumed.

If you want to connect over TLS/SSL, additional setup is required. For more information, see [Setting up TLS/SSL for the JDBC driver](#).

Mandatory: No

These parameters are used in a connection string for authentication. The connection string is a String object which should have a value pointing to OpenText Information Archive services and also must be compliant with the JDBC protocol and the JDBC driver's implementation.

The following example shows a connection string where the host is an instance of IA Web App that was installed as a standalone Spring Boot application:

```
jdbc:ia://iawebapp_host:8080/restapi?tenant=INFOARCHIVE&application=Tickets&database=Tickets-sql-db&user=connie@iacustomer.com&password=password&clientId=infoarchive.jdbc&clientSecret=secret&authEndpoint=iawebapp_host:8080/oauth/token
```

The following example shows a connection string where the host is an instance of IA Web App that was deployed to Apache Tomcat:

```
jdbc:ia://tomcat_host:8080/infoarchive-webapp/restapi?tenant=INFOARCHIVE&application=Tickets&database=Tickets-sql-db&user=connie@iacustomer.com&password=password&clientId=infoarchive.jdbc&clientSecret=secret&authEndpoint=tomcat_host:8080/infoarchive-webapp/oauth/token
```

The following example shows a connection string where the host is an instance of IA Server:

```
jdbc:ia://iaserver_host:8765?tenant=INFOARCHIVE&application=Baseball&database=Baseball-sql-db&user=sue@iacustomer.com&password=password&clientId=infoarchive.jdbc&clientSecret=secret&authEndpoint=iawebapp_host:8080/oauth/token
```

At a minimum, the connection string must contain at least the host and port. For example:

```
jdbc:ia://hostname:8080/restapi
```

You can either add the other parameters to the URL as query parameters, or if you are writing your own code using the JDBC API, you can specify the properties as in the following example:

```
com.emc.ia.jdbc.JdbcDriver:  
final JdbcDriver jdbcDriver = new JdbcDriver();  
DriverManager.registerDriver(jdbcDriver);  
final Properties info = new Properties();  
info.setProperty("user", "<USERNAME>");  
info.setProperty("password", "<PASSWORD>");  
info.setProperty("clientId", "<CLIENT_ID>");  
info.setProperty("clientSecret", "<CLIENT_SECRET>");  
final Connection connection = DriverManager.getConnection(<CONNECTION_STRING>, info);
```



**Note:** When using the JDBC API directly, these parameters are Properties objects, but some GUI-type clients might have a specific user interface for specifying name/value property pairs. You can mix and match these styles. For example, you can specify all values as query parameters, all values as Properties, or some as query parameters and others as Properties.

### 3.7.2 Example: configuring SQuirreL SQL client to use the JDBC driver to retrieve table data

After you have ingested table data into OpenText Information Archive, you need to set up the JDBC driver. The following example uses SQuirreL SQL Client version 4.0.0 and the Baseball example application.

1. Download the SQL Client Squirrel from the SQuirreL SQL website and install it.
  2. Click the **Drivers** tab, then click the **Create a New Driver** (+) icon.
  3. In the **Name** field, type **infoarchive-jdbc-driver**. The JDBC driver can be found in the **lib** folder of IA Server distribution.
  4. In the **Example URL** field, type the following:
- ```
jdbc:ia://<HOSTNAME>:8080/restapi
```
5. Click the **Extra Class Path** tab, then click the **Add** button.
  6. Select the JDBC driver (**infoarchive-jdbc-driver.jar**) in the **lib** folder of an IA Server instance.
  7. Click **List Drivers**, and then click **OK**.
  8. Click the **Aliases** tab, then click the **Create New Alias** (+) icon.
  9. In the **Name** field, type **infoarchive-baseball**.
  10. Select **infoarchive-jdbc-driver** from the **Driver** drop-down list.
  11. Specify the **URL** (the connection string), the **User Name**, and **Password**.

12. Click the **Auto logon** check box.
13. Click the **Properties** button, then click the **Driver properties** tab.
14. Click the **Use driver properties** check box.
15. For each of the following properties, click the **Specify** check box and specify the value for the property in the **Value** column:
  - **tenant**
  - **application**
  - **database**
  - **user**
  - **password**
  - **clientId**
  - **clientSecret**
  - **authEndpoint**

 **Note:** You do not need to specify the host and port properties in this window because the URL in the **Add Alias** window includes these values.

16. Click **OK**, then double-click the **infoarchive-baseball** alias to open a connection.

### 3.7.3 Setting up TLS/SSL for the JDBC driver

If the OpenText Information Archive component that you are connecting to with the connection string is set up to use TLS/SSL, then you must also set up TLS/SSL for the JDBC driver. For more information, see Section 13.2.2 “Setting up TLS/SSL in a demo configuration” in *OpenText Information Archive - Installation Guide (EARCORE-IGD)*.

**!** **Important**

PostgreSQL supports PEM and DER formats for private keys used for SSL communication. IA Server uses two different protocols when connecting to PostgreSQL:

- JDBC
- Directly using PostgreSQL binaries, such as `psql` or `pg_dump`, depending on the functionality involved.

For the majority of communication, JDBC protocol is used, but for features such as application offlineing, PostgreSQL binaries are used directly. However, it seems that through a JDBC connection, only DER-based keys are supported and not PEM. As of this release, private keys other than DER are not supported by OpenText Information Archive. Furthermore, under two-way SSL configuration, when the PostgreSQL Server has been configured to use client certificates, everything works fine as long as the private keys are not password protected. When private keys are password protected, JDBC communication

works fine. However, communication using PostgreSQL binaries no longer works, meaning that features such as “offlineing” an application will not work.

**To set up TLS/SSL for the JDBC driver:**

1. Set the `ssl` parameter to `true` in the connection string. For more information about parameters in connection strings, see [Configuring a SQL client to use the JDBC driver](#).
2. Obtain a signed certificate for the JDBC driver.
3. Import the certificate back into the JDBC driver’s keystore so that the signed certificate is installed.
4. Import the JDBC driver’s certificate into the truststore for the OpenText Information Archive component.
5. Set the Java Secure Socket Extension (JSSE) system properties to enable the usage of custom keystores and truststores by an OpenText Information Archive JDBC driver host application. Whether you are using one-way or two-way TLS/SSL, you must specify the following properties for any Java application by passing the parameters to the JVM:
  - `javax.net.ssl.trustStoreType`
  - `javax.net.ssl.trustStore`
  - `javax.net.ssl.trustStorePassword`
  - `javax.net.ssl.keyStoreType`
  - `javax.net.ssl.keyStore`
  - `javax.net.ssl.keyStorePassword`

For example:

```
-Djavax.net.ssl.trustStoreType=jks
-Djavax.net.ssl.trustStore="E:\\infoarchive\\\\certificates\\\\client\\
\\client_truststore.jks"
-Djavax.net.ssl.trustStorePassword=abcdefg
-Djavax.net.ssl.keyStoreType=pkcs12
-Djavax.net.ssl.keyStore="E:\\infoarchive\\\\certificates\\\\client\\
\\clientCertStore.p12"
-Djavax.net.ssl.keyStorePassword=abcdefg
```

### 3.7.4 Example: configuring SQuirreL SQL client to use TLS/SSL with the JDBC driver

Make the following changes in the squirrel-sql.bat file.



**Note:** You should back up the squirrel-sql.bat file before you make the following changes.

Before:

```
start "SQuirreL SQL Client" /B "%LOCAL_JAVA%" -Xmx256m
-Dsun.java2d.noddraw=true -cp %SQUIRREL_CP% -splash:"%SQUIRREL_SQL_HOME%/icons/
splash.jpg"
net.sourceforge.squirrel_sql.client.Main %TMP_PARMS%
```

After:

```
start "SQuirreL SQL Client" /B "%LOCAL_JAVA%" -Xmx256m
-Dsun.java2d.noddraw=true -Djavax.net.ssl.trustStoreType=jks
-Djavax.net.ssl.trustStore="E:\|infoarchive\|certificates\|client\|client_truststore.jks"
-Djavax.net.ssl.trustStorePassword=abcdefg -Djavax.net.ssl.keyStoreType=pkcs12
-Djavax.net.ssl.keyStore="E:\|infoarchive\|certificates\|client\|clientCertStore.p12"
-Djavax.net.ssl.keyStorePassword=abcdefg -cp %SQUIRREL_CP%
-splash:"%SQUIRREL_SQL_HOME%/icons/splash.jpg"
net.sourceforge.squirrel_sql.client.Main %TMP_PARMS%
```

# Chapter 4

## Configuring a SIP application

*OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)* explains the conceptual differences between table-based and SIP-based archiving. There are also significant differences in the way they are configured.

The central configuration object in SIP-based archiving is the holding. This chapter explains the holding configuration, which includes settings for ingestion, cache and library management, storage, and more.

### 4.1 SIP datamodel

The SIP datamodel is based on OAIS terminologies. For more information see the OAIS website. The following is a list of the main configuration objects for SIP applications:

|                         |                                                                                                                                        |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Holding</b>          | A holding is a logical destination archive where to ingest and store data, usually of the same type that share common characteristics. |
| <b>Holding Crypto</b>   | This object defines cryptographic parameters to apply for an archive holding.                                                          |
| <b>Confirmation</b>     | This object defines the confirmation messages to generate in reaction to an AIP event.                                                 |
| <b>Delivery Channel</b> | This object defines a destination where to write the confirmation message.                                                             |
| <b>PDI</b>              | This object contains the configuration to apply during the ingestion.                                                                  |
| <b>PDI Crypto</b>       | This object contains the encryption configuration to apply during the ingestion.                                                       |
| <b>PDI Schema</b>       | This object contains the XML Schema to use to validate a PDI XML file during the ingestion.                                            |
| <b>Transformation</b>   | This object defines the XQuery/XSLT to use to perform a transformation.                                                                |
| <b>Ingest Node</b>      | This object defines parameters for the ingestion and the enumeration processes.                                                        |
| <b>Receiver Node</b>    | This object defines parameters for the reception process.                                                                              |
| <b>Library Policy</b>   | This object defines the rules how to manage the libraries.                                                                             |
| <b>AIC</b>              | This object defines a set of AIPs for a search. AIC is an acronym for Archival Information Collection.                                 |
| <b>Query</b>            | This object defines every option needed to build a query.                                                                              |
| <b>Store</b>            | This object defines a store used by an application. Stores can be for search results, confirmation, or for default use.                |

## 4.2 Holding

A holding is the central configuration object in SIP-based archiving. This configuration object centralizes directly or indirectly many settings, including storage, retention, schema, library management, indexing, and ingestion.

A holding is also a logical destination where data that shares common characteristics is archived. The common characteristics can be the same source application, same format, same type of data, or data that belongs to the same business entity.

An application can contain multiple holdings. Multiple holdings can exist for a single data type.

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

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

- Number of records per SIP
- Ingestion mode
- Indexes
- Partitioning keys

In IA Web App, the list of existing holdings for a SIP application appears on the application's **Holdings** tab. The list is a table that contains the following information:

|                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Holding Name</b>                                                                 | Indicates the name of the holding. If a holding is in use, the name can no longer be edited.                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|  | Allows you to either: <ul style="list-style-type: none"><li>• Edit a holding (refer to <a href="#">Editing a holding</a> for more information).</li><li>• Export the holding configuration to a ZIP file.</li></ul>                                                                                                                                                                                                                                                                                                                                           |
| <b>Holding Description</b>                                                          | Contains a brief description of the holding and its purpose. Even if a holding is in use, its description can be edited.                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>In Use</b>                                                                       | Indicates whether the holding is currently being used.<br><br>In use is calculated by looking for any AIP that is ingested and if the holding is used in any search.<br><br>If a holding is not in use, user can delete it.<br><br>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. |

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

|                            |                                                                                                                                                                                              |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Summary</b>             | Contains general information about the holding.                                                                                                                                              |
| <b>Data &amp; Contents</b> | Contains information about search fields, encrypted fields and unstructured content for the holding.                                                                                         |
| <b>Ingestion</b>           | Contains information about ingestion mode, library management, encryption, custom metadata, and indexation of unstructured contents for the holding.                                         |
| <b>Stores</b>              | Contains information about stores that are connected to the holding.                                                                                                                         |
| <b>Confirmation</b>        | Contains information about confirmation options for the holding, as well as the holding's settings for SIP, AIU, and PDI queries, as well as the <b>Add Root Element Confirmation</b> field. |
| <b>Retention</b>           | Contains information about the retention policy, retention classes, retention partitioning and disposition policy for the holding.                                                           |

#### To view the list of holdings for a SIP application:

- In IA Web App, click Applications > [APPLICATION NAME] > Holdings.

### 4.2.1 Creating a holding with the holding wizard

The **Holdings** tab includes a wizard that helps a Developer create and configure a holding for an application. The OpenText Information Archive holding configuration wizard helps you build a 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, and so on.

Before you start the holding wizard, make sure that a space with a file system storage has already been created for your application. Otherwise, you will have to stop the wizard partway through, create a space and a file system storage, and then resume the wizard.

You can perform the following actions with the wizard:

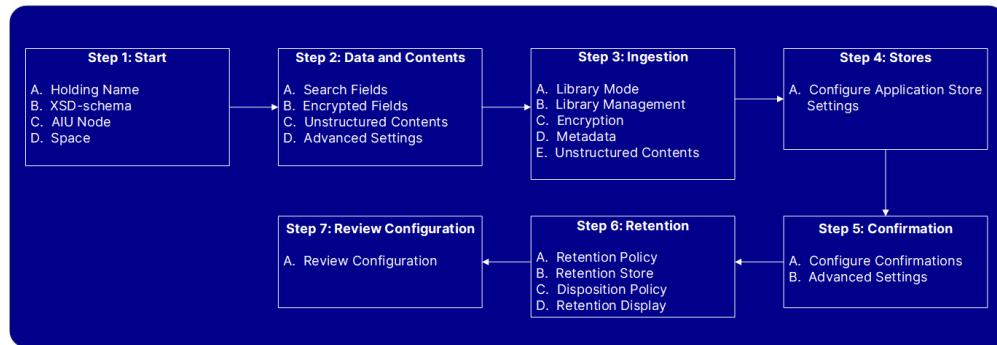
- You can save an unfinished holding configuration and load the saved configuration file later to continue with the configuration. When you load an existing holding configuration that you saved earlier, you are returned to the **Holdings** tab and resume from there.

You can also delete an unfinished holding configuration.

- When a holding configuration is fully created, you can install it.
- You can export a holding configuration into a declarative format (for more information, see [Declarative configuration](#)).

The wizard allows you to create and configure holdings. It allows you to generate a holding based on declarative configuration. The following illustrates the main steps

required to create a holding. The numbers relate to the steps in the following procedure:



#### To create a holding:

1. On the [APPLICATION NAME] > Holdings tab, click + > Create Holding to initiate the wizard.
2. Complete the steps in the following sections.

##### 4.2.1.1 Step 1: Start

1. The **Start** step of the wizard allows you to enter the following preliminary information:

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Holding Name</b>      | Specify a descriptive name that uniquely identifies the holding. The first character of the name cannot contain the following characters: # < > ! = \$ ` & * ( ) + \ / " " ' : ?   @ % { } [ ] , - The name cannot contain the following characters: # < > ! = \$ ` & * ( ) + \ /                                                                                                                                                                     |
| <b>Description</b>       | Enter a brief description of the holding that outlines its purpose.                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Application Space</b> | Select the space associated with the application. If only one space has been configured for the application, it will appear by default.                                                                                                                                                                                                                                                                                                               |
| <b>Local File System</b> | Select the local file system associated with the application's space. If only one has been configured, it will appear by default. This file system will be used later to create a library store and a regular store (if none has been set before).                                                                                                                                                                                                    |
| <b>Schema File</b>       | <p>There is an option of uploading a new XSD file or selecting an existing schema file. Refer to <a href="#">Mapping XSD data types</a> 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 can upload several schemas, if needed.</p> |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Archival Information Unit (AIU) Node</b></p> <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"> <li>• Click the hand cursor on the schema diagram and drag it to a new section of the page.</li> <li>• Click the plus sign (+) on a node to expand it; click the minus sign (-) to collapse a node. You also have the options of collapsing and expanding all of the nodes by clicking the <b>Collapse all</b> and <b>Expand all</b> options.</li> <li>• Drag the slide bar to zoom in and out of the schema diagram.</li> </ul> <p> <b>Note:</b> 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 unit (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 (eas_pdi.xml) in a SIP describes all the AIUs in the package. An AIU in the eas_pdi.xml 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> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

2. Click **Next**.

#### 4.2.1.2 Step 2: Data and Contents

1. The **Data & Contents > Search Fields** step allows you to select the fields to be made available for search composition. The Search Designer should select at least one search field.

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

Use the following navigational operations to locate the desired fields:

- Click the hand cursor on the schema diagram and drag it to a new section of the page.
- Click the plus sign (+) on a node to expand it; click the minus sign (-) to collapse a node. You also have the options of collapsing and expanding all of the nodes by clicking the **Collapse all** and **Expand all** options.
- Drag the slide bar to zoom in and out of the schema diagram.

b. Click **Select**.

c. Set the **Full Text** and **Partition Key Method** for each field:

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Full Text</b>            | Creating indexes speeds up searches but consumes more storage space. This option is only available with the STRING field type.<br><br>Creating full text indexes is mandatory to perform searches with full text operators (contains, for example).                                                                                                                                                                                                                                                         |
| <b>Partition Key Method</b> | For more information about partition keys, see <a href="#">Using partition keys</a> . The following partitioning techniques at the time of data ingestion help with search performance: <ul style="list-style-type: none"> <li>• <b>None:</b> No partitioning techniques were added to the AIP object prior to ingestion</li> <li>• <b>Min/Max:</b> Uses the minimum and maximum number of records in the package.</li> <li>• <b>List:</b> Uses the unique values of the records in the package.</li> </ul> |

It is possible to delete previously selected fields.

- d. Click **Next**.
2. The **Data & Contents > Encrypted Fields** tab allows you to select the fields from a PDI schema that should be encrypted at the time of ingestion.

- a. Select the **Data encryption object** from the drop-down list.
- b. Click **Select from Schema** to select the desired fields.

Use the following navigational operations to locate the desired fields:

- Click the hand cursor on the schema diagram and drag it to a new section of the page.
- Click the plus sign (+) on a node to expand it; click the minus sign (-) to collapse a node. You also have the options of collapsing and expanding all of the nodes by clicking the **Collapse all** and **Expand all** options.
- Drag the slide bar to zoom in and out of the schema diagram.

- c. Click **Select**.
- d. Click **Next**.

The **Next** button may be disabled for these reasons:

- If an encryption object was selected and no field has been selected; or
- If at least one field is selected, but no encryption object has been selected.

3. The **Data & Contents > Unstructured Contents** step allows you to define one or several content fields.
  - a. For the **Is there unstructured content associated with record?** field:
    - If checkbox is not selected, click **Next**.
    - If checkbox is selected, complete the following sub-steps:

- i. Select a **Content encryption object**.
- ii. Click **Select from Schema**.

Use the following navigational operations to locate and select the desired fields:

- Click the hand cursor on the schema diagram and drag it to a new section of the page.
  - Click the plus sign (+) on a node to expand it; click the minus sign (-) to collapse a node. You also have the options of collapsing and expanding all of the nodes by clicking the **Collapse all** and **Expand all** options.
  - Drag the slide bar to zoom in and out of the schema diagram.
- iii. Complete the following for each field selected:

| Field Name         | Indicates the name of the field being configured.                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Encryption</b>  | Select 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.                                                                                                                                                                                                                                                                |
| <b>Compression</b> | Select to save storage space by allowing unstructured content files to compress with the GZIP compression algorithm.                                                                                                                                                                                                                                                                                                                       |
| <b>Default</b>     | Select a MIME type that represents file formats for unstructured content (for example, documents, images or PDFs) from the list. Select a type of content file format from the drop-down list.<br>Click <b>Add New File Format</b> to add a format that does not appear in the list. <ul style="list-style-type: none"> <li>• Select a <b>Mime Type</b>.</li> <li>• Select a <b>Mime Sub Type</b>.</li> <li>• Click <b>Add</b>.</li> </ul> |
| <b>Offset Mode</b> | Select to configure the file offset values. If selected, you are prompted to select <b>Page Offset</b> (start page node) and <b>Page Length</b> (number of pages node). Only INTEGER, BIGINTEGER, and LONG nodes are enabled.                                                                                                                                                                                                              |
| <b>Dynamic</b>     | Choose a dynamic MIME type node in the schema viewer dialog. Only STRING nodes are enabled.                                                                                                                                                                                                                                                                                                                                                |

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Node</b>    | Select the dynamic hash value node in the schema viewer dialog. Only STRING nodes are enabled. The value of this field is used for content integrity check during ingestion phase.<br><br>Hash values for each content file can either be generated based on configuration during the ingestion or provided in the PDI. In the later, OpenText Information Archive generates a hash value from the corresponding content and, if this value does not match the provided value, ingestion will fail. Hash algorithm and encoding could either be dynamically provided by the PDI or statically set in the configuration. When a content is downloaded, the corresponding hash value is provided in the headers of the HTTP response. |
| <b>Default</b> | Select hash algorithm and hash encoding values from the lists provided.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Dynamic</b> | Select hash algorithm and hash encoding values from the schema viewer dialogs. Only STRING nodes are enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

- b. Click **Next**.

4. The **Data & Contents > Advanced Settings** step allows you to configure the OpenText Output Transformation plugin, which provides on-demand dynamic transformation from natively stored AFP or PDF formats to general, user-consumed formats, such as accessible PDF or XML. More information about setting up the plug-in can be found in *OpenText Output Transformation for OpenText Information Archive Installation Guide*.

For the **Configure** box:

- If not checked, click **Next**.
- If checked, complete the following sub-steps:
  - a. Select the **Mode** of content transformation, which contains the following options:
    - **AFP > PDF**: Transforms AFP print stream format directly to accessible PDF format.
    - **AFP > Optimized AFP > PDF**: Transforms AFP print stream format, first, to an intermediate optimized AFP format, and then to the accessible PDF format.

Complete the following fields to configure the OpenText Output Transformations project files:

- b. If **AFP > PDF** is selected, upload indexing project zip file by clicking on the corresponding **Upload Project ZIP** button.
- c. If **AFP > Optimized AFP > PDF** is selected, upload both indexing and rendering project ZIP files by clicking on the corresponding **Upload Project ZIP** buttons.

Additional information about project files can be found in the collapsed **Show Details** section under the **Upload Project ZIP** buttons.

Uploaded project files will be in the exported ZIP configuration of the holding.

- d. Click **Next**.

#### 4.2.1.3 Step 3: Ingestion

1. The **Ingestion > Library Mode** step allows you to indicate which library mode will be applied to the holding and which events will be confirmed.

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

For the Pooled mode, each AIP is ingested in individual libraries. The Close job selects AIPs that can be pooled and merges their libraries into a shared definitive library.

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

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

|                                                    |                                                                                                     |
|----------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <b>Maximum number of packages in library</b>       | A library stores archived data. Specify the maximum number of AIPs that can be stored in a library. |
| <b>Maximum number of records in a library</b>      | Specify the maximum number of records that a library can contain.                                   |
| <b>Close immediately when the quota is reached</b> | Select to close the library once it contains the maximum number of records.                         |

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Library closing date</b> | Specify the close condition for the library: <ul style="list-style-type: none"><li>• <b>Manual</b></li><li>• <b>Creation Date:</b> CLOSE PERIOD + LIBRARY CREATION_DATE</li><li>• <b>Last Modified Date:</b> CLOSE PERIOD + LIBRARY LAST MODIFIED DATE</li><li>• <b>Close Hint Date</b><ul style="list-style-type: none"><li>– If selected, enter a value in the <b>Close period in days</b> field, and</li><li>– Enter the <b>Closing Hint Date Query</b>.</li></ul></li></ul>                                                                                                                                       |
| <b>Library partitioning</b> | 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 pooled libraries. Multiple pooled libraries can be open at a single point in time for a given library pool. To archive data from 2018 Q3 through Q4 with a quarter partitioning policy, the corresponding 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. |
| <b>AIU Threshold mode</b>   | 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. Click **Next**.
2. The **Ingestion > Library Management** step allows you to configure the library management options during the holding creation. This step contains various scenarios to distribute data between databases. The library management options can be changed at any time. Access the **Holdings** tab and, for the holding being updated, click  > **Edit library management configuration**. Complete the following:
- a. Select the **Load Balancing Mode**, which contains the following options:
- **Primary:** Creates or restores the library to its primary location only. If the selected data node is not available, a background ingestion request is created and the system waits until the data node is available. This mode is the default one. In this mode, provide only one primary library.
  - **Failover:** Attempts to create or restore the library to the primary location. If unavailable, additional locations are attempted sequentially. If no library is ready, a background ingestion request is created. In this mode, provide a primary library and additional locations.

- **Random:** Attempts to create or restore the library to primary or additional locations randomly. In this mode, provide a primary and additional library locations. The primary location does not have more priority than the additional ones.

Complete the following fields to configure the structured data root libraries.

- b. Select the **Primary Location** where the library must be created and restored. The default is to create a new library store, but it is possible to choose from an existing library store.
  - c. Select the **Restore Mode**. There are two modes:
    - To current parent: In this mode, the library uses directly its (precedent) parent to be restored.
    - To parent defined at holding level: In this mode, the holding configuration is used to choose how to restore the library
  - d. Set the **Cache Lock Period (days)** value. This indicates the number of days that the library will be temporarily protected from package removal performed by the Cache-Out job.
  - e. Click **Next**.
3. For the **Ingestion > Encryption** step, see Section 6.3 “Allotting encryption keys” in *OpenText Information Archive - Encryption Guide (EARCORE-AGE)* and Section 6.2.1 “Holding crypto” in *OpenText Information Archive - Encryption Guide (EARCORE-AGE)* to learn how to configure encryption for a new holding.
  4. The **Ingestion > Metadata** step allows you to declare custom fields at the package level to define common information for all AIUs/records. The custom fields can be included in search results, as well as exports. The number of custom fields is limited to 50 per holding, and each value field cannot exceed 256-characters in length.

**! Important**

During ingestion, if the SIP descriptor contains custom fields that do not match the **Field Type** entered here, the ingestion will fail.

- a. Click the **Declare new field** button and complete the following:
  - **Field Name:** Enter the technical name of the custom attribute.
  - **Field Label:** Enter a label for the custom attribute that will be displayed in the **Packages** page.
  - **Field Type:** Select the type of the custom attribute from one of the following: STRING (default), DATE, DATETIME, INTEGER, LONG, DOUBLE, FLOAT, BIGINTEGER, BIGDECIMAL.
- b. If required, click **Declare new field** to enter another field. Otherwise, click **Next**.

5. The **Ingestion > Unstructured Contents** step allows you to index unstructured contents and to search the extracted text.

Select **Index unstructured contents** field and complete the following steps; otherwise, click **Next**.

- a. Select **Mime Type** to extract:
  - **All**: All unstructured contents will be extracted.
  - **Include**: Only the unstructured contents with the mime types defined in the field provided will be processed. Regular expressions are accepted.
  - **Exclude**: Only the unstructured contents with the mime types defined in the field provided will **not** be processed. Regular expressions are accepted.
- b. Configure the **Extraction Policy** during SIP ingestion:
  - **Blank Accepted**: If selected, allows the extraction result to be blank.
  - **Error Accepted**: If selected, allows the extraction result to include errors.

If neither option is selected, only the text is expected as a result. Consequently, if the result is blank or an extraction error occurs, ingestion and, respectively, the Post Ingest Processing job will fail.
- c. Click **Support Container Files** to index embedded attachments.

#### 4.2.1.4 Step 4: Stores

The **Stores** step allows you to configure the application store settings.

1. Complete the following, as required:

|                                            |                                                                                                                            |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>Keep SIP file after ingestion</b>       | If selected, a SIP file that has been successfully ingested will be stored                                                 |
| <b>Keep SIP file upon any invalidation</b> | If selected, a SIP file that has not been successfully ingested will be stored                                             |
| <b>Enable Log store</b>                    | If selected, reception and ingestion logs will be stored, either in a common store with other data or in a separate store. |

|                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Same store for all contents</b> | <p>If selected, all regular content will be stored in a common store. Select a value for the following:</p> <ul style="list-style-type: none"> <li>• <b>Common Store:</b> If not selected, select values for the following:           <ul style="list-style-type: none"> <li>– <b>Reception Store:</b> This store is used to save the original SIP (<code>sip_zip</code>). The SIP is archived as other content if the customer decides to keep the SIP file after the ingestion.<br/>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.<br/>The reception store is not a working directory, such as <code>data/tmp</code>.</li> <li>– <b>Log Store:</b> Saves log files generated during the reception, ingestion and invalidation.</li> <li>– <b>XML Store:</b> Saves the SIP XML, as well as the original PDI XML file (<code>.gzip</code> format), if the customer decides to keep the PDI after the ingestion.</li> <li>– <b>Unstructured Content Store:</b> Saves the CIs. This includes the CI Container and the RI XML files.</li> <li>– <b>Library Backup Store:</b> Saves the library backup containing the metadata.</li> <li>– <b>Analytics Rendition Store:</b> 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.</li> <li>– <b>Retention Backup Store:</b> A store for the library backup of compliance information. It is used in disaster recovery operations.</li> <li>– <b>Staging Store:</b> This store indicates where the read/write operations prior to closing the aggregate will occur (reception, ingestion, invalidation, etc.). This type of store is only available for holdings with the aggregate mode.</li> </ul> </li> </ul> |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

2. Select a **Common Store** if you do not want to use the default one.

3. Click **Next**.

#### 4.2.1.5 Step 5: Confirmation

1. The **Confirmation > Confirmations** step allows you to select the event types for which you want to generate a confirmation message (**Receipt**, **Available**, **Storage**, **Purge**, **Invalidation**, **Rejection**). It also allows you to select where the confirmation message should be delivered: as a message attached to the confirmation audit (**Audit**) and/or as a message written to the confirmation store (**Store**). If you select **Audit**, you can view the content of the confirmation as a downloadable file attached to the confirmation audit after running the Archive Audits job.

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. For more help on confirmations, see [Using the confirmation](#).

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.
- Confirmations can be written to following types of storage:
  - **File Storage**
    - Local File System
    - PowerScale
  - **Object Storage**
    - Pure Storage FlashBlade
    - Dell EMC Elastic Cloud Storage
    - NetApp StorageGrid
    - Amazon S3 Storage
    - Microsoft Azure Blob Storage
    - Google Cloud Storage
    - PowerScale OneFS S3
    - Scality RING
  - **Custom Storage**
    - Custom Storage
  - **Archive Center**
    - OpenText Archive Center
    - OpenText Core Archive

- Click **Next**.
2. The **Confirmation > Advanced Settings** step allows you to edit SIP and AIU queries. Furthermore, there are options to configure the destination (sub-path and filename).
    - a. By default, a confirmation message is based on a SIP descriptor enriched with a Package ID, the confirmation type and date. This message can be changed by entering information in the **SIP Query**, if desired. Click **Template** to enter a query template in the field. Click **Clear** to remove data from the field.
    - b. If the message needs to return record information, it is possible to configure an AIU query. This query is possible only for AVAILABLE, STORAGE and PURGE events. It will overwrite the SIP query when it is defined.  
If desired, check the **Add Root Element Confirmation** box and enter the applicable information in the **AIU Query** field. You can also click **Template** to enter a query template in the field. Click **Clear** to remove data from the field.
    - c. Click **Next**.

#### 4.2.1.6 Step 6: Retention

1. 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:

##### Package

Retention will be applied to entire package.

The retention policy configured at the application will be applied. You can configure a default retention class to override this choice. During the ingestion, the retention class to apply can be driven by the package description with the `sip>dss>retention_class` field. If the retention class is not listed below, ingestion will result in an error.

**Retention Classes** 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. By default, the retention policy configured for the application is applied. Select the checkbox to configure a default retention class to override the retention policy configured for the application.
- iii. Enter a **Name** for the retention class.
- iv. Select a **Retention Policy** from the list.

When choosing the retention policy, if a retention policy is disabled, the system displays a **Disabled** message after the policy's name. Unless the policy is enabled, no disposition processing will be done for the package until the retention policy is enabled.

#### Granular

Retention is applied to each record.

##### If Granular was selected, complete the following:

- i. **Retention Policy:** Select a retention policy to use as the base date. Only duration-based retention policies can be used. Once selected, the details of the policy appear below.
- ii. Click  to select the **Base Date Field**.

2. The **Retention > Retention Store** step allows you to select the library store.
3. The **Retention > Disposition Policy** step allows you to configure how disposition impacts structured data and unstructured contents.

For the **Structured Data** section, select one of the following:

- Select **Keep PDI XML Gzip** to ensure that the PDI XML Gzip is not touched during disposition.
- Select **Delete PDI XML Gzip** to delete PDI XML Gzip if a granular disposition occurs (only if there is no retention placed on the storage).

For the **Unstructured Contents** section, select one of the following:

- **Keep:** Ensures that CIs are not changed during AIU disposition.
- **Blank:** Ensures that CIs are overwritten with a blank sequence during the AIU disposition (only if partial updates are supported and no retention has been placed on the storage).
- **Shrink:** Ensures that, when the shrink ratio is reached, the CI container and RI XML are re-factored during the disposition (only if no retention has been placed on the storage). If selected, set the **Shrink Ratio**.
- **Blank and Shrink:** Ensures that, when the shrink ratio is reached, the CI container and RI XML are re-factored during the disposition. Otherwise, the CIs are overwritten with a blank sequence (only if partial updates are supported and no storage retention has been placed on the storage). If selected, set the **Shrink Ratio**.

4. The **Retention > Retention Display** step allows you to configure the search template to display records in retention sets, hold sets and purge lists.

Define a **Search Name** and **Search Set Name**. Purge list generation depends on a search template to display columns for records.



**Tip:** It is recommended to set the search name and search set name for the holding after you have created at least one search using the [edit a holding's retention feature](#).

Indicate if you want to use the first search template to display records when the above one is not defined or not authorized.

#### 4.2.1.7 Step 7: Review Configuration

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.

#### 4.2.1.8 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: “/ # < > ! = \$ & ( ) \* + \ ” ‘ ’ : ? | @ % { } [ ] ~ ; , ^ - ”
- The uploaded PDI schema should not contain any of the following tags: `<appinfo>`, `<documentation>`, `<group>`, `<attributeGroup>`, `<override>`, `<anyAttribute>`, `<key>`, `<unique>`, `<keyref>`, `<alternative>`, `<union>`, `<enumeration>`, `<assert>`, `<openContent>`, `<pattern>`, `<defaultOpenContent>`, `<assertion>`, `<redefine>`, `<selector>`, `<field>`, `<notation>`, `<fractionDigits>`, `<minExclusive>`, `<maxExclusive>`, `<list>`.
- The holding wizard requires a target namespace within the schema file.

#### 4.2.2 Editing a holding

The ability to edit a holding is only available if the holding is not associated with any ingested packages or searches. It is via the AIC on a search that indicates that it is in use.

When you want to edit an existing holding, click the button beside the desired holding. Whether you can change the following settings depends on whether the holding is in use:

- [Edit a holding's name](#)
- [Edit a holding's ingestion settings](#)
- [Edit a holding's store settings](#)
- [Edit holding's confirmation settings](#)
- [Edit holding's retention settings](#)
- [Edit a holding: If a holding is not in use, this option allows you to access the holding wizard to update all aspects of the holding.](#)

- Export to ZIP file

#### 4.2.2.1 Editing a holding's name

A holding's name can only be changed if the holding is not currently in use. However, even if the holding is in use, the description can always be updated.

#### 4.2.2.2 Editing a holding's ingestion settings

Even if a holding is in use, you can update the **Ingestion Mode** of the holding. Change the setting to Private, Pooled or Aggregate.

Even if a holding is in use, you can update the **Library Management** settings of the holding, which determines how the holding supports various scenarios to distribute data between databases.

The information presented here is the same information that was initially configured when the holding was created (refer to the [Ingestion](#) step of the holding wizard). Even if a holding is in use, you can update the following store settings of the holding:

- **Load Balancing Mode**
- For any structured data root libraries, you can edit the following:
  - **Primary Location:** The location where a library must be created and restored.
  - **Restore Mode:** The location where to restore library based on the holding's configuration.
  - **Cache Lock Period:** The number of days to protect a library temporarily from package removal performed by the Cache Out job.

Even if a holding is in use, you can update the following **Encryption** settings of the holding:

- **Key Scope**
- **Key Allotment Scheme**
- **Crypto Encoding**
- **Encrypt SIP**
- **Encrypt PDI**
- **Encrypt CI**

Even if a holding is in use, you can declare fields in the **Metadata** tab of the holding. Declare some custom fields at the package-level to define common information for all AIUS/records. These will be available to be included in search results to be displayed, as well as included in exports. The number of custom fields is limited to 50 per holding and each value field cannot exceed 256 characters in length.

**!** **Important**

During ingestion, if the SIP descriptor contains custom fields that do not match the **Field Type** entered in the metadata section, the ingestion will fail.

Even if a holding is in use, you can opt to index unstructured contents. Refer to the [Ingestion > Unstructured Contents](#) step of the Holding wizard for more information. As long as the **Index unstructured content** field is enabled, you can enable or disable the **Support Container Files** field.

#### 4.2.2.3 Editing a holding's store settings

The information presented here is the same information that was initially configured when the holding was created (refer to the [Stores](#) step of the holding wizard). Even if a holding is in use, you can update the following store settings of the holding:

- **Keep SIP file after ingestion**
- **Keep SIP file upon any invalidation**
- **Enable Log store**
- **Same store for all regular contents**
- **Keep PDI XML after commit:** This field is deprecated and will be removed in the next release. By default, the `pdi.xml` will be kept.

The following stores appear only if **Same store for all regular contents** is unchecked. Otherwise, only **Common Store** appears

- **Reception Store**
- **Log Store**
- **XML Store**
- **Unstructured Content Store**
- **Library Backup Store**
- **Analytics Rendition Store**
- **Retention Backup Store**
- **Staging Store**

#### 4.2.2.4 Editing a holding's confirmation settings

Even if a holding is in use, you can update the confirmation settings of the holding. The information presented here is the same information that was initially configured when the holding was created (refer to the **Confirmation** step of the holding wizard).

All of the information contained in the **Confirmations** and **Advanced Settings** tabs can be changed.

#### 4.2.2.5 Editing a holding's retention settings

Even if a holding is in use, you can update the following retention settings. For more information about these settings, refer to [Step 6: Retention](#).

When choosing the retention policy, if a retention policy is disabled, the system displays a **Disabled** message after the policy's name. Unless the policy is enabled, no disposition processing will be done for the package until the retention policy is enabled.

---

##### Retention Policy tab

For the **Retention Policy** tab, you can select how the retention will be applied during the ingestion.

If **Package** is selected, retention is applied to the entire package. The **Retention Policy applied by default** is displayed and you can add or remove any existing retention classes.

By default, the retention policy configured at the application-level will be applied. You can configure a default retention class to override this choice.

During the ingestion, the retention class to apply can be driven by the package description with the `sip > dss > retention_class` field. If the retention class is not listed below, ingestion will result in an error.

If **Granular** is selected, retention is applied to each record. For each PDI Schema, you can update:

|                         |                                                                                                                                                                                                                                                                                                         |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Retention Policy</b> | You are able to select a retention policy.                                                                                                                                                                                                                                                              |
| <b>AIU Query</b>        | Click <b>Template</b> to enter a query template in the field. Click <b>Clear</b> to remove data from the field. Once a template is added, update the data with the corresponding the <code>baseRetentionDate</code> path in the AIU Query field. Optionally, specify the <code>retentionPolicy</code> . |

---

##### Retention Store tab

You are able to select a store from the **Retention Data Store** list.

---

##### Disposition Policy tab

For the **Disposition Policy** tab, you can update the following fields:

- Structured Data:** If the package is being pruned (partially disposed), this setting indicates that if the PDI XML GZIP should be kept or not. If it is kept,

it is possible to access the original records that were disposed from the **Package** tab (but not via search).

- **Unstructured Contents:** If you update this field to **Shrink or Blank and Shrink**, you may want to adjust the **Shrink Ratio** setting. View the holding wizard instructions for details on the options.

If retention was applied to the package, and the store was configured for hardware retention, the PDI XML GZIP will always be kept independent of this setting and content will be kept, as no changes can be made to the content.

---

#### Retention Display tab

---

##### Search Name

You are able to enter a search name.

##### Search Set Name

You are able to enter a search set name.

Indicate if you want to use the first search template to display records when the above one is not defined or not authorized.

---

#### 4.2.2.6 Exporting a holding

Select **Export to ZIP file** to export the holding's configuration information.

### 4.3 Holding configuration details

#### 4.3.1 Library mode

When creating a holding, you can define the library mode.

Proper ingestion mode is the key aspect for optimal search performance. In general, the library mode depends on the SIP package characteristics and search use cases.

The holding defines the library mode. The possible values are:

- PRIVATE
- POOLED
- AGGREGATE

The PRIVATE mode is the recommended choice to archive SIP packages with huge number of AIUs (for example 100,000 or more). With this mode, we create one library per package to store structured data.

The POOLED mode has been introduced to increase the search performance when the partition keys are not enough to restrict the search scope. With this mode, after the close, the library to store structured data and the indexes are shared by all packages as part of the same pooled library.

The AGGREGATE mode has been designed for unitary deposit (1 SIP = 1 AIU). Unitary deposit allows a client application to synchronously archive data in OpenText Information Archive. The archived data can be immediately searched. To reduce the database footprint, all unitary packages are aggregated and only one aggregated package is persisted. This task is performed by the Close job. After the aggregation, the AIU IDs and CID remain unchanged. Smaller AIPs are aggregated into larger AIPs to use fewer partition keys. Therefore, retention is not applied until the aggregate is closed.

When the POOLED and AGGREGATE mode are selected, the holding is linked to a LibraryPolicy item to specify how the packages are assigned and when the library will be closed.

The effective close date is computed based on the information in LibraryPolicy item:

```
closeMode  
CREATION_DATE, LAST_MODIFIED_DATE, CLOSE_HINT_DATE  
  
closePeriod  
Number of days  
  
closeHintDateQuery  
XQuery to compute a DateTime based on the SIP XML information when the  
close mode selected is CLOSE_HINT_DATE.
```

The effective close date is computed with the following rules:

|                    |                                           |
|--------------------|-------------------------------------------|
| NONE               | MAX_DATE_TIME                             |
| LAST_MODIFIED_DATE | CLOSE PERIOD + LIBRARY LAST MODIFIED DATE |
| CREATION_DATE      | CLOSE PERIOD + LIBRARY CREATION_DATE      |
| CLOSE_HINT_DATE    | CLOSE PERIOD + XQUERY DATE TIME           |

A POOLED or AGGREGATE library can be closed only:

- When the effectiveCloseDate is expired.
- When the AIP/AIU quota are exceeded AND the property closeWithQuota is TRUE.
- If a manual closing has been requested.
  - For Pooled mode: AIP for which the close has been manually requested will have its library merged in the same pool that any other AIP eligible its close mode or also manually requested. All AIPs that are expected to be merged in the same pool when closing manually should have the manually requested action performed.

- For Aggregate mode: the action is available only at the aggregate-level. When called, the whole aggregate will be closed during the next run of the Close job.

OpenText Information Archive also allows you to dynamically select the library mode to use during 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 `LibraryPolicy.aiuThreshold` property).

When `LibraryPolicy.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.

### 4.3.2 Cache management

If needed, define a cache size at the application-level to limit the volume of structured data by storage system. This option is valuable to reduce the costs by not keeping all structured data libraries online. The default value is 0 and means unlimited. It is important to notice that it is not possible to perform a synchronous search if one or more packages have an offline library.

In addition to specifying a cache size, it is mandatory to schedule the Cache-Out job in **Administration > Jobs** tab of IA Web App. This job is responsible to automatically cache-out the libraries from the structured data stores to ensure the size stays within the size limit. If the job is not enabled and/or not run frequently, the cache size limit will be not honored. The cache-out algorithm is based on the search activity statistics. The oldest libraries accessed will be removed from the cache first.

A job parameter `maxLibraryPerApplication` is available to control the maximum number of libraries that are removed from the cache by application and by storage system for one job run. The default value is 1000.

If needed, manual cache-in/cache-out requests can be made from the **Packages** tab in IA Web App.

#### 4.3.2.1 Frequently asked questions about caching

---

##### How does system decide which AIPs and how many to cache-out?

The AIP selection is based on the read access statistics.

At the application level, you have a property `metadataCacheSize` to indicate how many metadata in bytes you want to keep in the metadata repository. By default, the value is 0 for unlimited.

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. This only occurs when the CacheOut job is run.

---

**How does system know that something is cached-out and it may have records from the search criteria?**

For each AIP, we have the information if the associated library is online or offline.

The search runs 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 libraries where we want to run the XQuery.

---

**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 hard limit, meaning the cache may exceed the values. The cache-out is performed by the Cache-Out job. If this job is never runs, 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 library is considered.

---

### 4.3.3 Structured data libraries management

The libraries management allows you to have more control on how the structured data are mapped to the data nodes.

It allows you to distribute the data between different databases to speed up search runs (using parallel mode) and ingestion times, as well as improve the high availability if a data node becomes unavailable.

Before choosing the library management mode to use, the Administrator must define the location of the libraries. According to the desired design, the Administrator must define root library objects that will correspond to one or multiple data nodes.

### 4.3.3.1 Modes

Three modes are provided:

---

#### Principal

This mode allows you to define one root library on one data node. This mode, however, does not allow the system to have a dynamic failover. If the data node becomes unavailable, ingestion will be transformed into a background request and will be performed when the data node becomes available again. This mode does not allow full performance during the parallel search because all searches will use the same data node. If nothing is set, this is the default mode.

---

#### FailOver

This mode requires you to define one primary root library on one data node and, optionally, another root library on a different data node. In this case, by default, all the libraries will be created and restored in the primary data node. If the data node becomes unavailable, the ingestion process checks the first additional data node available and uses it to perform ingestion.

---

#### Random

This mode requires you to define root libraries on multiple data nodes. In this mode, ingestion is performed randomly on the multiple data nodes. If a data node becomes unavailable, it does not affect the process, as the other data nodes are used. If all data nodes are unavailable, ingestion becomes a background request. This mode also allows for better performance during a parallel search because the searches are parallelized in multiple data nodes.

During the lifecycle of the libraries, some of them are detached. If a search or another process (such as confirmation) needs the library to be restored, the library can be restored using the precedent location or using the holding configuration. If the latter mode has been chosen, a library can be restored in a different data node if the random mode is selected or if the old data node is unavailable.

### 4.3.3.2 Load balancing

The use case of the load balancer is to allow the data nodes to be load balanced using a command after the configuration has been updated.

A typical example is if the data node is full, then the Administrator decides to add two new data nodes in order to scale. Doing so, the structured data is only in the full data node. The load balancer allows the Administrator to load balance the data in the available data nodes.

To run the load balancer, use the IA Shell command. Conversely, the REST API can be used using the “apply-library-policy” resource link on the application. For example:

```
http://localhost:8765/systemdata/applications/76f63329-b487-4ad3-92c4-a4e37f89eb14/apply-library-policy
```

The load balancing is considered from an application point of view, meaning if there are more applications using the same data node, only the data of the application being load balanced is considered.

For instance, consider the two following applications:

- Application 1 contains 30 GB of data on data node 1 and 60 GB of data on the data node datanode 2.
- Application 2 contains 100 GB on data node 1 and 1 GB on the data node 2.

With this configuration, when running the load balancer on the Application 1, data is moved from the data node 2 to data node 1.

If we consider Application 1, the average size of data by node is only computed by application. In this example, the average data node size is around  $(60+30)/2 = 45$  GB for Application 1. Consequently, the data is moved from data node 2 to data node 1, even if the total size (for all applications) of data node 1 is greater than the data node 2 (130 GB versus 61 GB).

#### 4.3.3.3 Locking the library in the cache

There is a specific option to indicate a minimal period in the cache.

A field `cacheLockPeriod` has been added to `LibraryPolicy` configuration object.

If set to a value above 0, which is the default value, the data libraries created following this policy will not be eligible to be cached out until its creation date + the number of days specified is reached. This applies even if the size of the cache is above the amount specified at the application level and there are no other candidates.

The cache lock period is displayed in the **Ingestion** tab of the Holding list page under the field **Cache Lock Period (days)**.

The cache lock date at the Library level (actual date until which the library is eligible to be cached out) is displayed in the **Library** tab of the **Package** list page under the field **Cache Lock Date**.

#### 4.3.4 Partition keys

#### 4.3.4.1 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 run. Partition keys are created during ingestion 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>).

**!** **Important**

Because of a PostgreSQL limitation, a partition key LIST<STRING> can only be used with the EQUAL and NOT EQUAL operators. Using operator other than EQUAL or NOT EQUAL results in the following error message: "A repeated partition key cannot be used with this operator: <OPERATOR\_NAME>."

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

- SIP-based searches will be sensitive to the amount of ingested data. 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.



#### Notes

- To significantly improve ingestion performance, all fields used as partitioning keys must also be defined as indexes.
- The partition key values list is not designed to store individual functional IDs of all archived records for an application. As there is a technical limit, increasing the number of entries will have a negative impact on ingestion and search time, and significantly enlarge the system database footprint and backup.

We recommend finding another partition key based on the sub-part of the functional ID and/or using a Min/Max partition key instead (valuable only if the records are sorted during SIP packaging).

#### 4.3.4.2 Types of partition keys

There are different types of partition keys:

| XSD Data Type | Partition Key Method | Maximum Number of Partition keys Allowed | Corresponding AIP Fields |
|---------------|----------------------|------------------------------------------|--------------------------|
| String        | Minimum or Maximum   | 4                                        | pkeys.stringXX           |
| Date/DateTime | Range                | 3                                        | pkeys.dateTimeRangeXX    |
| Int           | Range                | 2                                        | pkeys.integerRangeXX     |
| Integer       | Range                | 2                                        | pkeys.bigIntegerRangeXX  |
| Long          | Range                | 2                                        | pkeys.longRangeXX        |
| Double        | Range                | 2                                        | pkeys.doubleRangeXX      |
| Float         | Range                | 2                                        | pkeys.floatRangeXX       |
| Decimal       | Range                | 2                                        | pkeys.bigDecimalRangeXX  |
| Any data type | List                 | 4                                        | pkeys.valuesXX           |

#### 4.3.5 PDI indexes

Indexes are computed during ingestion and cover the structured data contained in one AIP. Indexes are used in the second tier of a search, whereby the system scans packages for individual records via the use of indexes.

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

The `Pdi.Search` lists criteria. Each criterion defines if a full-text index must be created to support full text operators.

➡ **Example 4-1: Example of PDI in YAML format (search part only):**

```
pdi:
  ...
  search:
    criteria:
      - name: SentToArchiveDate
        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}SentToArchiveDate
        type: DATE
        hint: true
        fulltext: false
        indexed: true
      - name: CustomerFirstName
        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}CustomerFirstName
type: STRING
fulltext: true
indexed: true
...

```



By default, the `fulltext` value is set to `false`.



**Note:** Lucene supports only values larger than 32 K for full-text indexes. Other index types will throw an exception. If values of type STRING are larger than 32 K, set `fulltext` to `true` and `indexed` to `false`.

Refer to the previous version of *OpenText Information Archive - Configuration Guide (EARCORE-CGD)* to learn about previous PDI format definitions with the `pdi.xml` file.

If PDI index's configuration is updated after the installation, and some AIPs have already been ingested, the libraries corresponding to those AIPs will be considered as out-of-sync by the system. Therefore, the user will not be able to launch synchronous searches targeting those AIPs. In this case, the system will propose to launch a background search instead. Doing so will re-index the library during the background search run. All out-of-sync libraries can also be re-synchronized using the Post Ingest Processing job. Refer to Section 3.7.18 "Post Ingest Processing job" in *OpenText Information Archive - Administration Guide (EARCORE-AGD)* to learn about the job's usage.

### 4.3.6 Stores

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

|                                   |                                                                                                                                                                                                             |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Reception Store</b>            | Indicates where the entire received SIP will be stored.                                                                                                                                                     |
| <b>Log Store</b>                  | Indicates where the log files will be stored.                                                                                                                                                               |
| <b>XML Store</b>                  | Indicates where the XML files will be stored (PDI XML, SIP XML).                                                                                                                                            |
| <b>Unstructured Content Store</b> | Indicates where the content information containers items and RI XML will be stored.                                                                                                                         |
| <b>Library Backup Store</b>       | Indicates where the sub-library will be stored.                                                                                                                                                             |
| <b>Analytics Rendition Store</b>  | Indicates where analytical rendition and purge list exports will be stored.                                                                                                                                 |
| <b>Retention Backup Store</b>     | Indicates where the retention data (managed items data) will be stored.                                                                                                                                     |
| <b>Staging Store</b>              | Indicates where the read/write operations prior to closing the aggregate (reception, ingestion, invalidation, etc.) will happen. This type of store is only available for holdings with the aggregate mode. |

If you plan on either applying retention or holds on records in an application, you must have a library store to store the information. This store can be shared with the

same store as structured data for a SIP application, although you may want to consider setting up a distinct store.

It is also recommended that you configure a regular store to store the backups of the retention information. If you chose to not do this, then OpenText Information Archive has no way of recovering from a disaster on your retention information.

The configuration of these stores is on the holding object for a SIP archive. For more information, see [Configuring a SIP application](#).

#### 4.3.6.1 About offline storage

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.

**!** **Important**

Avoid using a Glacier store with the aggregate library mode for a SIP holding configuration because it is NOT SUPPORTED.

**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.

**libraryBackupStore**

The backup of the libraries will potentially be offline. In this event, a cache-in of the library 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 backup 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.

#### 4.3.7 Handling unstructured content (CI)

The PDI format the CI section is defined directly in the CIS part of the PDI configuration:

```
pdi:  
  ...  
  cis:  
    - 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}Attachments/  
  {urn: eas-samples:en:xsd:phonecalls.1.0}Attachment/{urn: eas-samples:en:xsd:phonecalls.  
1.0}FileName  
      mimeType: audio/mpeg  
      compress: false  
      encrypt: false  
      offsetMode: SEGMENT  
      hashes:  
        - encoding: hex  
          algorithm: SHA3-224  
          enforced: false
```

As with the pdi.xml file, it is possible to define as many CI paths as you want, including zero, indicating no structured content for any records. For each CI path, the path property corresponds to the path containing the file name in the SIP ZIP package.

The `mimeType` property corresponds to the static value of the MIME type of the CI. If the value is not static, it is possible to specify a path with the property `mimeTypePath`. By default, the value is `application/octet-stream`.

**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. This mechanism allows the browser can open the file with the proper extension/plugin. The OpenText Information Archive viewer and OTIV viewer support many MIME types. The browser native viewer, however, depends on the browser being used and the plugins loaded on the browser. Therefore, it is important to ensure that the MIME type is correctly configured for each machine, if using the native viewer.

To view a list of supported document, image, and video formats:

- For OpenText Intelligent Viewing, click here (<https://www.opentext.com/assets/documents/en-US/pdf/opentext-intelligent-viewing-supported-formats-en.pdf>).
  - For OpenText™ Brava!™, click here ([https://www.opentext.com/file\\_source/OpenText/en\\_US/PDF/opentext-so-brava-enterprise-supported-formats-en.pdf](https://www.opentext.com/file_source/OpenText/en_US/PDF/opentext-so-brava-enterprise-supported-formats-en.pdf)).
  - For other browsers, refer to the specific browser documentation to learn about supported formats.

The `compress` and `encrypt` properties allow you to specify if the unstructured content must be compressed and/or encrypted before to be stacked in the CI container.

The hash properties allow you to compute a hash value during the ingestion, if the hash value is not provided or to validate a provided hash value defined with the hashPath property. The encoding and algorithm properties can be used to specify a static value for the encoding and the algorithm. If the values are not static, the properties encodingPath and/or algorithmPath can be used.

The offsetPath and lengthPath properties can be specified if one path corresponds to a segment (PDF or binary):

```
pdi:  
  ...  
  cis:  
    - path: /{urn:eas-samples:en:xsd:invoices.1.0}root/{urn:eas-  
      samples:en:xsd:invoices.1.0}  
        documents/{urn:eas-samples:en:xsd:invoices.1.0}content_file  
        mimeType: application/pdf  
        compress: false  
        encrypt: true  
        offsetPath: /{urn:eas-samples:en:xsd:invoices.1.0}root/{urn:eas-  
          samples:en:xsd:invoices.1.0}  
            documents/{urn:eas-samples:en:xsd:invoices.1.0}start_page  
            lengthPath: /{urn:eas-samples:en:xsd:invoices.1.0}root/{urn:eas-  
              samples:en:xsd:invoices.1.0}  
                documents/{urn:eas-samples:en:xsd:invoices.1.0}page_count  
                offsetMode: PAGE  
                hashes:  
                  - encoding: hex  
                    algorithm: SHA3-224  
                    enforced: false  
                  ...
```

### 4.3.8 Using the confirmation

A confirmation is a message generated in reaction to an AIP event (Reception, Storage, Available, Rejection, Invalidation, Disposition, Partial Disposition). The message is generated by an XQuery and the output format can be XML, CSV or TXT document.

The message can be attached to a confirmation audit or written at the root-level of a storage end point designated by a confirmation store.

The configuration is based on two items: Confirmation and Delivery Channel.

#### 4.3.8.1 Confirmation

**name**  
Confirmation name

**deliveryChannel**  
Delivery Channel to use

**deliveryChannelParameters**  
Parameters to pass to the delivery channel

**holdings**  
List of holding where the confirmation must be applied

**inputSchema**

List the pdiSchema of the confirmation (if none is provided, the confirmation will be applied to every PDI schema of this holding).

**types**

Event types to trigger. The allowed types are RECEIPT, AVAILABLE, STORAGE, REJECT, INVALID, PURGE.

**sipQuery**

XQuery to use to query the SIP XML

**aiuQuery**

XQuery to use to query the PDI XML.

**xmlRoot**

Boolean to indicate if the confirmation message must be enclosed with the confirmation root element with attributes for the type of confirmation, the date of confirmation and the AIP ID. The default value is TRUE. Only used with aiuQuery.

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

To perform a query on the PDI, 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. For Query on the SIP descriptor, the file is generated from the fields of the AIP object so it is always available.

The following event types can trigger a confirmation:

| Type      | Description                                                                  | SIP Query | AIU Query | Generated by     |
|-----------|------------------------------------------------------------------------------|-----------|-----------|------------------|
| Receipt   | The SIP has been received.                                                   | ✓         |           | Confirmation Job |
| Available | The AIP has been ingested and is available for search and content retrieval. | ✓         | ✓         | Confirmation Job |
| Storage   | The AIP is available and structured data has been successfully backed up.    | ✓         | ✓         | Confirmation Job |
| Reject    | The AIP has been rejected.                                                   | ✓         |           | Invalidation Job |
| Invalid   | The AIP has been invalidated.                                                | ✓         |           | Invalidation Job |

| Type  | Description                                      | SIP Query | AIU Query | Generated by    |
|-------|--------------------------------------------------|-----------|-----------|-----------------|
| Purge | The AIP has been disposed or partially disposed. | ✓         | ✓         | Disposition Job |

For the Reject and Invalid events, the system determines the audit based on what state the AIP was in.

If you want the confirmation to include some information about the AIU, it is necessary to specify an aiuQuery, which is typically used for the certificate of destruction proving that records were disposed.

As seen in the SIP query example, some external variables are available during SIP and AIU query runs and initialized by OpenText Information Archive with relevant information:

| Parameter           | Description                                                                                     | SIP Query | AIU Query |
|---------------------|-------------------------------------------------------------------------------------------------|-----------|-----------|
| aip_id              | ID of the AIP.                                                                                  | Available | Available |
| conf_type           | Confirmation type (for example, PURGE).                                                         | Available | Available |
| conf_datetime       | Date Time of the confirmation.                                                                  | Available | Available |
| sip_aiu_count       | Number of AIUs                                                                                  | Available | Available |
| ci_count            | Number of CIs                                                                                   | Available | Available |
| aggregate_aiu_seqno | Index of the first AIU ID. This number is equal to 0 when ingestion mode is not aggregate mode. | Available | Available |
| aggregate_ci_seqno  | Index of the first CI ID. This number is equal to 0 when ingestion mode is not aggregate mode.  | Available | Available |

| Parameter               | Description                                                                                                | SIP Query     | AIU Query |
|-------------------------|------------------------------------------------------------------------------------------------------------|---------------|-----------|
| number_of_disposed_aiu  | Number of AIUs from the AIP disposed during disposition. For other events, this number will be equal to 0. | Available     | Available |
| number_of_remaining_aiu | Number of AIUs remaining into the AIP.                                                                     | Available     | Available |
| row                     | Row of the current AIU submitted to the AIU query.                                                         | Not available | Available |

To use them in the XQuery, they must be declared in the following manner:

```
declare variable $aip_id as xs:string external;
declare variable $conf_type as xs:string external;
declare variable $conf_datetime as xs:dateTime external;
declare variable $sip_aiu_count as xs:long external;
declare variable $ci_count as xs:long external;
declare variable $number_of_disposed_aiu as xs:long external;
declare variable $number_of_remaining_aiu as xs:long external;
declare variable $aggregate_aiu_seqno as xs:long external;
declare variable $aggregate_ci_seqno as xs:long external;
declare variable $row as xs:long external;
```

If you need to include some properties of the AIP in the confirmation message, you can use the property-retriever function provided by OpenText Information Archive:

- Name : property-retriever
- Namespace: urn:x-emc:ia:functions
- Return: Depends on the type of the property in the AIP. Default is String.
- Parameter: One string with the property name (for example, sipAiuCount). If the property is not at the first level, the parameter should contains all the names in the property path separated by dots (for example, dss.producer).

Here is an example of an xQuery returning the producer of the AIPs in a attribute of each AIUs in the confirmation message for PhoneCalls application:

```
declare namespace n = "urn:eas-samples:en:xsd:phonecalls.1.0";
declare namespace ia-pdi = "urn:x-emc:ia:schema:pdi";
declare namespace ia-fun = "urn:x-emc:ia:functions";
<aiu id="@ia-pdi:id/string()" producer="{ia-fun:property-retriever('dss.producer')}>
  <CustomerID>{n:CustomerID/string()}</CustomerID>
  <CustomerFirstName>{n:CustomerFirstName/string()}</CustomerFirstName>
  <CustomerLastName>{n:CustomerLastName/string()}</CustomerLastName>
</aiu>
```

#### 4.3.8.2 Delivery channel

##### **name**

Name of the delivery channel.

##### **Store**

Destination store where to write the message. All storage systems are supported with the exception of Dell EMC CAS (Content Addressed Storage) Elastic Cloud Storage.

##### **fileName**

File name of the message. The value can contain variables like %key1% provided by the confirmation.

##### **subPath**

Relative path where to write the message. The value can contain variables like %key1% provided by the confirmation.

##### **prefix**

File name prefix. The value can contain variables like %key1% provided by the confirmation.

##### **suffix**

File name suffix. The value can contain variables like %key1% provided by the confirmation.

##### **overwrite**

Remove the message if it already exists. The default value is FALSE.

##### **compress**

Compress the file and add the zip extension. The default value is FALSE.

#### 4.3.9 Custom SIP format support

##### 4.3.9.1 Goals

By default, OpenText Information Archive is able to receive and to ingest a SIP corresponding to the SIP ZIP format. It is possible to accept custom SIP formats by adding a custom implementations to support them.

There are three public APIs to handle custom formats during the *reception*, the *ingestion* and the *content retrieval*.

### 4.3.9.2 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 an implementation based on the interface `SipReceptionHandler`

The contract is simple. From the SIP file, it's required 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 is 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 IA Shell

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
```

### 4.3.9.3 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 an 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 and to import external resources.
- Edit the YAML script

```
pdi:  
  name: Invoices-pdi  
  handlers:  
    - format: sip_custom  
      class: com.acme.ia.ingestion.SipCustomIngestionHandler  
      properties:  
        key1: value1
```

NB: The optional external resources are accessible on demand with the `context` object and the method `downloadConfigurationContent`.

### 4.3.9.4 Content retrieval

When the content needs to be transformed before to be returned to Emma, it's possible to ask OpenText Information Archive 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 an 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 is 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 instead.

### 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 is 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
```

## 4.4 AIC configuration details

To be able to configure a search later, it is necessary to define an AIC, a Query, and a ResultConfigurationHelper.

### 4.4.1 Configuring the AIC

The AIC defines the scope of the search, the list of the allowed search criteria and the list of partition keys to use.

**name**  
AIC name

**criteria**  
List of allowed criteria

**holdings**  
List of holding part of the view

**predicate**  
SpEL expression to restrict the view based on AIP metadata

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:

**name**  
Criterion name. The name must match the submitted search criteria.

**type**  
Criterion type. The allowed types are: STRING, INTEGER, BIGINTEGER, LONG, DOUBLE, FLOAT, BIGDECIMAL, DATE, DATETIME, ID and CID. The type is used to validate the submitted values

**pKeyMinAttr / pKeyMaxAttr**  
Indicates which, if any, min/max partition key can be used.

**pKeyValuesAttr**

Indicates which, if any, partition key list/simple can be used.

#### 4.4.1.1 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 run 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<br/>(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<br/>(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.).

## 4.4.2 Configuring the query

This object contains the mapping between the search criteria and the AIUs. The mapping must be configured for all PDI schemas potentially included in the AIC scope. An optional query template can be set to transform the output. A query can be associated to one or more AICs.

**name**  
Query name

**aics**  
AIC list

**namespaces**  
List of namespaces to declare

**resultSchema**  
Result schema name. This is an alias name to help to link with the result configuration helper.

**libraryPdiConfigs**  
Mapping configuration by PDI schema.

For each PDI configuration:

**schema**  
PDI Schema name

**entityPath**  
AIU path

**operands**  
List of operands

For each operand, it is necessary to specify the name, the path and the type:

**name**  
Operand name. The name must match the submitted search criteria.

**type**  
Operand type. The allowed types are: STRING, INTEGER, BIGINTEGER, LONG, DOUBLE, FLOAT, BIGDECIMAL, DATE, DATETIME, ID and CID. The type is used to validate the submitted values.

**path**  
Relative path

### 4.4.3 Configuring quota for search

Quota can be applied to the different steps of SIP search. These quotas are set by fields at the search composition level:

**name**

Quota name

**aipQueryQuota / aipQueryQuotaAsync**

Indicates the maximum number of AIPs a background search can span. A value of 0 indicates that there is no limit.

**aiuQueryQuota / aiuQueryQuotaAsync**

Indicates the maximum number of AIUs a background search can span. A value of 0 indicates that there is no limit.

**dipQueryQuota / dipQueryQuotaAsync**

Indicates the maximum number of AIU returns by a (background) search. A value of 0 indicates that there is no limit.

In case of a multithreaded search, this quota can still be exceeded.

If the `aipQueryQuota` or `aiuQueryQuota` are reached during the search runtime, the request returns error code `SEARCH_ERROR_EXCEEDED` associated to HTTP Status code 416 (Requested range not satisfiable).

If the `dipQueryQuota` is reached during the search runtime, the search stops and the result is returned with the results found before the quota was reached (with no guarantee on the order). The response contains a Boolean field `results.partial` that is then set to `True` to indicate to the client that the result is partial.

### 4.4.4 Configuring the ResultConfigurationHelper

The `ResultConfigurationHelper` item has been introduced to describe the XML structure returned by the query. The `ResultConfigurationHelper` will enumerate all elements, with the type and path. This information will be used during the result page composition.

**name**

ResultConfigurationHelper name.

**resultSchema**

List of result schema handled by this configuration.

**propagateChanges**

Update `ResultMaster` items with the new values (path and type) when the configuration is updated.

**content**

XML document with mapping configuration.

The XML document is composed with `<element/>` and `<group/>`. Each `<element/>` is composed with `<name/>`, `<label/>`, `<type/>`, and `<path/>`. An optional `<encrypt/>`

can be added to indicate that the value must be encrypted. A `<group/>` is composed with `<name/>`, `<label/>`, `<path/>` and with a nested `<element/>`. A `<group/>` must be used to handle repeated structures. All namespaces must be declared at the root level.

The allowed types are: STRING, INTEGER, BIGINTEGER, LONG, DOUBLE, FLOAT, BIGDECIMAL, DATE, DATETIME, ID and CID.

```

<resultConfigurationHelper
    xmlns:= "urn:ia-samples:en:xsd:sample.1.0"
    xmlns:ia= "urn:x-emc:ia:schema:pdi">

    <!-- 1 to n -->
    <element>
        <name>RepresentativeID</name>
        <label>Representative ID</label>
        <type>INTEGER</type>
        <path>n:RepresentativeID</path>
    </element>
    <!-- 0 to n -->
    <group>
        <name>Authors</name>
        <label>Authors</label>
        <path>n:author</path>
        <!-- 1 to n -->
        <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>
</resultConfigurationHelper>

```

 **Tip:** You can write the path (XQuery 3.0 based) you want. It can be useful to perform some value massage (string-join, duration computation, etc.).

The `id` field is a reserved field and is not to be used for customer data.

## 4.5 Enabling full-text on unstructured content in a SIP archive

You can configure an application to extract and index text of unstructured content during ingestion, search on them with full-text capabilities, and highlight them in the search results.

A ready to use example of this feature is available in the Invoices example application (refer to the "Search on unstructured contents" search).

A search whose full-text criterion contains only blank value is run as if the criterion had not been provided.

**To enable this feature for an application:**

1. In the **Global Settings** tab, ensure the `ingestion.ci.text.enabled` parameter is set to true (true is the default value). Refer to Section 3.12 “Global Settings” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)* for more information.

This step enables the full-text indexing for both SIP and table applications. To learn more about how the feature interacts with a table application, see [Enabling full-text for CI in a table application](#).

2. Complete the following steps to update application configuration:

- a. Edit the `application-config/application.yml` file to add the full-text option in the `libraryPolicy` section:

```
libraryPolicy:  
  name: Invoices-library-policy  
  libraryStores:  
    - default-library-store  
  aipQuota: 10  
  ...  
  ciText: true  
  ...
```

- b. Edit the `application-config/searches/application.yml` to declare the new CI criteria in the `aic` section:

```
aic:  
  name: Invoices-aic  
  criteria:  
  ...  
  - name: _ci-text  
    label: CI Text  
    type: ci  
  - name: _ci-mime-type  
    label: CI Mime-Type  
    type: ci
```

3. Import the updated DC configuration using one of the following methods:
  - The `import` command in the IA Shell, or
  - The **Import from ZIP file** function of the IA Web App’s **Applications** page
4. During search composition in the IA Web App:
  - a. In **Search Form** tab, add a new search criterion in the form.
  - b. In **Result List/Detail**, include a View/Downloadable content column.

It is possible to search on unstructured metadata fields, such as `_ci-mime-type` or `_ci-filename` with the operators:

- Exact Match
- Begins with (case sensitive)

The full-text search is performed by Lucene and the list of full-text operators is described on the Lucene Apache website.



**Note:** The current implementation has the following limitations:

- English locale is used by default to index and to perform the search.
- If the PDI configuration contains many CI paths, the full-text is enabled for all (all or nothing).
- It is not possible to display snippets into a dedicated column.

### 4.5.1 Language option

By default, the extraction process for unstructured content uses an English analyzer to process the index. It is possible to change the default language, if the contents are using another language. Changing the default language setting improves search result accuracy and has no additional cost in terms of performance or volume. For more information, see [Supported languages](#).

Note that indexing is based on Lucene, with the default analyzers defined by Lucene.

It is also possible to define multiple languages if a data set is composed of documents in different languages. Using multiple languages has a cost in terms of performance and disk space usage. Each time an additional language is specified, a new index is created for this language. Also, the search results use all the indices in the configuration, so it could have an impact. For instance, searching a stop word of a language (for example, the in English) will not be a stop word in other languages, thereby making the search results inaccurate.



**Tip:** Before using the multi-language option, verify if it is possible to have one application per language.

In any case, if the language setting is changed in the configuration after the ingestion, the Post Ingest Processing job must be run to re-index the data; otherwise, the desynchronization of the indices lead to inappropriate search results.

Language configuration can be done in an application's `application-config\configuration.yml` file:

```
application:
  name: Invoices
  primaryLanguage: FR
  additionalLanguages:
    - DE
    - EN
  archiveType: sip
  category: Example Application
  description: The SIP application archives customer invoices.
  offline: false
  type: active archiving
```

The `primaryLanguage` option is used to set the first language used for extraction (English is the default setting). Change this setting to set the language for extraction.

The additionalLanguages is used to set multiple languages. If the data set contains multiple language, set the primaryLanguage most used in the data set, and add the other used language in the additionalLanguages list option.

For instance, in the sample above, it is a multi-language configuration, the language most used is the French, and the additional languages are German and English.

## 4.5.2 Additional settings

Some options are available to modify the behavior of the extraction. It is possible to modify the libraryPolicy configuration object, in the application.yml file with the following options:

### ciText

**Type:** Boolean

**Default:** False

**Comments:** True /false, enables/disables the unstructured content extraction for the contents.

### ciTextMimeTypePatterns

**Type:** List<String>

**Default:** Null

**Comments:** Contains a list of regular expression patterns to include or exclude (accordingly with ciTextMimeTypeMode option).

### ciTextMimeTypeMode

**Type:** ENUM

**Default:** ALL

**Possible Values:**

- ALL
- INCLUDE
- EXCLUDE

**Comments:** Define the policy for the ciTextMimeTypePatterns list.

- ALL: All contents are extracted (ignores the ciTextMimeTypePatterns list).
- INCLUDE: Only the contents defined in the ciTextMimeTypePatterns list are taken into account.
- EXCLUDE: excludes the content with the mime types defined in the ciTextMimeTypePatterns.

### Field Names: List<CiTextPolicyEnum>/ CiTextPolicy

**Type:** ENUM

**Default:** BLANK\_ACCEPTED, ERROR\_ACCEPTED

**Possible Values:**

- BLANK\_ACCEPTED
- ERROR\_ACCEPTED

**Comments:** Defines the error policy applied during the unstructured content extraction.

- BLANK\_ACCEPTED: Allows the extraction to return a blank result.
- ERROR\_ACCEPTED: Allows the extraction to generate errors. In this case, the result of the extraction will be ignored and it will not be possible to retrieve unstructured content in the search.



## Chapter 5

# Ingesting data for a SIP application

After you have configured the holding for a SIP application, you can configure ETL tools or connectors for ingestion, and then generate SIPs and start ingestion.

For AIPs, retention and holds do not go into effect until the AIP is confirmed to be received. When ingesting, it is possible to auto-confirm this step.

The following sections of this guide contain relevant information about ingestion:

- The [Creating a holding with the holding wizard](#) section details the steps of configuring which library mode is applied to the holding,
- [Editing a holding's ingestion settings](#)
- The [Library mode](#) section describes the different library modes for a holding,
- The [Improving performance for a SIP archive](#) includes tips about improving ingestion speed,
- Ingestion is primarily done using IA Shell. *OpenText Information Archive - OpenText Information Archive Shell Guide (EARCORE-ARE)* contains information about the following:
  - Refer to the sections dedicated to the following commands related to ingestion:
    - enumerate command (Section 2.7.4 “enumerate” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*)
    - ingest command (Section 2.7.5 “ingest” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*)
    - receive command (Section 2.7.11 “receive” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*)
  - The configuration file properties that impact ingestion are documented in Section 1.5 “Configuration files” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*,
  - Details about ingesting SIPs asynchronously (Section 3.4.1 “Ingesting SIPs asynchronously” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*)
  - Section 3.4 “Scenario: How to ingest a SIP” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)* walks through the process of ingesting a SIP,
  - The sections Section 4.2.3 “Investigating the root cause of ingestion failure” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)* and Section 4.2 “Troubleshooting issues related to ingestion” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)* contain valuable troubleshooting tips,

- The Section 6.7.1 “Ingestion mode” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)* contains invaluable information related to ingestion for SIP archives.
  - Section 4 “Applications and data” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)* contains information about how data is ingested and stored.
  - Section 6 “SIP archiving fundamentals” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)* contains general information about SIPs, as well as information about:
    - Best practices when creating SIPs,
    - Descriptions for the elements of the SIP descriptor,
    - Data submission sessions (DSS) and external IDs,
    - PDI documents and PDI schemas,
    - The different ingestion modes of a holding,
    - Cache-in and cache-out mechanisms, and
    - The methods of adding a SIP to an archive and how to invalidate or reject data.

## 5.1 SIP ZIP format

The SIP .zip is the container that is submitted to OpenText Information Archive 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.



### Caution

All the file names must UTF-8-encoded.

For more information, see Section 6 “SIP archiving fundamentals” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.

## 5.1.1 SIP SDK

Building and ingesting a SIP is easiest accomplished using the Software Development Kit (SDK) for OpenText Information Archive available on the GitHub website.

## 5.1.2 Connectors

OpenText Information Archive's simple XML-based ingestion template and integration framework offers a simple, open integration interface.

As the OpenText Documentum CM Connector has been removed, the functionality has been merged and improved inside OpenText Documentum CM Content Server since 20.2. It provides a better integration and can be used on previous versions based on supported DFC versions.

OpenText Information Archive is ETL-agnostic and can be integrated with multiple choices of ETLs (market vendors, custom internal, P.S. Solutions, such as AppBuilder, OpenText Products, such as ContentBridge, OTS, Magellan Text Mining, OpenText Documentum CM or even partner connectors, such as Solex partners). The choice of integration can be REST API or file-based through IA Shell. Refer to the most recent Release Notes for additional information about the connectors.

OpenText Information Archive 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            |
| PeopleSoft                      | Unix           | XML databases  |
| Baan                            | Solaris        | ADABAS         |
| BASE T24                        | VMS            | Not applicable |
| ASG Mobius                      | LINUX          | Not applicable |
| ERPs                            | Not applicable | Not applicable |
| Financial Applications          | Not applicable | Not applicable |
| HR Systems (multiple)           | Not applicable | Not applicable |
| Core Banking Applications       | Not applicable | Not applicable |
| Customer Statement Applications | Not applicable | Not applicable |
| Healthcare Applications         | Not applicable | Not applicable |
| Life Sciences Applications      | Not applicable | Not applicable |

## 5.2 Delayed ingestion support

If the structured data node is missing during ingestion, an order can be created to complete the ingestion of SIP at a later time.

When running the `ingest` command, specify the `--allow-background-request` parameter. If specified, and the metadata database is not available during the ingestion, the ingestion becomes an asynchronous ingestion. For example, use the `ingest` command along with the `allow-background-request` option. For more information, see Section 2.7.5 “`ingest`” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*.

## 5.3 Receiving data and ingesting a SIP using the IA Web App

The following scenario demonstrates how to first receive data prior to ingesting it via the IA Web App. The advantage of initially using the `receive` command is that retention is not applied right away. You can also view the state of the data while it is being uploaded and, once received, verify that the data is correct (for example, the data has not been corrupted or it has not previously been ingested). The disadvantage of running the `receive` command prior to ingestion is that the data has only been uploaded to the IA Server. Users cannot run searches for this data.

1. Once connected to the IA Shell, run the following command from within the application the data will be ingested into:

```
iashell> receive --from data\PhoneCallsSample-2001.zip
```
2. Access the **Packages** tab for the selected application in the IA Web App. The **Phase** column displays the current phase of the AIP, which reads Waiting ingestion.
3. For the file that was received, click the link in the **Name** column. The **Package Content** page is displayed. This page allows you to check the receive log for any errors. You can also download the SIP descriptor file or the actual SIP ZIP file to ensure that the correct data was received.
4. Once you have verified the uploaded data is correct, click  to begin ingesting the data.
  - a. Drag and drop the packages to the space provided or click **Choose** to select the packages. A maximum of 10 packages can be ingested at once.
  - b. Click **Ingest**.  
If the **Ingest** button is disabled, the size of the packages exceeds the limit configured in the **Global Settings** tab. Update the `iawa.packages.ingestion.max.size` property accordingly or remove some of the packages.  
This procedure is completed asynchronously. You can navigate to the **Background Requests** tab to check the status of the procedure.

Once the data has been ingested, the **Phase** column in the **Packages** tab will read as Completed and the **Retention** column will indicate that the data is, in fact, under retention.

## 5.4 Limiting the size of files transferred through REST

The <IA\_ROOT>/config/iaserver/application.yml file allows you to limit the size of files that are transferred through REST during reception. Update the following section:

```
infoarchive:  
  rest:  
    maxRequestSize: 10MB  
    maxUploadSize: 2048MB  
    multipartFileSizeThreshold: 100KB  
    multipartLocation: ${localStorage.tempFolder}/multipart
```

The following explains the properties:

**maxRequestSize**

The maximum size of an inbound HTTP request (multipart excluded).

**maxUploadSize**

The maximum size permitted for uploaded file and multipart/form-data requests.

**multipartFileSizeThreshold**

The size threshold after which files will be written to disk.

**multipartLocation**

The directory where uploaded files will be stored.

Specify the values using long values or using more readable variants that accept KB, MB, or GB suffixes.



**Tip:** The `rest.maxUploadSize` default value can be increased. Prior to doing so, however, consider the architectural impact of the increase, particularly in terms of networking, monitoring, partitioning, and performance. For example, are the packages pure XML or do they contain files that will increase the package sizes in relation to the number of records/AIUs?



## Chapter 6

# Setting up searches for an application

Searching is the primary method that users use to access data ingested by OpenText Information Archive. To review conceptual information about searches, see Section 4.3 “How data is searched” in *OpenText Information Archive - Fundamentals Guide* (EARCORE-ACS).

Searches work for both SIP and table 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 searches require the name of the AIC and QueryConfiguration, which determine the criteria and results for the search.
- For a table archive, a search is associated with a schema or table. Table searches require an SQL query as well as the schema or table the search runs against. The Search Designer must understand SQL and how it is related to search design. The Search Designer needs to ensure:
  - The query is valid.
  - The parameters used are accurate.
  - The correct binding for elements is used in the result set.

For SIP archiving, OpenText Information Archive uses a two-tiered approach to searches. The following happens when a search runs:

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 search. For more information, see [Partition keys](#).
2. The system scans those packages for individual results (AIUs) via the use of indexes. For more information, see [Configuring indexes for a SIP archive](#).

All user roles can run searches except for the Administrator and IT Owner roles. For more information, see Section 7.3 “User roles and actions” in *OpenText Information Archive - Fundamentals Guide* (EARCORE-ACS).

The Search Designer can create a new search from scratch. While a search is being created, it remains in Draft status, which means only the Search Designer can see the search. Other user roles can only access a search when the status has been updated to Ready.

The Search Designer can:

- Duplicate a search set to create a new search without updating the original

- Export the search set to a .zip file and import the search into a similar application that has the same schema
- Configure for a search whether users can export the search

The End User role can only run searches against ingested data from a decommissioned application or active archive.

The Retention Manager can 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.

## 6.1 Differences based on application archive type

Search composition is slightly different depending on whether the Search Designer is creating a search for a table or SIP archive.

While composing a table search, the Search Designer uses a **Query Editor** tab to enter SQL code to perform the actual query against the data stored in the PostgreSQL database.

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 Search Designer can also use manual steps, for example, when creating a form. The Search Designer can use the Add Form element or use Add Column and Add Field in the Result List or Result Detail tabs.

## 6.2 Search CRUD

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

### 6.2.1 Search listing

Users need to access an application's **Search Forms** tab to view the available searches.

Each search may contain one or more search sets. The Search Designer can create different search sets, with each set having slightly different behavior in the actual search form and/or query and/or result. The Search Designer uses permissions to make each search set available to different groups.

When a search is in Draft mode, it is only visible to users with the Developer role.

Use the **Status** filter to filter the searches based on search status: Ready, Draft or All.

For table applications, the Search Designer can filter the list of searches based on the schema. If the application contains more than one schema, use the **Schema** filter to toggle between the different schemas.

For SIP applications, the Search Designer can filter the list of searches based on the AIC. If the application contains more than one AIC, use the **Archival Collections** filter to toggle between the different collections.

The Search Designer can complete the following actions from the **Search Forms** tab:

- Create a search or cross-application search
- Edit a search
- Set a search to ready
- Create a duplicate of a search.
- Export the search to a .zip file
- Import a search
- Configure the debug options for a search
- Delete a search

### 6.2.2 Configuring one-click display for a search form

If an application contains only one search form, the Developer can configure the form to display once the user accesses the application. This way, when the application is selected, the user does not have to click the search to access the form. Instead, the form instantly displays.

To enable this functionality, update the `enableOneClickSearchExecution` parameter in the `<IA_ROOT>/config/iawebapp/application.yml` file to `true` (the default value is `false`). The parameter can be found in the `webapp:` section of the `application.yml` file.

### 6.2.3 Creating a single application search

Only users in the Developer role can create new searches.

For a table archive, determine which tables and columns will be searched and displayed in the results. Searching table archives requires the creation of an SQL query.

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 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 side of the page indicates the column name and the data types of those columns. The 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 (or refer to [Creating a cross application search](#)):

1. Select the application and select + and select **Create Search**.
2. Enter the following information:

---

#### Search Name

Enter a name for the search that is unique within the application. The following special characters are not supported: '#', '<', '>', '\', '/', '!', '='.

---

#### Description

Enter a description for the search.

---

#### Categories

Select a previously created category or enter a new category. This information will help the Search Designer filter through the list of searches in an application.

---

#### Type

Indicate whether you are creating a:

- Primary search, which can be accessed directly by users.
- Nested search, which is a search to be linked from other searches to retrieve additional information on a result.

Most searches that you create will be primary searches.

---

#### Archival Collection

If you are creating a search set for a table archive:

1. Select an RDB database that was created for the application.
2. Select a schema the search form will access when run.

3. Optionally, if your query will only query a single table, select a table the search form will access when run.

If selected, the Search Designer is presented with a list of tables. Browse through the list or conduct a search to locate a specific table and click **Select**.



**Tip:** Specify the table if the search will be used to apply a hold to the results.

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.

### Configuration

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 form (for example, the list of fields that can be queried, the partition key, the indexed fields, the field format, etc.) and the result page (the list of result fields).

### Search Form

Select one of the following:

- **Composed:** 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 **Next**, you must complete the **Criteria** step of the search wizard. Go to step 3.

- **Custom:** You will use the XForms Editor to create the search form.

If selected for a SIP search, you must complete the **Results** step of the search wizard. Go to step 4.

If selected for a table search, click **Create** to start composing the search form with the XForms Editor. Refer to [Composing the search form using the XForms Editor](#) for further information.

---

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, which is only applicable to SIP searches:
  - 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 a search form using IA Web App](#).
- If **Custom** was selected, refer to [Composing a search form using the XForms Editor](#).

#### 6.2.4 Creating a cross-application search

Search designers can create searches that span different applications and can effectively:

- Specify which applications to include in the search, and which searches from those applications to be included in a cross-application search.
- Select which fields to use for mapping the specified applications. Search criteria mapping is performed on the IA Server side using XForm binding options.

IA Server runs the search in the context of an application. Each application's search result set is then returned to IA Web App.

By default, search results of a cross-application search appear in a tile view, with each tile corresponding to a delegate application and search. The tile view is

intended to provide a summary of search results, therefore only the first three columns and first five rows of the results appear in each tile.

To view complete results of a search, users can switch to a tab view by clicking a tile's header or the **Show All** link in the bottom-left corner of the tile. The columns and rows that appear in a tile or tab, as well as their order, are the same as would appear if the delegate search were run as a single, primary search (see [Creating a single application search](#) above).

In addition, the order of the search result tiles or tabs as well as their labels are determined by the configuration set in the cross-application search's **Result Order** tab (see discussion below).

A cross-application search can search the archives of multiple SIP and table applications. The searches included in a cross-application search are referred to as *delegate searches*.

Cross-application searches are not audited in the same manner as primary searches.

The icons displayed for each form in the **Search Forms** tab indicate whether a search is a primary search or a cross-application search:

|  |                                                                                      |
|--|--------------------------------------------------------------------------------------|
|  | Indicates a primary search.                                                          |
|  | Indicates a cross-application search.                                                |
|  | Indicates a delegate search that has been used to create a cross-application search. |

The following illustrates the delegate search (**DropdownlistFirstName**), as well as the cross-application search (**Name Cross-Application Search**) that links the searches.



Click one of the searches to access the respective search form. Click the pencil icon to edit the respective search.

See [Section 2.5.6 “Viewing cross-application search results”](#) in *OpenText Information Archive - End User Guide (EARCORE-UGD)* to learn how an end user interacts with the results yielded from a cross-application search.

When composing a cross-application search, there are three primary tabs on the search form: **Search Form**, **Result Order**, and **Permissions**. The **Search Form** tab is where the search form is composed. The **Result Order** tab is where the order and labels of the result tiles and tabs are configured. And the **Permissions** tab is where access to the cross-application search can optionally be restricted to one or more user groups. By default, all groups will be able to access the search.

When composing a cross-application search form, there is also a sub-tab that contains **Form** and **Mapping**. Once a form element has been added, use the **Mapping** tab to connect the element to a delegate search in the same or a different application.

A single form element can be associated with multiple delegate search fields. While in the **Mapping** tab, use the **Group By** list to view the different mappings by form field, delegate search form, or no group at all. No matter what the **Group By** list is set to, it does not impact how an end user views the search or its results. The **Group By** list simply helps the Search Designer navigate many different mapping items at once.

#### To create a cross-application search:

1. Access an application, click + and select **Create Cross-Application Search**.
2. Enter the following information:

|                    |                                                                                                                                                                     |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <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>Categories</b>  | Select a previously created category or enter a new category. This information will help the Search Designer filter through the list of searches in an application. |

3. Click a form element to add it to the search form.
4. Access the **Mapping** and enter the following:
  - a. **Form Field:** If you have added only one form element to the search form, it is pre-selected. If multiple form elements have been added to the search form, select the element being updated from the **Form Field** list.  
The values present in the list refer to the data binding values of the elements that have been added to the search form.
  - b. **Delegate Application:** The form element needs to be associated with a specific field in another search form. Select the delegate application that houses the desired search in the **Delegate Application** list.
  - c. **Delegate Search Form:** Select the delegate search form that contains the field being associated with the selected form element.
  - d. **Delegate Search Set:** Select the delegate search set that contains the specific field being associated with the selected form element.
  - e. **Delegate Search Field:** Select the delegate search field you want to associate with the selected form element. Any mandatory fields are marked with a red asterisk. If a checkmark appears beside a field, it indicates that this particular field has already been mapped.



**Note:** Only delegate search fields not marked as hidden can be selected.

- f. Complete one of the following:
    - To associate the same form element or a different form element with another delegate search field, click **+**. Repeat steps 4a to 4e.
    - To add another form element, repeat steps 3 to 4.
  - g. If you have completed the mapping process, click **Save**. Any incomplete mappings will not be saved.
5. In the **Result Order** tab:
    - a. Drag and drop rows to set the order of the delegate application and search tiles and tabs.
    - b. **Optional** Edit the **Tab Label** fields to change the labels on the result tiles and tabs.  
Click the **Preview** button to see how the search results from the delegate searches will appear to the search user.  
Select a tab to view that particular application's search results. Click the Information icon  to view:
      - The name of the selected delegate search.
      - The search set that was used from the delegate search.
      - The application that contains the delegate search.
  6. To restrict the cross-application search to specific groups, in the **Permissions** tab, select the **Grant access to specific groups** box. Select the groups that you want to have access to the cross-application search.  
When **Grant access to specific groups** is selected, at least one group has to be selected; otherwise, you will be unable to save changes to the cross-application search.  
Use the **Find a group** field to locate a specific group.  
The dropdown list beside the **Find a group** field can be changed to:
    - Show all groups
    - Show only groups with access
    - Show only groups without access
  7. Click **Save**.

## 6.2.5 Search importing and exporting from/to ZIP

The Developer can export a search:

1. Access the desired application.
2. For the search being exported, click the context menu and select **Export to ZIP file (YML)** or **Export to ZIP file (JSON)**.

The contents of the search are exported into a .zip file.



**Note:** When importing or exporting nested searches, you only need to import or export the primary search. The rest will be automatically included.

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 OpenText Information Archive and configure the imported search to suit your needs.

### To import a search from a ZIP file:

1. Access the application the search is being imported into and navigate to the **Search Forms** tab.
2. Click + and select **Import from ZIP file**.
3. Select the ZIP file being imported and click **Open**.

If the search name exists, a dialog prompts you to acknowledge whether you want to merge the imported search with the existing search.



### Tips

- If you intended to replace the search (because some search sets were removed), delete the existing search. The search cannot be deleted, however, if it is referenced by either a saved search, retained set, or hold set.
- If you wish to keep the existing search, it is better to rename the existing search first.

When importing a ZIP from the search page, the ZIP file must include exactly one search. You cannot import a configuration that is for more than one search.

The system does not automatically delete obsolete search compositions. The Developer must manually delete obsolete search compositions, which is only possible if the search is not in use.

The search name now appears in the list of searches.



**Note:** After importing a search, the search will be in Draft mode.

### 6.2.6 Copying a search

The Developer can 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. **Optional** Add 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.

### 6.2.7 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 **Edit** button for the search being edited.
3. Edit the search, as desired.
4. Click **Save**.

### 6.2.8 Deleting a search

The Developer role can 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**.

## 6.2.9 Search workflow

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 the **cog** icon.
2. On the **Edit Search** page, update the Status field to **Ready**.
3. Click **OK**.
4. Click **Save**.

## 6.3 Configuring search sets

This section illustrates how to view and create search sets.

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.

### 6.3.1 Creating a search set

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 run 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.

### 6.3.2 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 run a search.

The Developer can access all search sets by selecting a set from a drop-down in **Search Detail** tab and running the search.

To restrict the cross-application search to specific groups, in the **Permissions** tab, select the **Grant access to specific groups** box. Select the groups that you want to have access to the search.

When **Grant access to specific groups** is selected, at least one group has to be selected; otherwise, you will be unable to save changes to the search set.

Use the **Find a group** field to locate a specific group.

The dropdown list beside the **Find a group** field can be changed to:

- Show all groups
- Show only groups with access
- Show only groups without access

### 6.3.3 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.

### 6.3.4 Editing search set options

#### 6.3.4.1 Configuring a retention period for background searches

You are able to configure how long the system keeps background search results and exports. By default, order items are deleted by the Clean job after 7 days.

**To configure the retention period of a search set:**

1. Edit the applicable search and select the search set being configured.
2. Click the  icon for the search set and select **Edit Search Set**.
3. For the **Retention Period** field, enter the number of days the background search result will be available before being cleaned up by the system.
4. Click **Save**.

#### 6.3.4.2 Configuring a search result quota

A Search Designer can add a quota to a search set that limits the number of results returned in a search. A quota can be placed on either synchronous or asynchronous searches, or both. If a quota has been configured, a notification alerts end users that they are viewing partial results, as the search exceeded the maximum number of results.

**To configure a search result quota:**

1. Edit the applicable search.
2. For the applicable search set, click the cog icon > **Edit Search Set**.
3. Configure one or both of the fields in the **Search Quota** section:
  - **Search Result Quota:** Apply a quota to limit the number of records returned by the search.
  - **Search Result Quota for background searches:** Apply a quota to limit the number of records returned by the background search.
4. Click **Save**.

#### 6.3.4.3 Configuring a search to run in parallel

A SIP search set can be configured to run over multiple libraries in parallel. The search runs in two phases:

1. A list of AIPs that contains data to be returned is obtained.
2. The search is then split into sub-queries based on a limited number of libraries per sub-query. Offline libraries are evaluated one by one.

A search can be configured to run those sub-queries in parallel at the search composition level.

1. Edit the desired search form.

2. Click **Edit Search Set**.
3. Configure the search set, as required. The settings added here will take precedence over the default settings in the <IA\_ROOT>/config/iaserver/application.yml file. The default values for these settings can also be updated by changing the values in the application.yml file:
  - **Search Set Name**  
Indicates the name of the selected search set.
  - **Time out**  
Indicates the number of milliseconds before a synchronous search times out (default value is 8,000 milliseconds).
  - **Library Threshold**  
Indicates the maximum number of libraries included in the sub-query (default value is 100).
  - **Threads**  
Indicates the maximum number of threads per search available to run the sub-queries. If the value is less than 2, parallel execution is disabled. The default value is 1. The sub-queries are decided based on the Library Threshold, described above. If the number of sub-queries is less than the maximum number of threads, fewer threads will be used. When the number of sub-queries exceed the maximum number of threads, some threads will run some sub-queries in sequence.
  - **Background Priority**  
Indicates the default priority used to run the order item for the corresponding background search. The value must be set accordingly to the priority server settings. By default, the value is defined by the search.orderItemDefaultPriority parameter in the **Global Settings** tab.
4. Click **Save** to update the search set.

#### 6.3.4.4 Defining target languages at the search level

By default, a search on an archive containing unstructured content is done in all languages defined at the application-level. A search designer can, however, restrict the languages used to search on unstructured content. A use case for this feature is if you have invoices from different companies archived. Restrict the language of a search set to search the archive for invoices from a specific country by defining a certain language. For more information about defining languages at the application-level, see [Editing applications](#).

**To define target languages at the search level:**

1. For the desired search set, click the Settings icon  and select **Edit Search Set**.

2. In the Advanced Settings section of the Edit Search Set dialog box, click the **Inherited from application** box to remove the checkmark.
3. Select the languages you want to restrict the search set to. Language selection at search set level is for SIP applications only.
4. Click **Save**.

### 6.3.5 Renaming a search set

The Developer can change the name of a search set:

1. Click the **cog** icon and select **Set Name**.
2. Enter a new name for the search set and click **Save**.

### 6.3.6 Deleting a search set

The Developer can delete a search set:

1. Click the **cog** icon and select **Edit**.
2. For the set being deleted, click the **cog** icon and select **Delete Set**.
3. When prompted to confirm the deletion, click **DELETE**.

### 6.3.7 Duplicating a search set

The Developer can 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 the **cog** icon and select **Edit**.

The search sets are displayed on a panel on the right side of the page.

3. Click the **cog** icon and select **Create Duplicate**.

The new search set appears in the panel.

## 6.4 Composing a 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, the XForm data needs to match the SQL query external parameter names, which allow the custom XForm to run the search. SQL query external parameters reference the 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.

OpenText Information Archive'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>
```

```
</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

## 6.5 Composing a search form using IA Web App

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.

When creating or editing a search, you are able to toggle between running a search or editing the search:

 Run Search	Click to navigate to the search form and be able to run the selected search. If you click <b>Run Search</b> while editing a search, you will be prompted to save any changes before you are able to run the search.
 Edit Search	Click to be able to edit the selected search form. When you run the search, you are able to click <b>Edit Search</b> from the search results to continue editing the search.

### 6.5.1 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.

A pallet 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. The text element can also display HTML, including logos and images. The text element 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 using the following, which displays the content as an HTML document:

```
<html><body> ...</body></html>
```

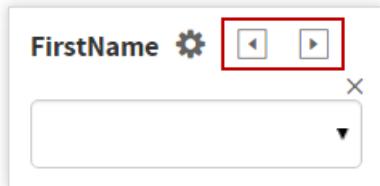
The following form controls are supported but are not listed in the control toolbar:

- 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

### 6.5.2 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.

### 6.5.3 Adding default values

#### 6.5.3.1 Adding a default value to a text field

The search designer can 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. Edit the `FirstName_Operator` search.
2. Click the **cog** icon 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**.

### 6.5.3.2 Adding default values 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. Edit the Certificate By Date Range search.
2. Click the **cog** icon to edit the settings for the Certificate Number field.
3. Enter the values in the **Default Value** field.
4. Click **OK**.

### 6.5.3.3 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**

### 6.5.3.4 Adding default date range values to a search form

The search designer can 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 the **cog** icon 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.

### 6.5.3.5 Adding default number values to a search form

The search designer can 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.

### 6.5.4 Searching for a value in multiple fields

The Developer can 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 the **cog** icon 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.



#### Notes

- 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.
- If the text in the data contains a comma, and the user wants to catch that using column filtering, the entire string should be contained in quotes.

Otherwise, the comma is used as a text delimiter and it is treated as two separate strings. For example, if the field is an address that contains a comma (for example, San Jose 95120, CA) and user searches for this string, the user should type “San Jose 95120, CA”. This way, the entire string is used in the match.

In the following, all first names or last names containing “Ja” will be matched and returned:

Operator for Employee Name	Customer First Name
Contains Exact Match Does Not Match Begins with (case sensitive) Begins with Ends with <b>Contains</b> Fulltext expression	<input type="text" value="Ja"/> <small>Search in First Name, Last Name</small>

### 6.5.5 Configuring an input element on a search form

An input element allows a search user to enter text, number and number ranges into a field during run-time. The following values can be customized in an input element:

---

#### UI Control

A read-only field that is set to Input for the search form criteria.

---

#### Control Type

Indicate if the search user must enter text, number or number range in run-time.

---

#### Field Label

Enter the name for the field that will be displayed in run-time.

---

#### Binding Type

One of the following values can be selected:

- **Single:** 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.
- **Composite:** 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.

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.

---

#### Data Binding

For table searches, enter the name of the variable in the query.

For SIP searches, select the value as a criterion.

---

**Aria Label**

Enter the string to be used as the accessible label for this particular object.

If Number Range is selected as the **Control Type**, enter aria labels for the From and To fields of the range.

---

**Composite Bindings**

If **Composite** is selected as a **Binding Type**, select binding from the list.

Essentially, you are selecting another field to bind the currently selected field to. Enter a label that will appear on the search form. Refer to [Searching for a value in multiple fields](#) for more information.

---

**Required**

Indicate whether an input value is required.

---

**Hidden**

Indicate whether this field should be hidden from the user on the search form. This allows the Search Designer to input values into a query, which the search user cannot change.

---

**Tooltip Text**

Enter concise, helpful information about the field that appears in a small “hover box” when the user hovers the cursor over the control.

---

**Prompt Text**

Enter text to help the search user enter specific criteria in the form (for example, “Enter a phone number”). The text a Search Designer enters here will be seen in the input box during run-time.

---

**Regex Pattern**

Enter the regular expression characters to specify a search pattern. Use the regular expression `^[\^s]+.*[\^s]+$[` to check for tail and trail spaces before submitting the form. If tail and trail spaces are found, an error will be displayed.

---

**Minimum Characters**

Define the minimum number of characters the user must enter to run a search. In run-time, the **Search** button is not activated unless the minimum number of characters are entered. This field is only displayed if **Text** is selected as the **Control Type**.

---

**Maximum Characters**

Define the maximum number of characters the user can enter to run a search. In run-time, the **Search** button is not activated if the user has entered more than the maximum number of characters. This field is only displayed if **Text** is selected as the **Control Type**.

---

**Allow multiple values**

Enter the values in the **Default Value** field. Use commas as a delimiter for multiple values. For example, enter Joe, Jane as default values.

**Default Value**

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.

If Number Range is selected as the **Control Type**, enter default values for the From and To fields of the range.

**Data Resolution**

Allows OpenText Information Archive to expand a single search value according to a configured value map. Then, all expanded values are used as search criteria to run search. Refer to [Configuring data resolution for supporting search value expansion](#) for further information.

### 6.5.5.1 Configuring number ranges for an input element

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).

### 6.5.6 Configuring key and value pairs for form elements

OpenText Information Archive allows you to use an editor to configure key and value pairs for the following form elements:

- Checkbox
- Radio Group
- Select/Drop-Down (single or multi-select drop-down lists)

The key and value editor allows you to enter multiple pairs in a table when you configure one of the form elements listed below. To enter another pair, click .

For the Select/Drop-Down element, the **Values** list must specify **Specified Manually** for the editor to be displayed. If the **Data Binding** field is set to date-related data, and the **Operator Selector** box is selected, the key and value editor is automatically populated with the following operators:

Key	Value
EQUAL	On
NOT_EQUAL	Not On
GREATER_OR_EQUAL	On or After
GREATER	After
LESS_OR_EQUAL	On or Before
LESS	Before

The list of operators depends on the data type and configuration (for example, if the full-text index is set).

The editor also allows you to select 0, 1, or multiple default values that appear in the form element during runtime. Multiple default values can be selected for the following:

- For checkbox elements, select the keys and values to be automatically selected in runtime.
- For drop-down lists, the **Selection Type** field must be set to **Multiple** before multiple default values can be configured.

For a radio group, only 0 or one default can be selected.

### 6.5.7 Configuring a checkbox element on a search form

A checkbox element allows a search user to select to select single values for submission in a form. Use the following values to configure a checkbox:

---

**UI Control**

A read-only field that is set to **Checkbox** for the search form criteria.

---

**Field Label**

Enter the name for the field shown on the form.

---

**Binding Type**

One of the following values can be selected:

- **Single:** 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.
- **Composite:** 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.

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.

**Data Binding**

For table searches, enter the name of the variable in the query.

For SIP searches, select the binding value as a criterion. The name value has to match in order to use the criterion for the SIP search.

**Aria Label**

Enter the string to be used as the accessible label for this particular object.

**Composite Bindings**

If **Composite** is selected as a **Binding Type**, select binding from the drop-down list. Essentially, you are selecting another field to bind the currently selected field to. Enter a label that will appear on the search form. Refer to [Searching for a value in multiple fields](#) for more information.

**Required**

Indicate whether an input value is required.

**Hidden**

Indicate whether this field should be hidden from the user on the search form. This allows the Search Designer to hard code values into a query, which the search user cannot change.

**Tooltip Text**

Enter concise, helpful information about the field that appears in a small “hover box” when the user hovers the cursor over the control.

**Checkboxes**

Use the editor to enter the keys and values that will appear for the form element. The editor also allows you to select 0, 1, or multiple default values. Refer to [Configuring key and value pairs for form elements](#) for more information.

**Data Resolution**

Allows OpenText Information Archive to expand a single search value according to a configured value map. Then, all expanded values are used as search criteria to run search. Refer to [Configuring data resolution for supporting search value expansion](#) for further information.

## 6.5.8 Configuring a radio group element on a search form

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:

**UI Control**

A read-only field that is set to Radio Group for the search form criteria.

**Field Label**

Enter the name for the field shown on the form.

**Binding Type**

One of the following values can be selected:

- **Single:** 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.
- **Composite:** 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.

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.

---

#### Data Binding

For table searches, enter the name of the variable in the query.

For SIP searches, define the binding value as a criterion. The name value has to match in order to use the criterion for the SIP search.

---

#### Aria Label

Enter the string to be used as the accessible label for this particular object.

---

#### Composite Bindings

If **Composite** is selected as a **Binding Type**, select binding from the drop-down list. Essentially, you are selecting another field to bind the currently selected field to. Enter a label that will appear on the search form. Refer to [Searching for a value in multiple fields](#) for more information.

---

#### Required

Indicate whether an input value is required.

---

#### Hidden

Indicate whether this field should be hidden from the user on the search form. This allows the Search Designer to hard code values into a query, which the search user cannot change.

---

#### Tooltip Text

Enter concise, helpful information about the field that appears in a small “hover box” when the user hovers the cursor over the control.

---

#### Radio Group

Use the editor to enter the keys and values that will appear for the form element. The editor also allows you to select 0 or 1 default value. Refer to [Configuring key and value pairs for form elements](#) for more information.

---

#### Data Resolution

Allows OpenText Information Archive to expand a single search value according to a configured value map. Then, all expanded values are used as search criteria to run search. Refer to [Configuring data resolution for supporting search value expansion](#) for further information.

---

## 6.5.9 Configuring the select element on a search form

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 select element:

---

### UI Control

A read-only field that is set to Select for the search form criteria.

---

### Control Type

Indicate if the search user will be presented with a list or a drop-down list.

---

### Selection Type

Indicate whether the user can select one or multiple values from the list.

---

### Field Label

Enter the name for the field that will be displayed in run-time.

---

### Binding Type

One of the following values can be selected:

- **Single:** 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.
- **Composite:** 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.

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.

- **Not a Search Criterion:** 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 **Selection Type** is set to **Single**.

---

### Data Binding

For table searches, enter the name of the variable in the query.

For SIP searches, define the binding value as a criterion. The name value has to match in order to use the criterion for the SIP search.

---

### Operator Selector

Check if the drop-down form field is to be bound to an operator. For more information, see Values in this table.

---

### Aria Label

Enter the string to be used as the accessible label for this particular object.

---

### Composite Bindings

If **Composite** is selected as a **Binding Type**, select binding from the drop-down list. Essentially, you are selecting another field to bind the currently selected

field to. Enter a label that will appear on the search form. Refer to [Searching for a value in multiple fields](#) for more information.

#### Required

Indicate whether an input value is required.

#### Hidden

Indicate whether this field should be hidden from the user on the search form. This allows the Search Designer to hard code values into a query, which the search user cannot change.

#### Tooltip Text

Enter concise, helpful information about the field that appears in a small “hover box” when the user hovers the cursor over the control.

#### Values

**Values:** Indicate the source of the values in the list:

- **Specified Manually:** Use the editor to enter the keys and values that will appear for the form element. The editor also allows you to select 0, 1, or multiple default values. Refer to [Configuring key and value pairs for form elements](#) for more information.
- **Derive from another list selection:**
  - Select an existing list to populate the values.
  - In the **Values** tab, create the mapping of the values from the existing list to the new list.
- **Derive from XQuery execution:**
  - Select an option in the **Query Source** 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 in the **Query** field. 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>Text label shown in web interface of item 1</label>
    <value>value_of_item_1_in_data</value>
  </item>
<!-- etc. --></data>
```

- Derive from another list selection + Query
  - For the **List Name** field, select an existing list to populate the values.
  - Select the **Query Source**.

- Enter the XQuery data in the **Query** field. 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 Resolution

Allows OpenText Information Archive to expand a single search value according to a configured value map. Then, all expanded values are used as search criteria to run a search. Refer to [Configuring data resolution for supporting search value expansion](#) for further information.

### 6.5.10 Configuring a date element on a search form

The date element allows you to enter a date, date range, a date and time, or a date and time range. The following values can be customized in a date element:

#### UI Control

A read-only field that is set to Date for the search form criteria.

#### Control Type

Indicate if the search user will be expected to enter a:

- Date
- Date Range
- Date Range (Composite)
- DateTime
- DateTime Range

#### Local Timezone

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 OpenText Information Archive repository, they can un-select this box to search the date-time values that do not include a time zone in the repository.

#### Field Label

Enter the name for the field that will be displayed in run-time.

#### Binding Type

One of the following values can be selected:

- **Single:** 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.
- **Composite:** 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.

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.

---

#### Data Binding

For table searches, enter the name of the variable in the query.

For SIP searches, select the binding value as a criterion. The name value has to match in order to use the criterion for the SIP search.

---

#### Aria Label for From/To Fields

Enter the string to be used as the accessible label for this particular object.

---

#### Composite Bindings

If **Composite** is selected as a **Binding Type**, select binding from the drop-down list. Essentially, you are selecting another field to bind the currently selected field to. Enter a label that will appear on the search form. Refer to [Searching for a value in multiple fields](#) for more information.

---

#### Required

Indicate whether an input value is required.

---

#### Hidden

Indicate whether this field should be hidden from the user on the search form. This allows the Search Designer to hard code values into a query, which the search user cannot change.

---

#### Tooltip Text

Enter concise, helpful information about the field that appears in a small “hover box” when the user hovers the cursor over the control.

---

#### Default Value for From/To Fields

A Search Designer can restrict a date range field so the end user is limited to entering a specified number of days, months, or years in the date range. For more information, refer to [Restricting a date range field](#).

---

#### Range Limit (for date and datetime range controls)

User can pick the number of days/months/range to limit the range entered by a user during runtime.

### 6.5.10.1 Adding date-time fields to a search form and results

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 OpenText Information Archive 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 the **cog** icon 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 OpenText Information Archive 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:
    1. 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.
    2. 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.
    3. Click **OK**.

9. If desired, in the **Result List** tab, click the **cog** icon 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, OpenText Information Archive 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)) };
```

### 6.5.10.2 Mapping a date range or datetime range control to two different columns

A date range or datetime range control can be mapped to multiple columns for the FROM and TO columns. In previous releases, the FROM and TO columns could only be mapped to one column each.

The date range (composite) and datetime range (composite) are only available for SIP-based searches and are not available for table-based searches or cross-application searches.

A likely scenario in which this functionality would be used would be for the PhoneCalls application. You are configuring a search that includes a date range control to search for phone calls that took place between a date range. However, a phone call takes place between the CallStartDate and the CallEndDate; therefore, you would need to map the Date Range to two different columns.

**To configure a date range or datetime range control to map to two different columns:**

1. When composing or editing the search form, for the **Control Type** field, select either:
  - **Date Range (Composite)**
  - **DateTime Range (Composite)**
2. How you proceed depends on whether the search is against SIP or table data. Essentially, for table searches, enter the name of the variable in the query. For SIP searches, however, define the binding value as a criterion.
  - For a search against SIP data, instead of selecting a single value from the **Data Binding** list, you must select:
    - **Data Binding: From**
    - **Data Binding: To**
  - For a search against table data, enter and select a value from the two **Data Binding** fields.:
    - 3. Click **Save**.

### 6.5.10.3 Restricting a date range field

During search composition, a Search Designer can restrict a date range field so the end user is limited to entering a specified number of days, months, or years in the date range. This ensures that the end user will not overwhelm the system searching for 200,000 records in a single search. If the user enters a date range that exceeds the specified limit, an error is issued and the end user cannot run the search. The range check applies to both date ranges and date-time ranges.

#### To restrict a date range field:

1. Edit a search that contains the date range field.
2. Select or enter an amount in the **Range Limit** field and select one of the following modifiers:
  - Days
  - Months
  - Years

For example, if you enter 2 and select Years as the modifier, the end user is blocked from entering a date range exceeding two years during runtime.

If the user enters a range exceeding the set limit, the **Search** button at run time or the OK button at composition time become disabled and an error message is displayed.

3. Click **OK**.
4. Click **Save**.

### 6.5.11 Configuring a text element on a search form

The text element allows the Search Designer to add information directly to a search form. The following values can be customized in a text element:

<b>UI Control</b>	A read-only field that is set to <b>Text</b> for the search form criteria.
<b>Text</b>	Enter the text that will appear on the search form during run-time.
<b>Aria Label</b>	Enter the string to be used as the accessible label for this particular object.

## 6.5.12 Configuring a container element on a search form

The following values can be customized in a container element:

<b>UI Control</b>	A read-only field that is set to Container for the search form criteria.
<b>Field Label</b>	Enter the name for the field shown on the form.
<b>Aria Label</b>	Enter the string to be used as the accessible label for this particular object.
<b>Background Color</b>	Select a background color for the container.
<b>Border Color</b>	Select a border color for the container.
<b>Conditionally Show</b>	<p>Allows you to configure how conditional elements will function in a search form. Refer to <a href="#">Using containers to create search forms that contain conditional fields</a> for more information.</p> <ol style="list-style-type: none"> <li>1. Select a field from the drop-down list.</li> <li>2. Select an operator from the drop-down list (for example, Exact Match or Does Not Match).</li> <li>3. Specify a value.</li> </ol> <p>Multiple conditions can be added to a container</p>

### 6.5.12.1 Using containers to create search forms that contain conditional fields

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 the **cog** icon to further refine the field.  
Only enter the following information. Otherwise, leave the field blank:

#### Control Type

Select the **Drop-down** option.

---

**Selection Type**

Select the **Single** option.

---

**Binding Type**

Select the **Single** option.

---

**Data Binding**

Select the **data5** option.

---

**Field Label**

Enter the name for the field shown on the form (for this example, it is **Direction**).

---

**Values**

Indicate the source of the values in the list. For this example, select **Specified Manually**. Use the editor to enter the keys and values that will appear for the form element. The editor also allows you to select the default value. Refer to [Configuring key and value pairs for form elements](#) for more information.

---

Click **OK**.

3. 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.

4. In the Container, click the **cog** icon to further refine the element.

Only enter the following information. Otherwise, leave the field blank:

---

**Field Label**

Enter **If Direction == Bank of Creditor**.

---

**Background Color**

Set the background color to **#BFDFF5**.

---

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 **BC**). 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 Text Field that you dragged into the container, click the **cog** icon to further refine the element.

Only enter the following information. Otherwise, leave the field blank:

---

**Type**

For this example, select **Text**.

---

**Binding Type**

For this example, select **Single**.

---

**Data Binding**

For this example, select **minos**.

---

**Field Label**

For this example, enter **Transport**.

---

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 the **cog** icon to further refine the element.

Only enter the following information. Otherwise, leave the field blank:

---

**Field Label**

Enter If Direction == 'Bank of Debitor'.

---

**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 the **cog** icon to further refine the element.

Only enter the following information. Otherwise, leave the field blank:

---

**Type**

For this example, select **Text**.

---

**Binding Type**

For this example, select **Single**.

---

**Data Binding**

For this example, select **vacation**.

---

**Field Label**

For this example, enter **Transport Batch Reference**.

Click **OK**.

In the second Text Field that you dragged into the container, click the **cog** icon to further refine the element.

Only enter the following information. Otherwise, leave the field blank:

---

**Type**

For this example, select **Text**.

---

**Binding Type**

For this example, select **Single**.

---

**Data Binding**

For this example, select **pain-ID**.

---

**Field Label**

For this example, enter **Transport Document Number**.

Click **OK**.

In the third Text Field that you dragged into the container, click the **cog** icon to further refine the element.

Only enter the following information. Otherwise, leave the field blank:

---

**Type**

For this example, select **Text**.

---

**Binding Type**

For this example, select **Single**.

---

**Data Binding**

For this example, select **pain-order**.

---

**Field Label**

For this example, enter **Transport Order Number**.

Click **OK**.

7. On the **Search Form** tab, click **SAVE**.

## 6.5.13 Value lists

### 6.5.13.1 Creating a value list

You can 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 they can be loaded by using [declarative configuration](#).

IA Shell can also be utilized for this task. For more information, see Section 3.18 "Value lists" in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*.

**If your application follows the same folder structure as the sample applications, you can add value lists to your application by performing the following steps:**

1. Create a directory with name `value-lists` in the `config` directory.
2. In that `value-lists` directory, create a `valueListName.xml` file with the value list document content.
3. Create a `configuration.yml` file that contains the name and description of the value list, and a reference to the value list document. See the following example of the Baseball application below:

```
valueLists:
- name: US_States
  content:
    format: xml
    resource: valueList-USStates.xml
  description: List of states in the USA
```

The resource element contains the reference to the Value List with name `valueList-USStates.xml`.

The `configuration.yml` file can contain multiple value lists.

To ensure that the value lists are imported with the other application resources, add the following line to the `includes` element value in the `config/configuration.yml` file:

```
includes:
- value-lists/configuration.yml
```

For more information about how to import a DC configuration, refer to [Import DC configuration](#) and Section 2.10.7 “import” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*.

When a full application that contains a `value-lists` directory is fully created, the value lists are loaded. OpenText Information Archive includes multiple examples of value lists. To see the examples, use IA Shell to navigate to the desired paths (for more information, see Section 3.18 “Value lists” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*).

### 6.5.13.2 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:

1. Store the values to be shown outside the search form.
2. Reuse the same value lists data in multiple forms.
3. 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.

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 running a query against a form.

### 6.5.13.3 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.

### 6.5.13.4 Samples

There is a variety of samples of value lists included in the sample applications:

<b>Order_Manage ment</b>	container-types
<b>Trades</b>	trader-names
<b>Audit</b>	event-types  system-event-types  tenant-event-types
<b>Baseball</b>	birth-years-selection  US_States

Several of the sample applications also use these value lists in their drop-down boxes.

The StockSectors and StateCities folders contain sample Spring Boot applications that can be built using Gradle.

To build a JAR file, run the following command within the StockSectors or StateCities folder:

```
.\gradlew clean assemble
```

- The stocksectors-datafile.properties file contains the stock sectors and their components key-values pair. For example:

```
Consumers=AAPL,NKE
Technology=CSCO,EMC,IBM,MSFT
Oil\ and\ Gas=WLL
Note: Escape space with backslash '\ '
```

The StockSectors.jar file is the runnable version of the project.

To run the web application, configure an environment variables STOCKSECTORS-DATAFILE point to thestocksectors-datafile.properties file (for example, STOCKSECTORS-DATAFILE="stocksectors-datafile.properties", then run the following command within the containing directory:

```
java -jar StockSectors.jar --server.port=2017
```

Alternatively, passing data properties file by system properties stocksectors-datafile.properties, run this command within the containing directory:

```
java -Dstocksectors-datafile="stocksectors-datafile.properties" -jar StockSectors.jar --server.port=2017
```

- The statecity-datafile.properties file contains the US states and their cities key-value pairs. For example:

```
Alabama=Birmingham,Dothan,Enterprise,Mobile,Tuscaloosa
California=Anaheim,Davis,Fresno,Castro\ Valley,Los\ Angeles,Oakland,Sacramento
Note: Escape space with backslash '\ '
```

The StateCities.jar file is the runnable version of the project.

To run the web application, configure an environment variables STATECITY-DATAFILE point to thestatecity-datafile.properties file (for example, STATECITY-DATAFILE="statecity-datafile.properties", then run the following command within the containing directory:

```
java -jar StateCities.jar --server.port=2017
```

Alternatively, passing data properties file by system properties statecity-datafile.properties, run the following command within the containing directory:

```
java -Dstatecity-datafile="statecity-datafile.properties" -jar StateCities.jar --server.port=2017
```

### 6.5.14 Configuring data resolution

### 6.5.14.1 Configuring data resolution for supporting search value expansion

Essentially, data resolution configuration allows OpenText Information Archive to expand a single search value according to a configured value map. Then, all expanded values are used as search criteria to run the 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 run 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, OpenText Information Archive can communicate with the external system via http request/response to get the resolved data.

### 6.5.14.2 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>
```

#### 6.5.14.3 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.

For XQuery of data resolution only supports external variable name \$input or the same name as the data binding.

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 runs 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 runs the search.

#### 6.5.14.4 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.

We provide an example of how to do this as one of the rest-services examples. The source code is provided in the <IA\_ROOT>/examples/rest-services/Trades directory. You will require Java11 in the classpath to build the library and you will need network access to download gradle.

```
cd <IA_ROOT>/examples/rest-services/Trades/StockSectors  
gradlew assemble
```

Using following command to start the sample external system:

```
cd <IA_ROOT>/examples/rest-services/Trades/StockSectors/build/libs  
java -jar StockSectors-20.2.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 a the `-Dstocksectors-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 comment:

```
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.

#### 6.5.14.5 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.

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 runs the search, the search will return results that ticker equates with one of the values 'CSCO', 'IBM' and 'MSFT'.

#### 6.5.14.6 Unsupported form controls

The following form controls do not support data resolution:

- Date range
- Number range
- Operator selector
- Text
- Group

#### 6.5.14.7 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>

#### 6.5.15 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 run.

If using images or logos, the files have to be pre-loaded in the following location:  
<IA\_ROOT>\first-time-setup\applications\Tenant\config\customization\branding\images. The HTML in the following example was tagged accordingly:

```
<html><body></body></html>
```

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
YearMonth	NMTOKEN
gYear	NMTOKENS
gMonthDay	integer
gMonth	nonPositiveInteger
stringboolean	negativeInteger
base64Binary	long
hexBinaryfloat	int
decimal	short
doubleanyURI	byte
QName	nonNegativeInteger
normalizedString token	unsignedLongunsignedInt
language	unsignedShort
Name	unsignedBytepositiveInteger
	listItem
	listItems
	dayTimeDuration

## 6.6 Composing the search result

After composing the search form, the Developer can create a result list by adding and configuring columns in the Result List.

### 6.6.1 Adding columns

When composing the result list for a table archive, you can add columns and bind them to query element names.

When composing the result list for a SIP archive, the following options are available:

- Manually adding columns to the search results.
- Select columns from a schema to add to the result list by clicking **Select from** and selecting **Schema**.
- Select columns from a package to add to the result list by clicking **Select from** and selecting **Package**. Any custom data that has been added to the package appears in the **Column name** column as <field>. For example, if a field was named `CustomerInfo`, it will be displayed as `metadata.CustomerInfo`.
- Select compliance columns to add to the result list by clicking **Select from** and selecting **Compliance**. The following table lists the possible compliance columns:

Column Name	Column Label	Type	Description
createdDate	Compliance Applied Date	datetime	The date when compliance was applied to the record.

Column Name	Column Label	Type	Description
directlyUnderHold	Directly Under Hold	string	Indicates if a hold was applied to the record directly. This includes if a hold was applied to a saved search or if the saved search containing the record was added to a legal matter.
directlyUnderRetention	Directly Under Retention	string	Indicates if a hold was applied to the record directly.
holdNames	Directly Under Retention	string	Indicates the unique set of names of holds that are applied. The holds could be applied directly to the record or inherited from the package, table, or application. If there is more than one, the list is separated by commas.
projectedDispositionDate	Projected Disposition Date	datetime	Projected disposition date of the record.
retentionPolicyNames	Retention Policy Names	string	Indicates the unique set of names of retention policies that are applied. The retention policies could be applied directly to the record or inherited from the package, table, or application. If there is more than one, the list is separated by commas.
underHold	Under Hold	string	Indicates if the record is under hold (either a direct application or inherited).
underRetention	Under Retention	string	Indicates if the record is under retention (either a direct application or inherited).



**Note:** This is also the list that is used for exporting compliance information when exporting a search result. For more information, see [Exporting compliance information](#).

## 6.6.2 Moving columns and fields in a set of search results

The Search Designer can move columns and fields from a set of search results. For example, if there are too many columns returned in the result list, the Search Designer can move one or more columns to the result details section of the search results.

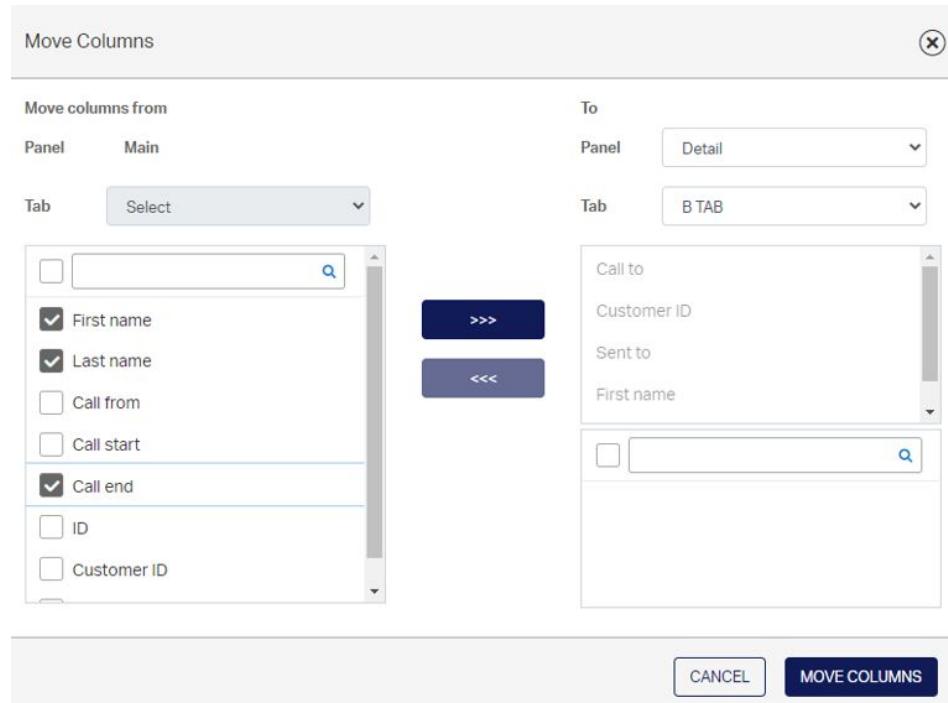
Essentially, the functionality cuts a column or field from the source panel or tab and pastes it into the target panel or tab. The Search Designer can move columns and fields from one panel or tab to any detail tab, inline or result list within the search form.

In the following scenario, a Search Designer is updating the Name Search results for the Tickets example application. The following columns specified in the Result List tab will be moved to the Result Detail tab:

- Member Since
- Balance
- Last Activity

1. While in the **Search Forms** tab of the Tickets application, edit the Name Search and navigate to the **Result List** tab.
2. Click **Move Columns**.

The **Move Columns** dialog box is displayed. The left side of the dialog represents the source panel, in this case the main result details grid. On the destination (target side), there are two list panels. The top one shows the current columns or fields in the target panel/tab. Whatever you choose to move appears in the bottom panel.



Since there are no tabs in this instance, the **Tab** dropdown list is inactive.

Each column is indicated in the box below. A search field allows you to search for a specific column. Moreover, select the box to the left of the search field to select all the columns.

3. For the purposes of this scenario, however, select:
  - Member Since
  - Balance
  - Last Activity
4. Since the selected columns are to be moved to the **Result Detail** tab, set the Panel dropdown list to **Detail**.  
Since there are no tabs in this instance, the **Tab** dropdown list is inactive.
5. Click the **>>>** button to bring the source columns to the selected target.  
The columns are now displayed in the target side of the dialog.  
The <<< button is to move any items back to the source panel. You cannot move items originally in the target panel to the source panel. For example, in this scenario, the fields Member Since, Balance and Activity are part of the target Result Detail panel. These fields cannot be brought over to the source panel during this transaction. To move these fields, you would have to click Move Fields on the Result Detail tab. Then, these fields would become the source panel (see the second procedure in this section for more information).

6. Click **Move Columns** to complete the change.

The system navigates you to the tab which was the target panel in the dialog, in this case the Result Detail tab. The columns selected in the previous step now appear in this tab.

7. Click **Save**.

The following scenario reverses the changes completed in the previous scenario (in other words, moving those same three columns back to the main search result panel).

Since the columns were moved to the **Result Detail** tab, the procedure starts there.

1. Click **Move Fields**.

The **Move Fields** dialog box is displayed. The left side of the dialog represents the source panel, in this case the **Result Detail** grid.

Since there are no tabs in this instance, the **Tab** dropdown list is inactive.

Each column is indicated in the box below. A search field allows you to search for a specific column. Moreover, select the box to the left of the search field to select all the columns.

2. For the purposes of this scenario, however, select:

- Member Since
- Balance
- Last Activity

The right side of the dialog represents the target panel.

3. Since the selected columns are to be moved to the main result details, set the **Panel** dropdown list to **Main**.

Since there are no tabs in this instance, the **Tab** dropdown list is inactive.

4. Click the **>>>** button to bring the source columns to the selected target.

The columns are now displayed in the target side of the dialog.

5. Click **Move Columns** to complete the change.

The system navigates you to the tab which was the target panel in the dialog, in this case the Result List tab. The columns selected in the previous step now appear in this tab.

6. Click **Save**.

### 6.6.3 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, the Brava! viewer, or the Intelligent Viewing viewer.

### 6.6.4 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 native browser, the Brava! viewer, or the Intelligent Viewing viewer.

To configure a normal column:

1. Click the **cog** icon to configure the desired column.
2. Enter the following information:

---

#### Column Label

Enter the label of the column, which will appear as the header in the result details.

---

#### Column Type

For a table archive, select **Query Reference**.

For a SIP archive, select **Schema Column Name**.

---

#### Column Name

For a table search, enter a name for the column defined in the row. Select the Binding.

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 Query configuration resource.

---

#### Enable Filter

- a. Select if you want to allow the user to filter the column's results or use the advanced filtering feature.
- b. 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.

---

**Sensitive Information**

Click to ensure that, when the user runs 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 binding is set to a field that is known to be encrypted, the field cannot be changed.

---

**Masked**

For a table archive, indicate if the column is masked in the query.

---

**Include in Export**

If selected, allows the user of the search to export the search results of the selected column.

---

**Sort**

Indicate whether:

- Sort will be disabled for the column.
- Sort will be enabled for the column.
- The column is to be displayed as a default sort.

The sort order column is hidden if the sensitive information is set.

---

**Group By**

Select to group the search results by the selected column in runtime. You have the option to group the search results for the column in ascending or descending order, or by the record. The Group By feature can only be selected for one column in a Result List.

---

**Data Type**

Select the data type for the column.

---

**Disable Thousands Separator**

Removes the thousands separator when displaying a number. For example, in English, the thousands separator is a comma: 12,345.

---

**Format**

Select or enter a date and time format.

The value of this field allows you to control the format of dates and numbers, based on a user's locale information. This field ensures that the Search Designer does not have to duplicate an entire search configuration simply to push a particular date/number format for different locales (for example, 100,000 for a North American user versus 100 000 for a user in France). For example, a DATETIME result may contain too much information and the Search Designer wants to only have the year and month displayed, the result can be configured to only show the year, month, hour in a 24-hour format, and the minutes (no seconds or milliseconds have to be included in the result). Refer to the rules regarding Angular DatePipe formatting for more information.

---

---

#### Hide Column

Indicate if you want the result column to be hidden from the user.

---

#### Binding

Binds a search to a specific table. The list of values for the binding also indicates if the field is encrypted.

It is possible to clear the binding by choosing **Select Binding** from the menu.

---

3. Click **OK**.

### 6.6.5 Disabling local time zone in search results

By default, dates and times in **Date** and **DateTime** fields in search results are converted to the user browser's local time zone before being displayed.

If the **Do not convert to browser local timezone** box is enabled, the date and time values are instead displayed as returned by the IA Server:

- For a SIP application search, the date or time is returned as originally ingested.
- For a table application search, the date or time is returned in the IA Server's time zone due to implicit intermediate PostgreSQL time zone conversion to UTC.

#### To disable local time zone for a Date or DateTime field:

1. Click the **Result List** tab in the search composition.
2. Click the **cog** icon in the field's column.
3. In the **Edit Column** dialog, select the **Do not convert to browser local timezone** box.



**Note:** The checkbox appears only when the **Data Type** is **Date** or **DateTime**.

4. Click **OK**.
5. Click **Save** to save the search composition.

## 6.6.6 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 can 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 the  icon 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, there are fewer possible filter operators. Normally, only exact matches are supported.

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, the search is now considered to be in draft and cannot be used by end users. Reset the search **Status** to **Ready** to make it available for general use.
7. Click **Save**.

### 6.6.7 Adding an external link to search results

The search developer can 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 the **cog** icon 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.

---

#### URL Protocol

Select whether the URL protocol identifier will be **Static** or **Dynamic**:

- If **Static** is selected, select either **http** or **https**.
- If **Dynamic** is selected:
  - For a table archive, enter either HTTP or HTTPS in the field.

- For a SIP archive, select the column from the list that contains the applicable URL protocol identifier.

---

**Host**

Select whether the host will be **Static** or **Dynamic**:

- If **Static** is selected, enter the name of the host in the field provided.
- If **Dynamic** is selected:
  - For a table archive, enter the name of the column that contains the applicable host.
  - For a SIP archive, select the column that contains the applicable host.

---

**Port**

Select one of the following to indicate the port number to which to connect:

- **None**: There is no port number.
- **Static**: Enter the port number to which to connect.
- **Dynamic**:
  - For a table archive, enter the name of the column that contains the applicable port number.
  - For a SIP archive, select the column that contains the applicable port number.

---

**Path**

Select one of the following to indicate the path:

- **None**: There is no path.
- **Static**: Enter the path to which to connect.
- **Dynamic**:
  - For a table archive, enter the name of the column that contains the applicable path.
  - For a SIP archive, select the column that contains the applicable path.

---

**URL Target**

Indicate if you want the target to open in a **New Browser** tab or a **New Window**.

- 
4. In the External Parameters Mapping section:

- a. Click +.
- b. 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.

- c. 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**.

To configure a column for a nested search:

1. Click the **cog** icon to configure the desired column.
2. Enter the following information:

---

**Column Label**

Enter the label of the column, which will appear as the header in the result details.

**Column Type**

Select **Linked Column (Nested Search)**.

---

**Column Name**

Enter one of the following:

- For a table search, enter a name for the column defined in the row.
  - 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 Query configuration resource.
- 

In the Nested Search Mapping section:

- a. Click +. You can map as many result columns to search fields, as required.
- b. 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.
- c. Click **OK**.

Alternately, you can also:

1. Click the **cog** icon to configure the desired column.
2. Access the **Linked Column** tab.
3. In the **Linked column as** radio group, select **Nested Search**.
4. 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:

- a. Click +. You can map as many result columns to search fields, as required.
- b. 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.
- c. Click **OK**.

### 6.6.8 Extending a nested search to another application

Previously, nested searches were originally created to provide additional information for a record using the context as input for the second search. The functionality has been expanded (now called “linked searches”) to allow the Search Designer to configure a column to trigger a search in another application, potentially linking two different sets of data, if they share common metadata that can be used as search criteria. The linked search can be any search from the same or different application.

Furthermore, nested/linked searches can now be run in the background (asynchronously).

For example, contract and invoice data are stored in different applications, but each archived invoice contains a contract ID. The results of an invoice search can provide a linked search to the data stored in the Contract application. Essentially, the user can jump to another application to search its data. In fact, the user can even run the secondary search in the background.

In previous releases, nested or linked searches running in the background (asynchronously) would time-out after 30 minutes. Currently, once a nested or linked search runs synchronously, the system will verify if a timeout value has been configured for the search in the Global Settings. The server-side timeout determines when the search becomes a background search.

Furthermore, you can also configure the time limit by editing a search set and updating the **Time out** field. The value of this field indicates the number of milliseconds before a synchronous search times out.

An application must be online if it is to use this functionality.

#### To extend a nested or linked search to another application:

1. Whether you are creating a new search or editing an existing one, navigate to the **Result List** tab.
2. For the column being updated, click the cog to edit the column and, in the **Edit Column** pop-up, click **Linked Column**.
3. For the **Link Column as** field, select **Linked Search**.
4. For the **Application** field, select the application that the search will link to.

5. In the **Linked Search Form** list, select the form that the linked search will use in the application selected during the previous step.
6. If desired, complete the **Linked Search Mapping** section, which includes:
  - Mapping options from the result list and the result detail in the **Value From Search Result** list.
  - Mapping options from the linked search in the **Linked Search Field** list.  
Click + to add further mapping details.
7. Click **OK**.
8. Click **Save**.

Remember to change the status of the search to Ready. Otherwise, users will not be able to access it.

### 6.6.9 Configuring a result column to allow downloadable unstructured content

You can configure a search result column so that a user can download unstructured content at runtime. For example, a user could download the MP3 audio file of a customer phone call.

#### To configure a result column to allow downloadable unstructured content:

1. Click the **cog** icon to configure the desired column.
2. Enter the following information:

---

##### Column Label

Enter the label of the column, which will appear as the header in the result details.

---

##### Column Type

Select **Downloadable Content**.

---

##### Column Name

Enter one of the following:

- For a table search, enter a name for the column defined in the row.
- 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 Query configuration resource.

---

##### Content Link Display

Select one of the following:

- **Custom Label:** 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.

- **From a column:** The value entered depends on the type of application:
  - For a SIP application, select the column
  - For a table application, enter a column

For example, if a Customer ID is picked when the Content link display is **From a column**, during run time, the user sees the value that is returned for the customer ID, such as 000453.
- **Download Icon:** Select to display the download icon in the search results during runtime.

---

#### **Enable Filter**

- a. Select if you want to allow the user to filter the column's results or use the advanced filtering feature.
- b. 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.

---

#### **Include in Export**

If selected, allows the user of the search to export the search results of the selected column.

3. Click **OK**.

### **6.6.10 Configuring a result column for downloadable content**

You can configure a column so that it allows the user to view content in runtime.

1. Click the  icon to configure the desired column.
2. Enter the following information:

---

#### **Column Label**

Enter the label of the column, which will appear as the header in the result details.

---

#### **Column Type**

Select **Downloadable Content**.

---

#### **Column Name**

Enter one of the following:

- For a table search, enter a name for the column defined in the row.
- For a SIP search, select the column name from the list. The column name is the name of the field from the schema definition (.xsd).

**Custom**

For a search form in a SIP application, select to enter a custom column path in the field provided.

**Tooltip Text**

Enter concise, helpful information about the field that appears in a small “hover box” when the user hovers the cursor over the control.

**Include in Export**

If selected, allows the user of the search to export the search results of the selected column.

**Label / Show Icon**

**Label:** Enter a customized label for each table cell for the View Content column.

**Show Icon:** Select this box if you would rather have an icon displayed as the View Content header.

**Hide Column**

Indicate if you want the result column to be hidden from the user.

3. Click OK.

### 6.6.10.1 OpenText Information Archive Viewer

The OpenText Information Archive Viewer supports the following content types: PDF, BMP, TIFF, PNG, JPEG, and GIF. Ensure that the appropriate MIME Type is selected for the content.

### 6.6.10.2 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
Chrome	Yes	No	No	Yes	Yes	Yes	Yes
Edge	Yes	No	No	No	No	No	No

## 6.6.11 Configuring a result column when the Column Type is View Content or Viewable Export

**Complete the following procedure when the Column Type is set to View Content:**

1. Click the  icon to configure the desired column.
2. Enter the following information:

---

### Column Label

Enter the label of the column, which will appear as the header in the result details.

---

### Column Type

Select **View Content**.

---

### Column Name

Enter one of the following:

- For a table search, enter a name for the column defined in the row.
- For a SIP search, select the column name from the list. The column name is the name of the field from the schema definition (.xsd).

---

### Custom

For a search form in a SIP application, select to enter a custom column path in the field provided.

---

### Content Link Display

Select one of the following:

- **Custom Label:** 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.
- **From a column:** The value entered depends on the type of application:
  - For a SIP application, select the column
  - For a table application, enter a column

For example, if a Customer ID is picked when the Content link display is **From a column**, during run time, the user sees the value that is returned for the customer ID, such as 000453.
- **Download Icon:** Select to display the default download icon in the search results during runtime.

---

### Tooltip Text

Enter concise, helpful information about the field that appears in a small “hover box” when the user hovers the cursor over the control.

### Viewer Type

Indicate the viewer that will be used to view the information, either Browser Native or OpenText Information Archive Viewer.



**Note:** Depending on your configuration, a third option may be available: OpenText Intelligent Viewing. For help on enabling this third option, refer to [OpenText Intelligent Viewing integration](#).

### Viewer Options

Indicate the viewer that will be used to view the information, either Browser Native or OpenText Information Archive Viewer.



**Note:** Depending on your configuration, a third option may be available: OpenText Intelligent Viewing. For help on enabling this third option, refer to [OpenText Intelligent Viewing integration](#).

- **Browser Native Viewer:** 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 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 the content will be 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.
  - For **Viewer Options**, indicate whether the user will be able to **Preview** the content contained in the panel's search results.
  - For **MIME Type**, indicate whether it will be **Static** or **From a column**.  
If **Static** is selected, select the format for a valid content type.  
If **From a column** is selected, select the column.
- **OpenText Information Archive Viewer** supports the following content types: PDF, BMP, TIFF, PNG, JPEG, and GIF.
  - For **Viewer Options**, indicate whether the user will be able to **Print**, **Download** and/or **Preview** the content contained in the panel's search results.
  - For **MIME Type**, indicate whether it will be **Static** or **From a column**.  
If **Static** is selected, select the format for a valid content type.  
If **From a column** is selected, select the column.

### Viewer for Mime Types

Click + Add Row and complete the following:

- Select a **Mime Type**.
- For **Viewer**, select one of the following:

- Browser Native Viewer
- Intelligent Viewer: Refer to the OTIV documentation for supported content types.

The Intelligent Viewer supports an extended set of MIME types/file formats (for example, Microsoft Word). For preview mode (optional viewer option, next to download and print), OpenText Information Archive does not use OTIV integration to render the preview and relies on standard HTML's <object> tag, which can be different based on the unstructured MIME type). If customers still want to use the preview option, it will require a compatible browser plugin.

For **Viewer Options**, indicate whether the user will be able to **Print**, **Download**, and/or **Preview** the content contained in the panel's search results.

#### **Allow Download on Failure**

Select to allow the user to download content if the selected viewer cannot properly display the content.

#### **Include in Export**

If selected, allows the user of the search to export the search results of the selected column.

#### **Hide Column**

Indicate if you want the result column to be hidden from the user.

3. Click **OK**.
4. Click **Save**.

**Complete the following procedure when the Column Type is set to Viewable Export:**

1. Click the  icon to configure the desired column.
2. Enter the following information:

#### **Column Label**

Enter the label of the column, which will appear as the header in the result details.

#### **Column Type**

Select **Viewable Export**.

#### **Content Link Display**

Select one of the following:

- **Custom Label:** 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.

- **From a column:** The value entered depends on the type of application:
  - For a SIP application, select the column
  - For a table application, enter a column

For example, if a Customer ID is picked when the Content link display is **From a column**, during run time, the user sees the value that is returned for the customer ID, such as 000453.
- **Download Icon:** Select to display the default download icon in the search results during runtime.

---

#### Tooltip Text

Enter concise, helpful information about the field that appears in a small “hover box” when the user hovers the cursor over the control.

---

#### Viewer Type

Indicate the viewer that will be used to view the information, either Browser Native or OpenText Information Archive Viewer.



**Note:** Depending on your configuration, a third option may be available: OpenText Intelligent Viewing. For help on enabling this third option, refer to [OpenText Intelligent Viewing integration](#).

---

#### Viewer Options

Indicate the viewer that will be used to view the information, either Browser Native or OpenText Information Archive Viewer.



**Note:** Depending on your configuration, a third option may be available: OpenText Intelligent Viewing. For help on enabling this third option, refer to [OpenText Intelligent Viewing integration](#).

- **Browser Native Viewer:** 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 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 the content will be 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.
  - For **Viewer Options**, indicate whether the user will be able to **Preview** the content contained in the panel's search results.
  - For **MIME Type**, indicate whether it will be **Static** or **From a column**.  
If **Static** is selected, select the format for a valid content type.  
If **From a column** is selected, select the column.
- **OpenText Information Archive Viewer** supports the following content types: PDF, BMP, TIFF, PNG, JPEG, and GIF.

- For **Viewer Options**, indicate whether the user will be able to **Print**, **Download** and/or **Preview** the content contained in the panel's search results.
- For **MIME Type**, indicate whether it will be **Static** or **From a column**.  
If **Static** is selected, select the format for a valid content type.  
If **From a column** is selected, select the column.

---

#### Viewer for Mime Types

Click + Add Row and complete the following:

- Select a **Mime Type**.
- For **Viewer**, select one of the following:
  - Browser Native Viewer
  - Intelligent Viewer: Refer to the OTIV documentation for supported content types.

The Intelligent Viewer supports an extended set of MIME types/file formats (for example. Microsoft Word). For preview mode (optional viewer option, next to download and print), OpenText Information Archive does not use OTIV integration to render the preview and relies on standard HTML's `<object>` tag, which can be different based on the unstructured MIME type). If customers still want to use the preview option, it will require a compatible browser plugin.

For **Viewer Options**, indicate whether the user will be able to **Print**, **Download**, and/or **Preview** the content contained in the panel's search results.

---

#### Allow Download on Failure

Select to allow the user to download content if the selected viewer cannot properly display the content.

---

#### Export Options

Select the pipeline for the export process. A pipeline represents a series of processing steps for the selected search result.

By default, OpenText Information Archive provides a number of tenant-level pipelines, and example applications may include custom application-specific pipelines. You can also define your own custom pipelines. For more information, see [Configuring the export of search results](#).

---

3. Click **OK**.
4. Click **Save**.

### 6.6.12 Configuring search result export

When a Developer creates or edits a search, they can allow search results to be exported in a specific format, such as HTML, PDF, CSV, and images. Users can then use this data in other applications, such as Microsoft Excel. The Developer can also configure whether the search can be exported as a background request or an instant download, which is also known as a synchronous export.

Export functionality is displayed to users on the list of search results 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. On the search result page, the **Export Options** drop-down list contains the options that were selected by the Developer during search composition, grouped by **Instant Download** and **Prepare Export in Background**.

For SIP applications, the Developer must enable the export functionality on the main search results page:

- The Developer can configure which export configurations are available for exporting both packages and purge list records.
- It is possible to export a package via a set of search results.
- If you want to export a repeating field (for example, a record with multiple attachments), you should use JSON as the export format rather than CSV. This is because a CSV file cannot display repeating fields in Microsoft Excel. Another advantage of JSON is that it can include supplemental data, such as a retention policy and approval date. You can view a JSON file in a text editor.

For SIP and table applications:

- When exporting purge lists for records, records not eligible will not be exported (meaning that a hold has been applied after the list was generated).
- There is no default for export configurations for searches. A configuration must be enabled by the Search Designer.

List Type	Export Format	Export Type	Data Exported	Encrypted Fields
Package	Based on result master	Per package	Only eligible records	Decrypted if configured to export
Table	CSV/ZIP only	Per table	Only raw table rows	Excluded (based on schema)
Record (AIU)	Based on result master	Per purge list	Only eligible records	Included decrypted, if specified

List Type	Export Format	Export Type	Data Exported	Encrypted Fields
Record (Table Row)	CSV/ZIP only	Per purge list	Only eligible raw records	Excluded (based on schema)

When a user downloads a search result export, the .zip option downloads with a .zip extension.

If there are multiple files in the archive, the content will be a TAR file, and the full file extension will be .tar.gz, .tgz or tar.gz, assuming the corresponding export pipelines are enabled.

There is an upper limit to the number of records that can be exported as an instant download, and this limit is known as the size threshold. You can set the size threshold at the application level or the tenant level, and the default size threshold is 50 records. The appropriate upper limit depends on your configuration and data. If the limit is too high and it takes longer than 60 seconds to prepare an instant download for a user, the request times out.

A user will only see instant download options in the **Export Options** drop-down list if the user selected a number of records equal to or less than the size threshold.

At the application level or tenant level, you can also choose to enable or disable instant downloads.



**Note:** If you configure the search results to include the ability to export to the main or detail tab, ensure that the export pipeline and format to use are specified. Otherwise, the export action will be removed upon saving the change.

#### To enable or disable synchronous exports at the application-level or tenant-level:

1. In a text editor, do one of the following:
  - If you want to enable or disable synchronous exports at the application level, open the application's config/application-config/exports/configuration.yml file.
  - If you want to enable or disable synchronous exports at the tenant level, open the <IA\_ROOT>/first-time-setup/applications/tenant/config/exports/configuration.yml file.
2. In the exportConfigurations section, set the appropriate sizeThreshold parameter to one of the following:
  - Greater than 0 if you want to enable synchronous exports (until the indicated threshold).
  - To 0 if you want to disable synchronous exports.

You must reinstall the Tenant application for this change to go into effect.

### 6.6.12.1 Configuring a result column to allow search result export

A Developer can configure a column in the search results so that each row has a link for exporting just that row, either to download or view.

If synchronous exports are enabled at the application level or tenant level, you can enable a downloadable export to be a synchronous export, which allows users to export the search result as an instant download rather than a background request. For more information about enabling synchronous exports at the application level or tenant level, see [Configuring search result export](#).

**To configure a result column to allow a downloadable export:**

1. Click the **cog** icon to configure the desired column.
2. Enter the following information:

---

#### Column Label

Enter the label of the column, which will appear as the header in the result details.

---

#### Column Type

Select **Downloadable Export**.

---

#### Content Link Display

Select one of the following:

- **Custom Label:** 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.
- **From a column:** The value entered depends on the type of application:
  - For a SIP application, select the column
  - For a table application, enter a columnFor example, if a Customer ID is picked when the Content link display is **From a column**, during run time, the user sees the value that is returned for the customer ID, such as 000453.
- **Download Icon:** Select to display the default download icon in the search results during runtime.

---

#### Tooltip Text

Enter concise, helpful information about the field that appears in a small “hover box” when the user hovers the cursor over the control.

---

#### Export Options

Select the pipeline for the export process. A pipeline represents a series of processing steps for the selected search result.

By default, OpenText Information Archive provides a number of tenant-level pipelines, and example applications may include custom application-

specific pipelines. You can also define your own custom pipelines. For more information, see [Configuring the export of search results](#).

**Enable Synchronous Exports**

Select if you want to enable instant downloads for search exports.

3. Click OK.

**To configure a result column to allow a viewable export:**

1. Click the  icon to configure the desired column.
2. Enter the following information:

**Column Label**

Enter the label of the column, which will appear as the header in the result details.

**Column Type**

Select **Viewable Export**.

**Content Link Display**

Select one of the following:

- **Custom Label:** 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.
- **From a column:** The value entered depends on the type of application:
  - For a SIP application, select the column
  - For a table application, enter a columnFor example, if a Customer ID is picked when the Content link display is **From a column**, during run time, the user sees the value that is returned for the customer ID, such as 000453.
- **View Icon:** Select to display the default view icon in the search results during runtime.

**Tooltip Text**

Enter concise, helpful information about the field that appears in a small "hover box" when the user hovers the cursor over the control.

**Viewer Type**

Indicate the browser that will be used to view the information, either **Browser Native** or OpenText Information Archive Viewer.

**Viewer Options**

Indicate whether the user will be able to **Preview** the content contained in the panel's search results.

**MIME Type**

Select one of the following:

- **Static:** Indicate the format of the content being viewed.
  - If **Viewer Type** is set to **Browser Native**, enter the MIME type.
  - If **Viewer Type** is set to OpenText Information Archive Viewer, select the MIME type from the drop-down list.
- **From a column:** Assign the MIME type from the value in the result set.
  - For SIP archiving, select the column name from the drop-down list.
  - For table archiving, enter the column name from the return section of the XQuery.
- 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.

---

#### Export Options

Select the pipeline for the export process. A pipeline represents a series of processing steps for the selected search result.

By default, OpenText Information Archive provides a number of tenant-level pipelines, and example applications may include custom application-specific pipelines. You can also define your own custom pipelines. For more information, see [Configuring the export of search results](#).

---

3. Click **OK**.

#### 6.6.12.2 Configuring the HTML Content View

To configure an HTML Content View, a column must be created and the following steps must be completed:

1. On the **Result List** tab, click the action icon for the column that you are adding for the Content View.
2. Complete the following fields:

---

##### Column Label

Enter the name of the column.

---

##### Column Type

Select **View HTML Content**.

---

##### Content Link Display

Select **Custom Label** and enter a name for the link.

---

##### Rendition Configuration

Enter the name of the HTML export configuration.

### Export Configuration

Optionally, the name of an export configuration. Often, the type of the transformation result is PDF.

3. Click **OK**.

When done, the column type in the Result List grid is set to **Embeddable Export**.

For further information, refer to [HTML Content View](#).

#### 6.6.12.3 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 exported CSV shows a column per repeating attribute table (Group). In the following example, 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"
```

#### 6.6.12.4 Setting sort functionality on repeating elements

The search Developer can set sort functionality on the repeating elements shown in an inline panel. Columns such as View, Content Download or encrypted columns cannot have the sort functionality applied to them.

To apply this functionality, edit a search. Navigate to the **Result Detail > Inline Panel** page. The column dedicated to column names identifies which result columns contain repeating element. For example, a column indicates “Item Description(repeating)” to let the Developer know that this result is a repeating element. Click the **cog** icon to edit the **Sort** and **Sort Order** fields for this particular result column.

#### 6.6.13 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 table columns.

In the **Result Detail** tab, the search designer can hide fields and tabs in the detail or inline panels of the search results.

Once the Hide Field or Hide Tab option is selected, the **Conditional** field is displayed. Once the **Conditional** field is selected, the builders are displayed that allow the designer to construct a condition statement.

**In the condition builder:**

1. Select a field name. The list includes all the values applicable to the selected search results.
2. Select an operator. The available operators depend on the data type selected in the previous step.
3. Select a value, either **Literal** or **Value Of**:
  - If **Literal** is selected, enter an applicable value.
  - If **Value Of** is selected, select the applicable field name.

Multiple conditions can be set for any field or tab. Click + to add another condition.

For example, a search designer wants to hide a specific invoice number from appearing in the detail panel of the search results for a specific search form. The user selects the invoice number field in the first step, the Equals operator in the second step, and selects **Literal** for the third step. Then, the user enters the invoice number (181916) in the available field. Whenever a search user runs a search using the form, the Invoice Number field will not be displayed if it contains the value '181916'.

Conversely, if conditions are applied to a tab, the tab will be hidden in run-time if the conditions are true.

#### **6.6.13.1 Adding details to a side panel**

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 or package:

1. In the **Result Detail** tab for the selected SIP search, select either **Select from Schema** or **Select from Package**.
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.

### 6.6.13.2 Adding an in-line panel

The following shows the **Result Detail tab > Detail 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 or package:

1. In the **Result Detail** tab for the SIP search, either **Select from Schema** or **Select from Package**.
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

### 6.6.13.3 Configuring autoexpand of in-line panels

By default, in-line panels are collapsed.

**To automatically expand in-line panels when results are viewed:**

1. Click the **Result List** tab in the search composition.
2. Select the **Expand Inline Panels** checkbox.
3. Click **Save** to save the search composition.

### 6.6.13.4 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**.

To remove a tab, click the X beside the desired tab.

To change the name of a tab:

1. Click the **cog** icon for the desired tab. Beside naming a tab, a search designer can opt to HIDE or HIDE CONDITIONALLY. This applies to tabs on the detail and inline panels.
2. Enter a **Tab Name** and click **SAVE**.

### 6.6.13.5 Customizing panel fields

After adding fields to a detail or inline panel, you are able to further customize them.

1. Click the  icon for the desired field.
2. Enter a **Field Label**. The name entered here will appear in the panel when a user runs a search and views the results in runtime.
3. Select a **Field Type**, depending on the type of application you are working with:

SIP Application	Table Application
Linked Search	Linked Search
External URL	External URL
Downloadable Content	Downloadable Content
View Content	View Content
Schema	Query Reference
Package	

4. Complete the rest of the fields on the **Edit Field > Properties** tab. The fields available depend on the **Field Type** value.

---

#### Query Reference (only available for table applications)

<b>Field Name</b>	Enter a name that will appear in the panel's search results.
<b>Data Type</b>	Select the PostgreSQL data type. If Number, Integer, Double, Long, or Decimal is selected, configure the following: <ul style="list-style-type: none"><li>• <b>Disable Thousands Separator</b>: Select to remove the thousands separator when displaying a number. For example, in English, the thousands separator is a comma: 12,345.</li><li>• <b>Format</b>: Allows you to control the format of dates and numbers, based on a user's locale information.</li></ul> If Date or DateTime is selected, configure the following: <ul style="list-style-type: none"><li>• <b>Display Original Time Zone</b>: For more information, see <a href="#">Disabling local time zone in search results</a>.</li><li>• <b>Format</b>: Allows you to control the format of dates and numbers, based on a user's locale information.</li></ul>
<b>Binding</b>	Select the binding.
<b>Sensitive Information</b>	Select to ensure that, when the user runs a background search, that the results are encrypted.

<b>Masked</b>	Indicate if the column is masked in the query.
<b>Include in Export</b>	If selected, allows the user to export the search results of the selected column.
<b>Hide Field</b>	Indicate whether you want the field to be hidden from the user.
<b>Conditional</b>	<p>Indicate if the field is to be hidden under the specified conditions. If selected:</p> <ul style="list-style-type: none"> <li>a. Select the field name. The list includes all the values applicable to the selected search results.</li> <li>b. Select the operator name. The available operators depend on the data type selected in the previous step.</li> </ul> <p>To learn how to configure the remaining fields, see <a href="#">Composing result details</a>.</p>

#### Schema or Package (only available for SIP applications)

<b>Field Name</b>	Select a name that will appear in the panel's search results.
<b>Custom</b>	For a search form in a SIP application, select to enter a custom column path in the field provided. For more information, see <a href="#">Configuring advanced XQuery expressions on SIP search results</a> .
<b>Data Type</b>	<p>Select the PostgreSQL data type.</p> <p>If Number, Integer, Double, Long, or Decimal is selected, configure the following:</p> <ul style="list-style-type: none"> <li>• <b>Disable Thousands Separator:</b> Select to remove the thousands separator when displaying a number. For example, in English, the thousands separator is a comma: 12,345.</li> <li>• <b>Format:</b> Allows you to control the format of dates and numbers, based on a user's locale information.</li> </ul> <p>If Date or DateTime is selected, configure the following:</p> <ul style="list-style-type: none"> <li>• <b>Display Original Time Zone:</b> For more information, see <a href="#">Disabling local time zone in search results</a>.</li> <li>• <b>Format:</b> Allows you to control the format of dates and numbers, based on a user's locale information.</li> </ul>
<b>Sensitive Information</b>	Select to ensure that, when the user runs a background search, that the results are encrypted.
<b>Include in Export</b>	If selected, allows the user to export the search results of the selected column.
<b>Hide Field</b>	Indicate whether you want the field to be hidden from the user.

<b>Conditional</b>	Indicate if the field is to be hidden under the specified conditions. If selected: <ol style="list-style-type: none"><li>Select the field name. The list includes all the values applicable to the selected search results.</li><li>Select the operator name. The available operators depend on the data type selected in the previous step.</li></ol> To learn how to configure the remaining fields, see <a href="#">Composing result details</a> .
--------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

The remaining field types are available for both table and SIP applications.

### Linked Search

<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>Application</b>	Select the application that contains the search form being linked to.
<b>Linked Search Form</b>	Select the search form being linked to.
<b>Linked Search Mapping</b>	Select a <b>Value From Search Result</b> . The available mapping options are from the result list and the result detail. Select a <b>Linked Search Field</b> . The available mapping options are from the linked search. Click  to add an additional mapping configuration.

---

### External URL

<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.
---------------------	------------------------------------------------------------------------------------------------------------------------------------------

To learn how to configure the remaining fields, refer to [Adding an external link to search results](#).

---

### Downloadable Content

<b>Detail Field Name</b>	Select the name that will appear in the panel's search results.
<b>Custom</b>	For a search form in a SIP application, select to enter a custom column path in the field provided. For more information, see <a href="#">Configuring advanced XQuery expressions on SIP search results</a> .

<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 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>
<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>Include in Export</b>	If selected, allows the user to export the search results of the selected column.
<b>Hide Field</b>	Indicate whether you want the field to be hidden from the user.
<b>Conditional</b>	Indicate if the field is to be hidden under the specified conditions. If selected: <ol style="list-style-type: none"><li>a. Select the field name. The list includes all the values applicable to the selected search results.</li><li>b. Select the operator name. The available operators depend on the data type selected in the previous step.</li></ol>

### View Content

<b>Detail Field Name</b>	Select the name that will appear in the panel's search results.
<b>Custom</b>	For a search form in a SIP application, select to enter a custom column path in the field provided. For more information, see <a href="#">Configuring advanced XQuery expressions on SIP search results</a> .
<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 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>
<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>Viewer Type</b>	<p>Indicate the viewer that will be used to view the information, either Browser Native or OpenText Information Archive Viewer.</p> <p> <b>Note:</b> Depending on your configuration, a third option may be available: OpenText Intelligent Viewing. For help on enabling this third option, refer to <a href="#">OpenText Intelligent Viewing integration</a>.</p> <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 native viewer is launched by the browser depending on the <b>MIME Type</b> provided. If the content plug-in for that MIME type is installed, it would be launched and the content will be 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.<ul style="list-style-type: none"><li>– For <b>Viewer Options</b>, indicate whether the user will be able to <b>Preview</b> the content contained in the panel's search results.</li><li>– For <b>MIME Type</b>, indicate whether it will be <b>Static</b> or <b>From a column</b>. If <b>Static</b> is selected, select the format for a valid content type. If <b>From a column</b> is selected, select the column.</li></ul></li><li>• <b>OpenText Information Archive Viewer</b> supports the following content types: PDF, BMP, TIFF, PNG, JPEG, and GIF.<ul style="list-style-type: none"><li>– For <b>Viewer Options</b>, indicate whether the user will be able to <b>Print</b>, <b>Download</b> and/or <b>Preview</b> the content contained in the panel's search results.</li><li>– For <b>MIME Type</b>, indicate whether it will be <b>Static</b> or <b>From a column</b>. If <b>Static</b> is selected, select the format for a valid content type. If <b>From a column</b> is selected, select the column.</li></ul></li></ul>
--------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>Viewer for Mime Types</b>	<p>Click + <b>Add Row</b> and complete the following:</p> <ul style="list-style-type: none"> <li>• Select a <b>Mime Type</b>.</li> <li>• For <b>Viewer</b>, select one of the following:           <ul style="list-style-type: none"> <li>– Browser Native Viewer. Refer to the browser's documentation to learn about the supported document, image, and video formats.</li> <li>– OpenText Information Archive Viewer (Brava!). To view a list of supported document, image, and video formats, click here (<a href="https://www.opentext.com/file_source/OpenText/en_US/PDF/opentext-so-brava-enterprise-supported-formats-en.pdf">https://www.opentext.com/file_source/OpenText/en_US/PDF/opentext-so-brava-enterprise-supported-formats-en.pdf</a>).</li> <li>– Intelligent Viewer: Refer to the OTIV documentation for supported content types.</li> </ul> </li> </ul> <p>The Intelligent Viewer supports an extended set of MIME types/file formats (for example. Microsoft Word). For preview mode (optional viewer option, next to download and print), OpenText Information Archive does not use OTIV integration to render the preview and relies on standard HTML's <code>&lt;object&gt;</code> tag, which can be different based on the unstructured MIME type). If customers still want to use the preview option, it will require a compatible browser plugin. To view a list of supported document, image, and video formats, click here (<a href="https://www.opentext.com/assets/documents/en-US/pdf/opentext-intelligent-viewing-supported-formats-en.pdf">https://www.opentext.com/assets/documents/en-US/pdf/opentext-intelligent-viewing-supported-formats-en.pdf</a>).</p> <p>For <b>Viewer Options</b>, indicate whether the user will be able to <b>Print</b>, <b>Download</b>, and/or <b>Preview</b> the content contained in the panel's search results.</p>
<b>Allow Download On Failure</b>	Select to allow the user to download content if the selected viewer cannot properly display the content.
<b>Include in Export</b>	If selected, the data in the selected column can be exported.
<b>Hide Field</b>	Indicate whether you want the field to be hidden from the user.
<b>Conditional</b>	<p>Indicate if the field is to be hidden under the specified conditions. If selected:</p> <ol style="list-style-type: none"> <li>1. Select the field name. The list includes all the values applicable to the selected search results.</li> <li>2. Select the operator name. The available operators depend on the data type selected in the previous step.</li> </ol> <p>To learn how to configure the remaining fields, see <a href="#">Composing result details</a>.</p>

5. Click **OK**.
6. Click **Save**.

### 6.6.13.6 Configuring a custom view for an inline panel

The search designer is able to configure an inline panel to display more metadata by configuring a custom view. If custom view is configured for an inline panel, you will not be able to configure a custom view for the search result list. To configure a custom view form the result list, remove the custom view from the inline panel.



**Tip:** The inline panel for the Invoice Number search in the Invoices application includes an out-of-the-box example of this feature.

#### To configure a custom view:

1. Navigate to the desired tab in the **Result Detail > Inline Panel** tab.
2. Enable the **Custom View** switch.
3. From the list, select the custom view. The options displayed include presenting the search results in JSON format and any custom views configured for the searches in the selected application.
4. Click **Save**.

### 6.6.13.7 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.

## 6.6.14 Configuring a horizontal scroll bar

When the **Result List** page and in-line panels have a large number of columns to display, they shrink the data to the point that it becomes unreadable. To remedy this situation, you have the option to enable a horizontal scroll bar.

#### To enable a horizontal scroll bar:

1. Click the **Result List** tab in the search composition.
2. Select the **Allow Horizontal Scrolling** check box.
3. Click **Save** to save the search composition.

### 6.6.15 Configuring search results page size

By default, the search results page displays 10 records per page and provides the ability to set that number to 20, 30, 40, or 50. For searches where fewer or more records per page are optimal, the search developer has the option to configure the default page size for a search as well as the other possible values.

#### To configure the default results page size:

1. Click the **Result List** tab in the search composition.
2. Click the **Configure** cog above the search columns.
3. In the **Configure Page Sizes** dialog, in the **Rows per page** field, enter a comma-delimited list of the desired possible search result values, e.g., 5,10,15,20,25.
4. In the **Default Page Size** field, enter the desired default page size, e.g. 5.
5. Click **Save**.
6. Click **Save** to save the search composition.

### 6.6.16 Custom view

The Developer can apply a custom layout to a search result set. The custom layout must be created and loaded to the system before it can be used.

#### To apply a customized layout to a result set:

1. Click the  icon and select **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 runs, the search results will be displayed according to the selected layout.

### 6.6.17 Enabling a timeline view of results

Using the Timeline View feature, the Developer can configure a search result set so that users performing date-based searches can view results in a timeline and select an element in the timeline to filter the results to display.

For example, a timeline view configured for an invoices number search allows the user to view invoice activity in a timeline. The user can then select an element in the timeline to view invoice activity for a particular year, month, day, and even time. In addition, by restricting search results to a particular customer, the user can view details of that customer's last invoice activity.

The feature allows for more efficient searching by providing more granular access to data, especially in the case of large data sets.

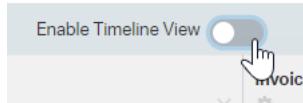


**Note:** The Timeline View feature is not supported in Nested searches, Delegate searches, or custom views.

**Limitation:** If the search designer selects **Encrypt for background search cache** for the column used for a timeline view, the **Sort** field is hidden. This is because columns encrypted in the results cannot be sortable and filterable.

#### To configure a timeline view for an invoice date search:

1. Click the **Result List** tab in the search composition.
2. Click the **Enable Timeline View** toggle.



3. In the **Timeline View Configuration** dialog, select **Invoice Date** as the **Date Field** that will be used by the timeline.



#### Important

Fields used in a timeline must be sortable and filterable. If a field is not sortable or filterable, it is labeled as such in the drop-down value.

4. Set the **Timeframe** slider to **Days**.
5. Under **Orientation** select how you want the timeline to appear during the search (to the left of the search results table or along the top). The default is **Vertical**.

Timeline View Configuration

Configure the timeline view for the search result screen.

Date Field \* InvoiceDate (DATE)

Time Frame \* Days

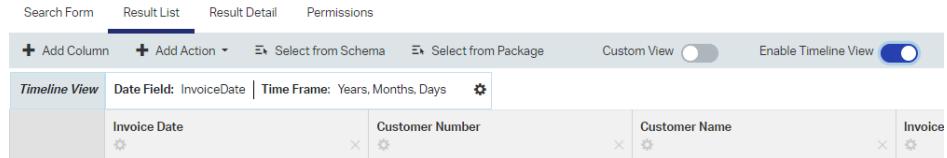
Years Months Days Hours Minutes Seconds

Orientation \* Vertical

Cancel Save

6. Click **Save**.

You should now see the Timeline View configuration displayed above the results table. To edit the configuration, click the icon.



7. Click **Save** in the upper-right corner of the screen.

**To view results in the timeline view:**

- In the Invoices application, run the **Invoice Date** search, entering today as the last date in the **Invoice Date** range.

You should now see the timeline with columns for **Year**, **Month**, and **Day** showing results for **Year**. The number in brackets next to a year represents the results count for that year.



**Note:** The **Year** column is in reverse-chronological order by default (last year first). To reverse the order, click the down arrow next to the column label. In the following screenshot, the numbers contained in parenthesis represent the number of records found in that particular year:

Invoice Date		
Year ▾	Month	Day
<span>2016 (3)</span>		
<span>2015 (3)</span>		
<span>2004 (1)</span>		
<span>2003 (1)</span>		
<span>2002 (2)</span>		
<span>2001 (4)</span>		
<span>2000 (3)</span>		

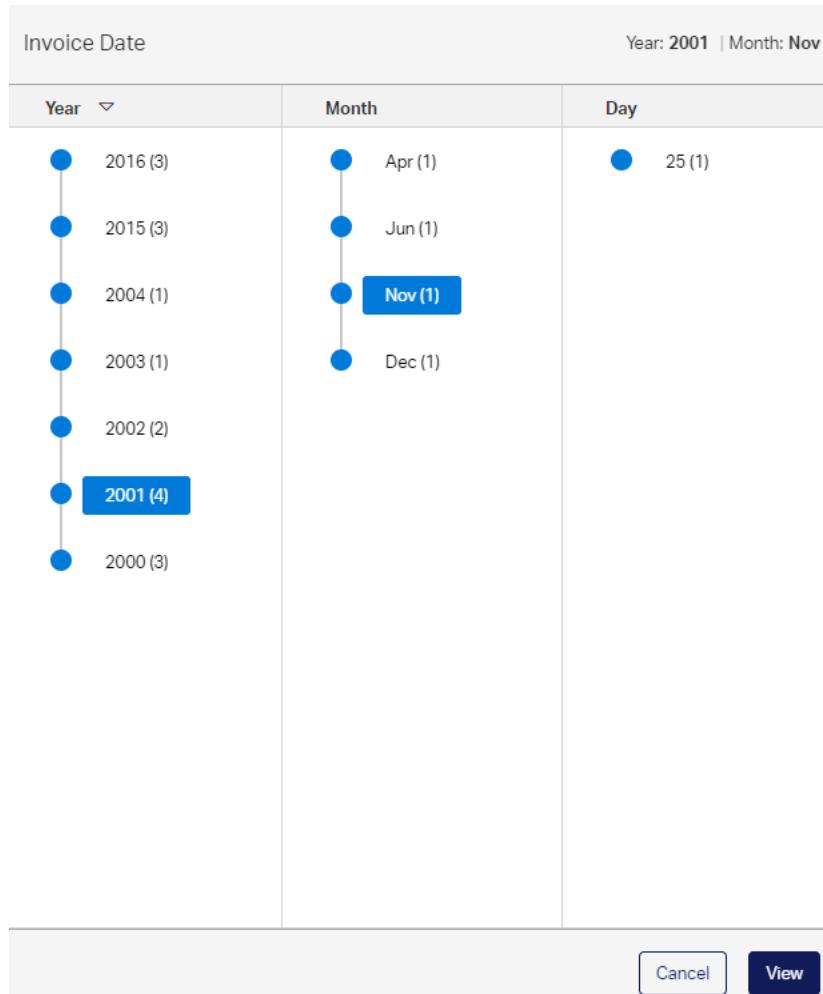
Cancel View



**Note:** In this example, the timeline is based on a DATE field type, so it consists of Years, Months, and Days. In a Phone Calls application search, for example, where phone call date and time data is stored, the timeline can be based on the phone call date and time (DATETIME field type) such that the timeline will consist of Hours, Minutes, and Seconds in addition to Years, Months, and Days.

2. To show results of invoice activity for November 2001, for example, click **2001(4)** in the **Year** column and **November(1)** in the **Month** column.

You should now see results in all three columns of the timeline.



3. To display results in the results table, make sure **Nov(1)** is selected in the **Month** column then click **View** at the bottom-right of the timeline.

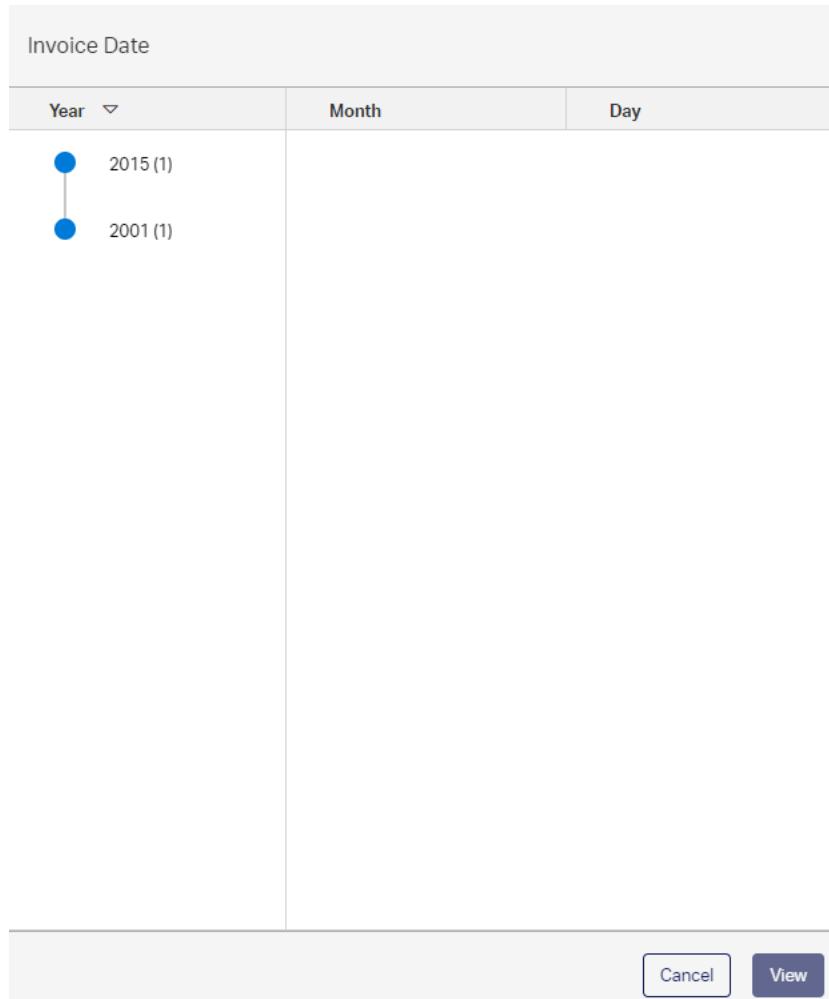
The results appear in the results table with breadcrumbs above displaying the selected year and month.

The screenshot shows a search results page for an invoice date search. The top navigation bar includes links for Search Forms, Retention Sets, Application Info, Spaces, Stores, Rules, Packages, and Holdings. Below the navigation is a breadcrumb trail: Applications > Invoices > Invoice Date > Invoice Date Results. A search bar at the top contains the query "Invoice Date: 1/1/2000 (-05:00) - 3/1/2022 (-05:00)". The main content area displays a table with the following data:

Invoice Date	Customer Num...	Customer Name	Invoice Number	Ship Date	Invoice
11/25/2001	76AN3	Collins Educational	7	04/22/1998	<a href="#">Download</a>

At the bottom of the table, there is a message "Displaying 1 - 1 of 1". To the right of the table, there are buttons for "Rows per page" (set to 10) and a dropdown menu.

4. To view details of a particular customer's invoice activity, re-run the **Invoice Date** search entering the customer's name (e.g., **Novell**).  
You should now see a timeline view of the customer's invoice activity broken out by **Year**.



5. To view details of the customer's last invoice, select the last **Year**, **Month**, and **Day** in the timeline.

Invoice Date			Year: 2015   Month: Dec
Year ▾	Month	Day	
<input checked="" type="radio"/> 2015 (1) <input type="radio"/> 2001 (1)	<input checked="" type="radio"/> Dec (1)	<input checked="" type="radio"/> 3 (1)	
			<a href="#">View</a> <a href="#">Cancel</a>

6. To view results in the results table, click **View**.

Search Forms Retention Sets Application Info Spaces Stores Rules Packages Holdings

Applications > Invoices In Test > Invoice Date > Invoice Date Results ⓘ

Invoice Date: 1/1/2000 (-05:00) - 3/1/2022 (-05:00) Customer Name: Novell ⚪

Invoice Date Year: 2015 (1) > Month: Dec (1) ⚪

Results 0.30 seconds ⚪

Invoice Date	Customer Num...	Customer Name	Invoice Number	Ship Date	Invoice
▶ 12/03/2015	8AX017	Novell	6	12/08/2015	Download

Displaying 1 - 1 of 1 Rows per page 10 (Default) ⚪

## 6.6.18 Configuring advanced XQuery expressions on SIP search results

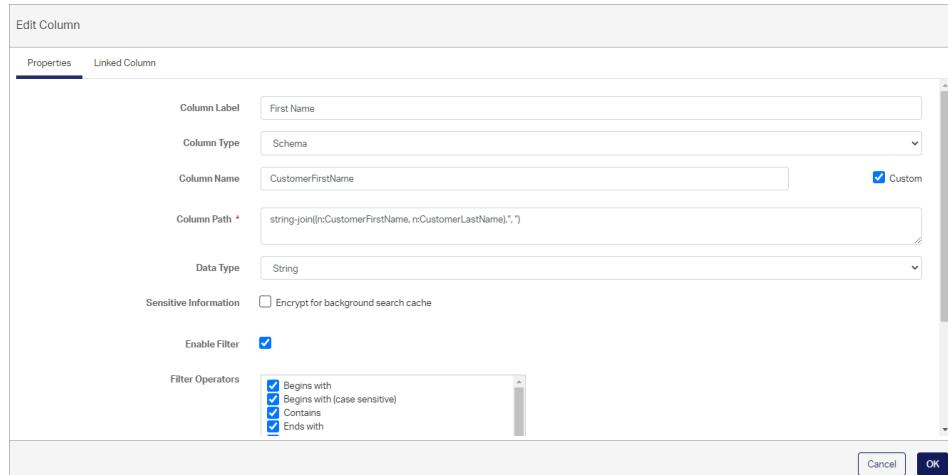
When configuring search results of a SIP application, the Developer has the option of using custom XQuery expressions to configure how the value from the AIU (XML) should be displayed in the search results. This feature is useful if you want to manipulate the data coming from the AIU node. Some typical examples are:

- Computing the difference between two dates to display the duration in seconds
- Joining some values, for example, first name and last name

**To configure a custom XQuery expression to join first and last name values:**

1. In the search **Result List** tab, click the  icon for the field/value you want to manipulate (e.g., **First Name**).
2. In the **Edit Column Properties** tab, select the **Custom** check box to the right of the **Column Name** field.
3. In the **Column Path** field (now enabled), enter the following:

```
string-join((n:CustomerFirstName, n:CustomerLastName), ", ")
```



**Note:** The **Column Name** must be unique. If not, the **Column Path** and **Data Type** must be consistent. An error will occur during the save or during the import of the configuration if these conditions are not satisfied

4. Click **OK**.
5. Click **Save** in the upper-right corner of the **Result List** tab.



**Note:** All standard XQuery 3.0 functions are available.

## 6.7 Creating a custom view using the IA Web App

The IA Web App allows you to easily create UI screens that allow end users to view search results in customized presentations. The **Custom Views** tab allows you to modify a default template or enter your own data to create a custom UI. While the system still supports previous methods of creating custom presentations, the **Custom Views** method simplifies the process.

The **Custom Views** tab contains any previously created custom views:

---

### Name

The name given to the custom view.

---

### Label

The label provided for the custom view. The value entered here is displayed on the **Result List** and **Result Detail** tabs when creating or editing a search and the **Custom View** switch is enabled. For more information, see [Configuring a custom view for an inline panel](#).

---

### In Use

Indicates whether the custom view is currently in use.

Along with the information above, a side panel also contains the following information related to the selected custom view:

---

### Static Resource enabled

If set to `true`, the custom view includes a related resource. If set to `false`, the custom view does not include a related resource.

---

### Created By

Name of the user who create the custom view.

---

### Created Date

Date the custom view was created.

---

### Modified Date

Date the custom view was last modified.

---

### Modified By

Name of the user who last modified the custom view.

---

### To create a custom view:

1. Click **Create Custom View** 
2. Complete the following fields of the **Summary** tab:
  - a. **Name:** Enter a unique name for the custom view.
  - b. **Label:** Enter a label for the custom view.
  - c. **Description:** Enter a description for the custom view.

3. For the **HTML Template** tab, complete one of the following:
  - Enter an HTML template in the space provided; or
  - Click **Default Template**, which sets the transformation content to the JSON view template based on the transformation type, input, and output format. Clicking **Default Template** overwrites the HTML content. When prompted to confirm if you want to continue, click **OK**.  
Update the `img` class path for any image files to be displayed in the custom presentation.
4. For the **Related Resource** tab, click **Choose** to select the desired resource. Click **Clear** to remove the related resource. Click **Replace** to change the related resource.
5. Click **Create**.



**Tip:** Custom views can also be enabled for results provided by Content Aviator. For more information, see [Custom views and Content Aviator](#).

## 6.7.1 Editing a custom view

**To edit a custom view:**

1. Select the custom view being edited and click **Edit**
2. Make changes to the information in the **Summary**, **HTML Template**, or **Related Resource** tabs.
3. When you are done editing the custom view, click **Save**.

## 6.7.2 Duplicating a custom view

**To duplicate a custom view:**

1. For the custom view you want to duplicate, click and select **Duplicate**.
2. Enter a **Name**, **Label**, and **Description** for the duplicated custom view.
3. Click **OK**.

### 6.7.3 Deleting a custom view

If a custom view is in use, it cannot be deleted.

**To delete a custom view:**

1. For the custom view you want to delete, click  and select **Delete**.
2. When prompted to confirm that the custom view is to be deleted, click **Delete**.

### 6.7.4 Custom views and Content Aviator

This section only applies if Content Aviator is enabled on your system. For more information, see Section 3.12.1 “Enabling Content Aviator” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

To enable Content Aviator at the record level for custom views, further configuration is required.



**Tip:** The PhoneCalls example application includes a custom view called `PhoneCalls-FirstNameCheckboxView`, which contains all the information outlined in the following procedure. Refer to the custom view’s **HTML Template** tab for more information.

**Use the HTML Template tab for the desired custom view to enable Content Aviator at the record level:**

1. Add a column with the following check:

```
cc.isAviatorEnabled = IACustomPresentationService.isAviatorEnabled();
```



**Note:** The `isAviatorEnabled` is a boolean property that determines if the column needs to be included or not. If the `isAviatorEnabled` property is not available, it means that Content Aviator has not been enabled on your system.

2. Create a function to call when the Content Aviator icon is clicked for a search result record in runtime:

```
/**  
 * Initiate record level aviator chat, pass the record and row index  
 */  
cc.initiateAviatorRecordChat = function (record, index) {  
    $scope.$emit('IA_SEARCH_CUSTOM_RESULT_AVIATOR_CHAT', {record: record,  
index: index});  
};
```

3. Add the following to the end of the row in the new column:

```
<td ng-if="cc.isAviatorEnabled" class="ia-aviator-action-column" ng-  
style="{ 'width': '3%' }">  
    <button ng-click="cc.initiateAviatorRecordChat(record, $index)" id="ia-search-  
result-aviator-row-{{$index}}" class="ia-aviator-action-button"  
title="{{$('ASK_AVIATOR_RECORD' | translate)}}>  
        
```

```
| translate}}}" class="ia-filter-blue">
    </button>
</td>
```

4. Click Save.

## 6.8 Advanced search configuration

### 6.8.1 Pre- and Post-Filtering

A SIP search is a search in an AIC that contains many AIUs (potentially from multiple holdings/schemas but, typically, only one). You may want to restrict the record access based on functional rules (for example, a Business Unit criteria) to not provide the same result for all users.

The filtering can be processed before running the query (pre-filtering) or after the search to write the search result in the cache (post-filtering).

- Pre-filtering is useful to enrich/modify submitted search criteria based on functional rules or context.
- Post-filtering is useful to dismiss some records based on functional rules or context.

A public interface `SipSearchHandler` is available to handle the two filtering scenarios, and custom implementation must be declared at the AIC-level under the `searchHandlerImpl` property by manually editing the `<APPLICATION_NAME>/config/application-config/searches/configuration.yml` file.

```
aic:
  name: ${holding.name}-aic
  searchHandlerImpl: com.acme.ia.search.CustomSipSearchHandler
  ...
```

The JAVA interface is available in the distribution in `<IA_ROOT>/lib/server/infoarchive-public-api-xx.x.jar`. To compile your project, it is necessary to include these libraries `<IA_ROOT>/lib/server/infoarchive-public-api-xx.x.jar` and `<IA_ROOT>/lib/server/slf4j-api-x.x.xx.jar`. The custom implementation and dependencies must be deployed in `<IA_ROOT>/lib/iaserver/external`.

**!** **Important**

Using post-filtering to filter a large dataset is not recommended because it may cause major performance issues.

## 6.8.2 Nested searches

A nested search is a search within a search, and it allows more flexibility for creating searches. For example, the results of a customer order search could include links for further order details. Clicking one of these links would then run a nested search for more information about the customer order, such as the receipt date.

There is no limit to how many levels appear in a nested search, so a nested search can contain another nested search.



**Note:** Only a Developer can see a nested search when viewing a list of the searches available for an application. Other users will only know about a nested search when they access it using the primary search.

When searches are nested, values are passed from the result list of the primary search into the form for the nested search. For example, a customer ID can be passed from a primary search for a customer order to a nested search for order details. You map these values when you create the searches.

You can map a range (date, dateTime, or number) from the primary search to the nested search. The from and to values, or minimum and maximum, are mapped separately. If you do not map these parameters, the nested search takes all values in the range.

During runtime, if a value is missing when a user clicks a nested search, it cannot run and the user is presented with the search form of the nested search so that the missing value can be entered.

## 6.8.3 Setting up Smart URL

OpenText Information Archive's Smart URL functionality provides a mechanism whereby applications can issue the command to run searches, and allows users to view the result page of OpenText Information Archive. If not already logged in, users need to provide their credentials to be taken to the result page.

To enable this feature, make sure to add the `infoarchive.iawebapp.profile.search` profile in the `<IA_ROOT>/config/iawebapp/application.yml` file (refer to the final lines in the following extract of the `application.yml` code):

```
profiles:  
  include:  
    - infoarchive.gateway.profile.AUTHENTICATION_IN_MEMORY  
    - CLIENTS  
    - infoarchive.iawebapp.profile.search
```

To be able to work with SSO, Smart URL functionality needs to work through an OTDS authentication layer. The `infoarchive.gateway.profile.OTDS` profile must be enabled in the `<IA_ROOT>/config/iawebapp/application.yml` file. The `infoarchive.gateway.profile.OTDS` profile is exclusive to other profiles and `CLIENTS` profiles. This is because, to use Smart URL feature, they have to use `infoarchive.iawebapp.profile.search`.



**Note:** If the Smart URL requests are frequently made, because of the redirect flow that causes the pages to reload, an error may occur. To recover from the error, restart your browser.

The following four parameters are mandatory. Otherwise, an error will be returned:

- infoarchive-user
- infoarchive-application-name
- infoarchive-search-name
- infoarchive-search-set-name

The first time the POST is attempted, the caller is challenged by the IA Web App login page. Successive calls to IA Web App go through as long as the login name and infoarchive-user posted by the caller are the same.

Upon successful login, more error checks are applied.

The following form is provided for testing and demo purposes only:

```
<html>
    <head>
    </head>
    <body>
        <h1>Sample Form for testing Smart URL</h1>
        <form action="http://localhost:8080/search" method="POST"
        target="infoarchive">
            <input type="text" name="infoarchive-user" value="connie@iacustomer.com"
            placeholder="Enter IA User Name"><br/>
            <input type="text" name="infoarchive-application-name" value=
            "PhoneCalls"><br/>
            <input type="text" name="infoarchive-search-name" value="A"><br/>
            <input type="text" name="infoarchive-search-set-name" value="Set 1"><br/>
            <h4>Search Criterias</h4>
            <input type="text" name="CustomerFirstName" value="Mia"><br/>
            <input type="text" name="CustomerLastName" value="Turner"><br/>
            <sub>For more criterias, need to add more input fields in the
            markup</sub><br/><br/>
            <button type="submit">Infoarchive >>></button>
        </form>
    </body>
</html>
```

The following form data is submitted using the following key value pairs:

infoarchive-user: connie@iacustomer.com
infoarchive-application-name: PhoneCalls
infoarchive-search-name: A
infoarchive-search-set-name: Set 1
CustomerFirstName: Mia
CustomerLastName: Turner

Make sure to set the target property of the form to some unique tab name (for example, infoarchive).

All the required parameters (1, 2, 3, and 4) mentioned above must be specified; otherwise, the IA Web App returns an error message and the job fails.

Additionally, the parameters for a particular search form need to be passed along with the required parameters. For example:

- CustomerFirstName: Mia
- CustomerLastName: Turner

OpenText Information Archive will challenge the user (POST action) the first time using the login page. A valid user permitted to run the search needs to login.

Once the user has logged in, the search runs, as long as the authenticated user has access to the search and search set mentioned in the form data.

Subsequent calls to the same instance do not require authentication. In fact, authentication expires after 30 minutes of inactivity, meaning the user will be need to login again if there is an activity gap of 30 minutes or more.

#### 6.8.4 Entering categories to organize search forms

Entering the following incorrect information causes the search not to run:

- Entering an invalid application name,
- Entering an invalid search name,
- Entering an invalid search set name,
- Entering an invalid criteria name,
- Entering an unmatched criteria name,
- Entering an unmatched user name, and
- Entering invalid user credentials on the login page.

#### 6.8.5 Searches and Glacier

Amazon Glacier is a more inexpensive (albeit slower) cloud location and service. The Developer can 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 runs, 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 OpenText Information Archive 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 can view or download the content.



**Note:** When content is copied to S3, the length of time the content will be available before requiring a restoration request 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 detail panel. The service (IAS) polls for content availability according to the following attribute set in the <IA\_ROOT>/config/iashell/application.yml file:

```
#Configuration for the order items
order:
  pollingDelayForRestorationOfflineContent: 900
  orderItemRetentionDays: 1
```

### 6.8.6 Content retrieval from Glacier

When searched content is stored in an offline stores, such as AWS Glacier, the search result will look different. The content cannot be downloaded immediately until it is not restored, so the download link specified for the search results will not be available and you will see a corresponding warning sign.

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 than see the new background search with a status of **In Progress**.

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.



**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.

### 6.8.7 HTML Content View

An HTML Content View is displayed when clicking on a link in a column of a search result item in the Results List. The HTML Content View displays the HTML result of a synchronous export transformation. It is displayed in a separate window.

The HTML Content View window can be configured with an export button that can be used to start the generation of a synchronous export transformation, based on the original search result item.

The Baseball sample application has an example of an HTML Content View, which is shown when running the **Search By Player Name** search and clicking on a **View** link in the **View Content** column.

For further information, refer to [Configuring the HTML Content View](#).

### 6.8.8 Search result queries

Each search run produces a search result, which is stored as XML in a Lucene index located under the search result store.

In some cases, additional analytical information must be retrieved from a search result. For instance, the sum, average, minimum or maximum calculation of search result values of a specific column.

This analytical information can be collected by running additional XQueries on the search result XML.

As search results can consist of a very large number of rows, it is required that a search result query is optimized by using the Lucene index. To ensure the indexes are created, columns accessed by the search result query must be set to filter- and sort-enabled in the ResultList.

The XQuery must use XQuery extension functions that access the Lucene index.

Examples of search result queries can be found in the Audit application Activity Reports search.

The following is a list of supported XQuery extension functions using the Lucene index. These functions are defined in namespace `urn:x-emc:ia:schema:fn`.

All functions have the complete search result as context.

Name	Description	Supported column data types	Number of parameters	Parameters	Return value
<code>ia:column-values</code>	Get the values of the search result columns of given name	All data types	1	<i>search result column name</i>	A sequence of string values
<code>ia:column-values</code>	Get the distinct values of the search result columns of given name filtered by SpEL expression	Long, Integer	2	<i>search result column name</i> <i>spel expression</i>	A sequence of long values

Name	Description	Supported column data types	Number of parameters	Parameters	Return value
ia:count	Get the total number of columns with given name filtered by SpEL expression.	Long, Integer, Date, DateTime	2	<i>search result column name spel expression</i>	Long value
ia:min	Get the minimum value of columns with given name filtered by SpEL expression.	Long, Integer, Date, DateTime	2	<i>search result column name spel expression</i>	Single value, depending on column data type
ia:max	Get the maximum value of columns with given name filtered by SpEL expression.	Long, Integer, Date, DateTime	2	<i>search result column name spel expression</i>	Single value, depending on column data type
ia:sum	Get the sum of all values of columns of given name filtered by SpEL expression.	Long, Integer	2	<i>search result column name spel expression</i>	Long value
ia:avg	Get the average of all values of columns of given name filtered by SpEL expression.	Long, Integer	2	<i>search result column name spel expression</i>	Double value

The following is an example search result XQuery containing XQuery functions optimized by Lucene:

```

declare namespace ia = 'urn:x-emc:ia:schema:fn';
<results>{
  for $date in ia:column-values('createdDateByDay')
  let $spelExpression := concat("createdDateByDay'='", $date, "'")
  return
    <row>
      <column name="date">{xs:string($date)}</column>
      <column name="count">{ia:count('executionTime', $spelExpression)}</column>

```

```

<column name="min">{ia:min('executionTime', $spelExpression)}</column>
<column name="max">{ia:max('executionTime', $spelExpression)}</column>
<column name="avg">{ia:avg('executionTime', $spelExpression)}</column>
<column name="sum">{ia:sum('licensedVolume', $spelExpression)}</column>
<column name="sumSipAiuCount">{ia:sum('sipAiuCount', $spelExpression)}</column>
<column name="sumCiCount">{ia:sum('ciCount', $spelExpression)}</column>
</row>
}</results>

```

Functions ia:count, ia:min, ia:max, ia:avg, and ia:sum are statistic functions.

The second parameter of a statistic function is the SpEL expression, which is used to filter search result rows.

The following is an example of a simple SpEL expression consisting of one single condition:

```
'applicationName'=='Baseball'
```

When using this SpEL expression in a statistic function, only rows are handled by the function where a column with the name application and value Baseball exists.

In the SpEL expression, both column name and column value are quoted. Supported conditions for SpEL expressions depend on the data type of the column values:

Data type	Supported	Examples
String	Operators == and != Functions: startsWith (case sensitive), beginsWith (not case sensitive), endsWith (not case sensitive), contains (not case sensitive)	startsWith('applicationName', 'Ba') 'applicationName'=='Baseball'
Other data types than String	Operators >, >=, <, <=, ==, !=	'recordCount' < '10'

SpEL expressions can consist of multiple conditions by using logical operators and and or.

For example:

```
'applicationName'=='Baseball' and 'recordCount' < '10'
```

## 6.8.9 Form composition details

This section explains how to work with searches.

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	<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>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	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> <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>
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>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	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>
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>

<b>Form Field</b>	<b>Control Type</b>	<b>Does Field Allow Default Value Configuration?</b>
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>
	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>

Form Field	Control Type	Does Field Allow Default Value Configuration?
	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>
	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>

Form Field	Control Type	Does Field Allow Default Value Configuration?
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>
	Date Range and Date Range (Composite)	<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	<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 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>• XQuery: Yes</li> <li>• External Service URL: Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• Source Select: No</li> <li>• Target Select: 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>• Local: No</li> <li>• UTC: No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• Required: Yes</li> <li>• Out of Range: Yes</li> <li>• Regex Pattern Not Match: 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?
	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?
	Number Range	<p><b>Data Resolution:</b></p> <ul style="list-style-type: none"> <li>• XQuery: No</li> <li>• External Service URL: No</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• Source Select: No</li> <li>• Target Select: 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>• Local: No</li> <li>• UTC: No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• Required: Yes</li> <li>• Out of Range: No</li> <li>• Regex Pattern Not Match: 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>• XQuery: Yes</li> <li>• External Service URL: Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• Source Select: Yes</li> <li>• Target Select: 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>• Local: No</li> <li>• UTC: No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• Required: Yes</li> <li>• Out of Range: No</li> <li>• Regex Pattern Not Match: 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>• XQuery: Yes</li> <li>• External Service URL: Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• Source Select: Yes</li> <li>• Target Select: 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>• Local: No</li> <li>• UTC: No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• Required: Yes</li> <li>• Out of Range: No</li> <li>• Regex Pattern Not Match: 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>• XQuery: Yes</li> <li>• External Service URL: Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• Source Select: No</li> <li>• Target Select: 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>• Local: Yes</li> <li>• UTC: Yes</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• Required: Yes</li> <li>• Out of Range: No</li> <li>• Regex Pattern Not Match: 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>• XQuery: Yes</li> <li>• External Service URL: Yes</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• Source Select: No</li> <li>• Target Select: 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>• Local: Yes</li> <li>• UTC: Yes</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• Required: Yes</li> <li>• Out of Range: No</li> <li>• Regex Pattern Not Match: 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>• XQuery: No</li> <li>• External Service URL: No</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• Source Select: No</li> <li>• Target Select: 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>• Local: No</li> <li>• UTC: No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• Required: No</li> <li>• Out of Range: No</li> <li>• Regex Pattern Not Match: 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>• XQuery: No</li> <li>• External Service URL: No</li> </ul> <p><b>Values from other list selection (+ XQuery):</b></p> <ul style="list-style-type: none"> <li>• Source Select: No</li> <li>• Target Select: 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>• Local: No</li> <li>• UTC: No</li> </ul> <p><b>Error Messages:</b></p> <ul style="list-style-type: none"> <li>• Required: No</li> <li>• Out of Range: No</li> <li>• Regex Pattern Not Match: No</li> </ul>

Form Field	Control Type	Does Field Allow Default Value Configuration?
		<b>Text:</b> No <b>Background/Border Color:</b> Yes <b>Conditional Show:</b> Yes

### 6.8.10 OpenText Intelligent Viewing integration

OpenText Intelligent Viewing) is a third and optional viewer type that can be enabled for a View column when composing search results (see [Configuring view columns](#)). OTIV is a cloud-first product for file viewing and transformation that provides a functionally rich and responsive experience that goes beyond the typical basic file viewing scenario.

Along with the fully licensed version of the product, there is also a complimentary version available to customers. The core product is the same, and it is the presence of a license or not that limits its functionality. In the case of OpenText Information Archive's OTIV integration, the features not available in the free version are:

- The ability to publish to PDF, and
- View CAD or other 3D documents.

To view a list of the supported document, image, and video formats, click here (<https://www.opentext.com/assets/documents/en-US/pdf/opentext-intelligent-viewing-supported-formats-en.pdf>).

For preview mode, OpenText Information Archive uses the HTML's `<object>` tag to render the preview. The `<object>` tag is basically implemented by the browsers. Browsers only support a limited set of MIME types. Some of the MIME types are natively supported (for example, image file formats, some audio formats, and some video formats). Most browsers also support PDFs. Specialized browser plugins are required to support formats such as Microsoft Word/Excel/PowerPoint.

Unlike the Native Browser viewer and OpenText Information Archive Viewer, OTIV is a server-side viewer that is dynamically loaded at runtime and that requires authentication integration with OpenText Directory Services (OTDS).

For integration to occur:

- OpenText Information Archive must be deployed in OTDS SSO mode for authentication and authorization.
- OTIV must be deployed alongside OpenText Information Archive and share the authentication and authorization services provided by OTDS using the OAuth2 tokens (JWT).

For more information about OpenText Information Archive and OpenText Intelligent Viewing integration, see Section 6.3 "Installing OpenText Intelligent

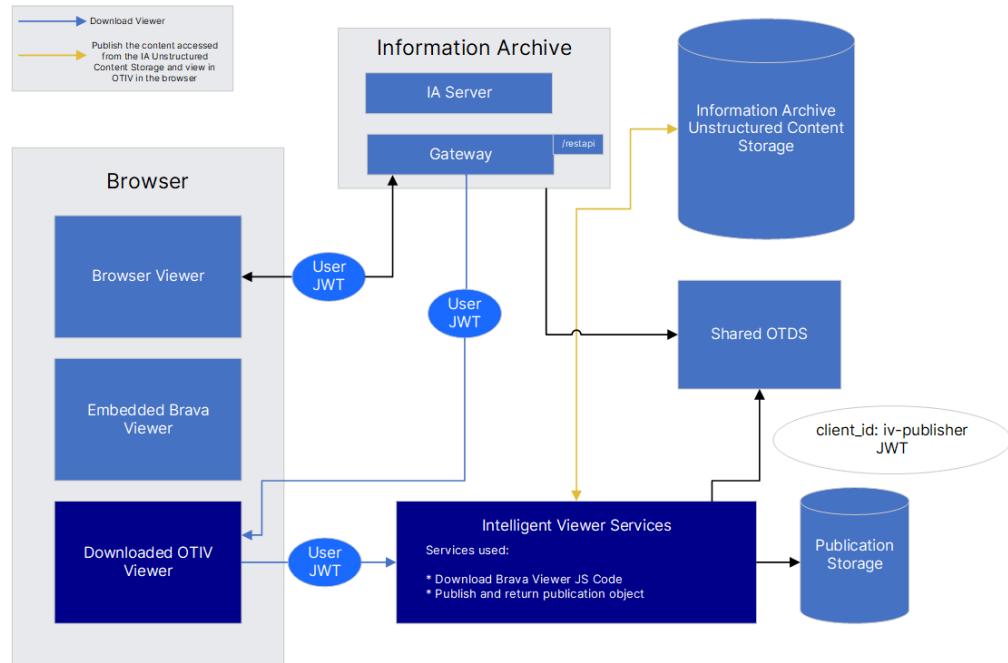
Viewing (OTIV)" in *OpenText Information Archive - Installation Guide (EARCORE-IGD)*.

For OpenText Intelligent Viewing installation instructions:

- For Windows, see Part II "Install and configure" in *OpenText Intelligent Viewing - Windows Install Guide (CLIVSA-IGD)*
- For Linux, see Part II "Installation" in *OpenText Intelligent Viewing - Linux Install Guide (CLIVSA-IGL)*

### 6.8.10.1 OTIV integration architecture

The following diagram illustrates the OTIV integration architecture:



For authentication and authorization, OpenText Information Archive interoperates with OTIV using the JWT tokens provided by OTDS. It makes use of the additional OAuth2 scopes needed to access the OTIV-related scopes (`create_publications`, `view_publications`, `view_any_publication`) in the JWT tokens. The JWT tokens are used when:

- the JavaScript-based OTIV code is loaded into the IA Web App running in the browser
- the document associated with a search result value is published to OTIV in order to perform the rendering and returning of the publication object
- the rendered document is viewed in OTIV

For OTIV integration, when any search form runs for the first time in the user's login session, OpenText Information Archive downloads the client-side code of OTIV and

configures the UI. The Search Developer (DEVELOPER role) can select OTIV (the third option) as the viewer to view documents in the search results list and the inline and details panels. The Search Developer can also configure the MIME type of the rendering that is published using the services provided by OTIV and consumed in OTIV. End Users, Retention Managers, and Search Developers (END USER, RETENTION MANAGER, DEVELOPER, AUDITOR, BUSINESS OWNER roles) can view the documents associated with the search result values.

For more information, see Section 6.3 “Installing OpenText Intelligent Viewing (OTIV)” in *OpenText Information Archive - Installation Guide (EARCORE-IGD)*.

#### 6.8.10.2 Example of using OpenText Intelligent Viewing in OpenText Information Archive

Document transformation to PDF can be done using the **Publish to PDF** option, which is available only for the licensed users in OTIV.

1. In the Edit search dialog, navigate to the **Result Detail** tab and select **Inline Panel**. Click **Select from** and **Schema**.
2. In the **Select Fields** dialog, select an attachment with the **Type of Mime Type** and click **OK**.
3. Edit the attachment field you added in the previous step and configure the following:
  - a. Change **Field Type** to **View Content**.
  - b. Change **Viewer Type** to **Intelligent Viewer**.
  - c. Change **Mime Type** to From a column and then select **Mime Type (Attachments::MimeType)**.
  - d. Ensure the **Preview** checkbox is not selected.
  - e. Click **OK**.
4. Save your changes.
5. Run the updated search. In the results, click the attachment that you added to view the result in the OpenText Intelligent Viewing viewer.



**Note:** The IA Web App triggers audit events for each action performed in OTIV. To view the audit information, run the Archive Audits job and then run the Application Audit search.

Both Print and Publish to PDF events are recorded as Viewer Print event types in the IA Server and they are differentiated by the **publishToPDF** field.

## 6.9 SIP specific search configuration

### 6.9.1 Cross holding

A search can be configured to look at two holdings of data at the same time. With this approach, every AIU returned by the search from each holding represents a single row in search results list (this is the “UNION” operation of AIUs).



**Note:** At this moment it is not possible to have a “JOIN” operation between AIUs from different holdings by an identifier, so that the AIUs from different holdings are presented on the single row at the search result page. The only way to simulate a JOIN is to configure a nested search to display the related information. One limitation with this approach is that the nested results are retrieved on demand and cannot be exported.

There are two ways to create cross-holding search:

1. By creating a plain search for two holdings without the addition of a nested search. Refer to [With a plain search](#) for further information.
2. By creating a primary search for the first holding along with a nested search for the second holding. Refer to [With creating primary and nested searches](#) for further information.

#### 6.9.1.1 With a plain search

To create a cross-holding-search, the following resources must be configured:

- AIC
- Query
- ResultConfigurationHelper
- Order

It is possible to have different names for the AIC and the index name, which is set in the PDI. The query object then defines the two names, one from the AIC and the other one from the PDI. The query has a field named `indexName`. It is filled only when the AIC name is different from the index name set in the PDI configuration. The name used in the `indexName` field then replaces the AIC name (to query the suitable Lucene index).

Refer to the sample in the CertificatesAndTrades application with the Lucene indexes named `CustomerLastName` and `CustomerFirstName`.

The following is a sample from a PDI:

```

pdis:
- name: Trades-pdi
  configure: create or update
  useDeclarativeConfiguration: true
  pdiSchema: urn:eas-samples:en:xsd:trades.1.0
  aiuPath: /{urn:eas-samples:en:xsd:trades.1.0}Trades/{urn:eas-samples:en:xsd:trades.1.0}CustomerLastNames
  aiuPath: /{urn:eas-samples:en:xsd:trades.1.0}Trades/{urn:eas-samples:en:xsd:trades.1.0}CustomerFirstNames
  
```

```

1.0}Trade
  cis:
    - path: /{urn: eas-samples:en:xsd:trades.1.0}Trades/{urn: eas-samples:en:xsd:trades.
1.0}Trade/{urn: eas-samples:en:xsd:trades.1.0}Attachments/{urn: eas-samples:en:xsd:trades.
1.0}Attachment/{urn: eas-samples:en:xsd:trades.1.0}FileName
      mimeTypePath: /{urn: eas-samples:en:xsd:trades.1.0}Trades/{urn: eas-
samples:en:xsd:trades.1.0}Trade/{urn: eas-samples:en:xsd:trades.1.0}Attachments/{urn: eas-
samples:en:xsd:trades.1.0}Attachment/{urn: eas-samples:en:xsd:trades.1.0}MimeType
        compress: false
        encrypt: true
        hashes:
          - encoding: hex
            algorithm: SHA3-224
            enforced: false
    search:
      criteria:
        - name: TradeDate
          path: /{urn: eas-samples:en:xsd:trades.1.0}Trades/{urn: eas-samples:en:xsd:trades.
1.0}Trade/{urn: eas-samples:en:xsd:trades.1.0}TradeDate
          type: DATETIME
          hint: true
          pkeys:
            - name: dateRange01
              func: RANGE
        - name: TradeID
          path: /{urn: eas-samples:en:xsd:trades.1.0}Trades/{urn: eas-samples:en:xsd:trades.
1.0}Trade/{urn: eas-samples:en:xsd:trades.1.0}TradeID
          type: INTEGER
        - name: CustomerLastName
          path: /{urn: eas-samples:en:xsd:trades.1.0}Trades/{urn: eas-samples:en:xsd:trades.
1.0}Trade/{urn: eas-samples:en:xsd:trades.1.0}CustomerLastName
          type: STRING
          fulltext: true
        - name: CustomerFirstName
          path: /{urn: eas-samples:en:xsd:trades.1.0}Trades/{urn: eas-samples:en:xsd:trades.
1.0}Trade/{urn: eas-samples:en:xsd:trades.1.0}CustomerFirstName
          type: STRING
          fulltext: true
        - name: CustomerID
          path: /{urn: eas-samples:en:xsd:trades.1.0}Trades/{urn: eas-samples:en:xsd:trades.
1.0}Trade/{urn: eas-samples:en:xsd:trades.1.0}CustomerID
          type: BIGINTEGER
        - name: Exchange
          path: /{urn: eas-samples:en:xsd:trades.1.0}Trades/{urn: eas-samples:en:xsd:trades.
1.0}Trade/{urn: eas-samples:en:xsd:trades.1.0}Exchange
          type: STRING
        - name: TraderName
          path: /{urn: eas-samples:en:xsd:trades.1.0}Trades/{urn: eas-samples:en:xsd:trades.
1.0}Trade/{urn: eas-samples:en:xsd:trades.1.0}TraderName
          type: STRING
        - name: Ticker
          path: /{urn: eas-samples:en:xsd:trades.1.0}Trades/{urn: eas-samples:en:xsd:trades.
1.0}Trade/{urn: eas-samples:en:xsd:trades.1.0}Ticker
          type: STRING
        - name: Status
          path: /{urn: eas-samples:en:xsd:trades.1.0}Trades/{urn: eas-samples:en:xsd:trades.
1.0}Trade/{urn: eas-samples:en:xsd:trades.1.0}Status
          type: STRING

```

The following is a sample from an AIC:

```

aics:
- name: Certificates+Trades-aic
  criteria:
    - name: FirstName
      label: First Name
    - name: LastName
      label: Last Name
    - name: CreatedDate
      label: Created Date

```

```

pKeyValuesAttr: pkeys.dateTimeRange01
type: datetime
- name: TraderName
  label: Trader Name
- name: CertificateNo
  label: Certificate number
- name: TradeID
  label: Trade ID
  type: hashed
holdings:
- Certificates
- Trades
# CRYPTO
pdiCryptoAccessGroup: ''
pdiCryptoQueryGroup: ''
# END OF CRYPTO
predicate: ''

```

The following is a sample from a query:

```

queries:
- name: Certificates+Trades-query
aics:
- Certificates+Trades-aic
resultSchema: urn:acme-corp:xsd:crossholdingsearch.1.0
libraryPdiConfigs:
- operands:
  - name: FirstName
  - name: LastName
  - name: CreatedDate
    type: DATETIME
  - name: TraderName
  - name: CertificateNo
    schema: urn:acme-corp:xsd:certificates.1.0
- operands:
  - name: FirstName
    indexName: CustomerFirstName
  - name: LastName
    indexName: CustomerLastName
  - name: CreatedDate
    type: DATETIME
  - name: TraderName
  - name: TradeID
    type: HASHED
    schema: urn:eas-samples:en:xsd:trades.1.0

```

Complete the following steps to configure the resources:

1. Prepare a YAML configuration file with all required resource defined. It is important to pay attention to how the ResultConfigurationHelper uses namespaces.

The example of a completed configuration file can be found in the OOTB CertificatedAndTrades application. The file can be located in the following directory: <IA\_ROOT>/examples/applications/CertificatesAndTrades/config/application-config/searches/configuration.yml.

For example, if the required resource was prepared in the cross-holding-search.yml file, as described in Step 1, and placed in the C:/ CrossHoldingDemo folder, for the resource to be installed in OpenText Information Archive, launch IA Shell and run the following command:

```
iashell> connect  
Connected to "http://localhost:8765/services" as sue@iacustomer.com iashell> import  
--from "C:/CrossHoldingDemo/cross-holding-search.yml"
```

The run should pass without issues and the report should contain information about which resources were created.

2. When creating a search in the IA Web App, the AIC and query resources are available for selection for a cross-holding search.

#### 6.9.1.2 With creating primary and nested searches

The following scenario illustrates how to create a cross-holding search by creating primary and nested searches. The search criteria from one search is then bound to the nested search results of the other search. The holdings in question are Certificates-aic and Trades-aic, which are part of the CertificatesAndTrades sample application, located at <IA\_ROOT>/examples/applications/CertificatesAndTrades/config/application-config/searches/configuration.yml.

With the exception of a few attributes, such as Document Date, the schema files for the two holdings are completely different.

1. In the CertificatesAndTrades sample application, click **Create Search**.
2. For the **Start** step of the Create Search wizard, complete the following steps:
  - a. In the **Search Name** field, enter **Primary**.
  - b. From the **Archival Collection** list, select **Certificates-aic**.
  - c. Set the **Type** field to **Primary Search**.
  - d. Click **Next**.
3. For the **Criteria** step of the Create Search wizard, complete the following steps:
  - a. Include the following fields in the search form by selecting them in the **Show In Form** column:
    - **CertificateNo**
    - **CreatedDate**
  - b. Click **Next**.
4. For the **Results** step of the Create Search wizard, complete the following steps:
  - a. Include the following columns in the search results by selecting them in the **Include In Results** column:
    - **CreatedDate**
    - **CustomerFirstName**
    - **CustomerLastName**
    - **CreatedBy**

- **TransRefGUID**
- b. Click **Finish**.
  5. Once you have returned to the Search Forms page, click **Create Search** again.
  6. For the **Start** step of the Create Search wizard, complete the following steps:
    - a. In the **Search Name** field, enter **Nested**.
    - b. From the **Archival Collection** list, select **Trades-aic**.
    - c. Set the **Type** field to **Nested Search**.
    - d. Click **Next**.
  7. For the **Criteria** step of the Create Search wizard, complete the following steps:
    - a. Include the following fields in the search form by selecting them in the **Show In Form** column:
      - **FirstName**
      - **LastName**
      - **Exchange**
      - **Status**
    - b. Click **Next**.
  8. For the **Results** step of the Create Search wizard, complete the following steps:
    - a. Include the following columns in the search results by selecting them in the **Include In Results** column:
      - **TradeID**
      - **CustomerID**
      - **CustomerFirstName**
    - b. Click **Finish**.
- The two forms now appear in the Search Forms tab for the application.
- c. Edit the search named Primary.
  - d. In the **Result List** tab, edit the **Customer First Name** column:
    - i. For the **Column Type** list, select **Linked Column (Nested Search)**.
    - ii. For the **Column Name** field, enter Nested.
    - iii. In the **Nested Search Mapping** section:
      - A. For the **Result Column Binding Name**, select **CustmerFirstName**.
      - B. For the **Search Field Binding Name**, select **FirstName**.
- This binds the nested search criteria (FirstName) to the Primary search's CustmerFirstName column.
- C. Click **OK**.

D. Click **Save**.

9. Run the search that was created and validate that it allows you to return the AIUs from both holdings.

For further information, refer [here on page 176](#) and Section 2.4 “Running a search” in *OpenText Information Archive - End User Guide (EARCORE-UGD)*.

## 6.9.2 Searching for records that are cached out

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.

Here is an example where there is exactly one AIP currently cached out that contains calls before 2008.

- If the customer searches for calls after 2008, since all of the data is available online the search can be made synchronous. If there is a timeout, the customer is given the option to reissue a background request.
- If the customer searches for calls prior to 2008, because it detects that some of the data is offline (an AIP is cached out), the user is prompted and asked if a background search should run instead

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 consider all of the needed AIPs. If some of them are cached-out, it will cache them in.
- The results of the background search are the same.

The background search launched in the previous section is completed even if it was targeting a cached-out AIP.

When clicking **View**, the result contains all the data.

If a user runs a background search, but the search has been deleted, a message is issued when the user tries to view the search results.

## 6.10 Table search configuration

### 6.10.1 Identifying the search criteria data type

Just before OpenText Information Archive runs a search query on PostgreSQL, for each non-empty search parameter, the data type must be set. The data type is determined by using the parameter metadata from the prepared statement (`PreparedStatement.getParameterMetadata()`). The resulting data types are usually correct, but in some rare cases PostgreSQL throws an exception. This exception is often thrown because PostgreSQL cannot determine the data type of a certain parameter. This problem can be solved by adding an explicit cast to the parameter in the query.

For example, the original query of the Order\_Management search Parts Specification is:

```
SELECT IA_ROWID, P_PARTKEY, P_NAME, P_MFGR, P_BRAND, P_TYPE, P_SIZE,
P_CONTAINER, P_RETAILPRICE, P_COMMENT FROM ORDER_MANAGEMENT.PARTS WHERE 1=1%
AND P_MFGR=$partMFGR%% AND P_BRAND=$partBRAND%% AND
P_CONTAINER = concat(concat($pContainerSize, ' '), $pContainerType)%
```

When running the search with a parameter value set for Container Type, the following exception is thrown:

```
org.postgresql.util.PSQLException: ERROR: could not determine data type of parameter $3
```

To solve this problem, add an explicit cast to the third parameter:

```
CAST ( $pContainerSize AS VARCHAR )
```

You will now get an error because the data type of the fourth parameter cannot be determined. After you have added an explicit cast to the fourth parameter, the query runs as expected.

The new query:

```
ELECT IA_ROWID, P_PARTKEY, P_NAME, P_MFGR, P_BRAND, P_TYPE, P_SIZE, P_CONTAINER,
P_RETAILPRICE, P_COMMENT FROM ORDER_MANAGEMENT.PARTS WHERE 1=1% AND
P_MFGR=$partMFGR%% AND P_BRAND=$partBRAND%% AND P_CONTAINER = concat(concat(CAST
( $pContainerSize AS VARCHAR ),
' '), CAST ( $pContainerType AS VARCHAR ))%
```

### 6.10.2 Creating the SQL query

A table search requires an SQL query, which must be written manually. A common SQL query has the following structure:

```
SELECT {SELECT part} FROM ( FROM part) WHERE {WHERE part}
```

The SELECT part contains the columns returned by the query. The column names must match the column names of the metadata. To bind them to OpenText Information Archive result list columns, an alias can be used. If a column name matches the OpenText Information Archive result list column name, ignoring case, no alias is required.

For example, if the Result List column name is lastName and the metadata column name is NAMELAST, you must define this column as shown in the example:

```
SELECT NAMELAST as "lastName" FROM ...
```

Values inserted in the search form are passed as variables to the query. Variable names must match search form names. The variables are used in the conditions of the WHERE part.

If a form field is optional and it does not have a value, the condition with the corresponding variable is removed from the query by OpenText Information Archive before it runs.

To help identify a condition with an optional value, the condition must be surrounded by '%' characters.

For example, if a variable name is lastName, the column name is NAMELAST and the form field is optional, the WHERE part can be written as:

```
WHERE 1=1% AND NAMELAST = $lastName%
```

The 1=1 part is inserted to make sure the query is still valid after removal of all optional conditions.

When a column in the SELECT part is encrypted, it should be prefixed with 'crypto'. OpenText Information Archive will encrypt the resulting column values before storing them in a search result.

When an encrypted column is used in a condition, the prefix should be used in combination with the variable name. This indicates the provided variable value must first be hashed before the SQL query runs.

For example, if an encrypted column name has name NAMEFIRST, corresponding OpenText Information Archive result column name is firstName and form name is firstName the SELECT and WHERE part can be written as:

```
SELECT crypto:NAMEFIRST as "firstName" FROM .... WHERE $crypto:firstName = NAMEFIRST
```

The FROM part contains the table name, including table joins if multiple tables are involved.

Table names are expected to include the schema name.

The SELECT part must always contain column IA\_ROWID. This IA\_ROWID is used internally for multiple purposes, for instance to uniquely identify a search result item. If this IA\_ROWID is missing, an error will be shown.

The following is an example of a complete query with two optional conditions:

```
SELECT IA_ROWID, PLAYERID, BIRTHYEAR, BIRTHMONTH, BIRTHDAY, NAMELAST as "lastName",
crypto:NAMEFIRST as "firstName" FROM BASEBALL.MASTER WHERE 1=1% AND NAMEFIRST =
$crypto:firstName%% AND NAMELAST = $lastName%
```

OrderManagement includes full-text queries on the FACTSHEETS table joined with the PARTS table on the CID field. Here is an example

```

SELECT PARTS.IA_ROWID, PARTS.P_PARTKEY, PARTS.P_NAME, PARTS.P_MFGR, PARTS.P_BRAND,
PARTS.P_TYPE, PARTS.P_SIZE,
PARTS.P_CONTAINER, PARTS.P_RETAILPRICE, PARTS.P_COMMENT
    %, ts_headline('english', FACTSHEETS.FACTSHEET_TEXT, to_tsquery('english',
array_to_string(regexp_split_to_array
($partFACTSHEET, E'\\s+'), '|')), 'MaxFragments=10, MaxWords=7, MinWords=3,
FragmentDelimiter="...",
StartSel="<b>", StopSel="</b>", HighlightAll=false, ShortWord=0') AS CID_SNIPPETS%
    , PARTS.P_FACTSHEET AS CID
    FROM ORDER_MANAGEMENT.PARTS
    % INNER JOIN ORDER_MANAGEMENT.FACTSHEETS ON PARTS.P_FACTSHEET = FACTSHEETS.CID AND
to_tsvector
('english', FACTSHEETS.FACTSHEET_TEXT) @@ plainto_tsquery('english', $partFACTSHEET)%
    WHERE 1=1
    % AND P_MFGR=$partMFGR%
    % AND P_BRAND=$partBRAND%
    % AND P_CONTAINER = concat(concat(CAST ( $pContainerSize AS VARCHAR ), ' '), CAST
( $pContainerType AS VARCHAR ))%
    ORDER BY PARTS.P_PARTKEY

```

It selects some columns from the PARTS table and (optionally) the LANGUAGE column of the FACTSHEETS table, and also returns a constructed column, which contains a list of snippets found in its FACTSHEET\_TEXT column (see the PostgreSQL documentation to learn more about the `ts_headline` function). The provided predicates include a full-text one:

```

AND to_tsvector('english', FACTSHEETS.FACTSHEET_TEXT) @@ to_tsquery('english',
$partFACTSHEET)

```

This means that it wants to look for FACTSHEET\_TEXT column values matching what the user has provided in the \$partFACTSHEET search form field.



**Note:** In order to hit the corresponding GIN index on the FACTSHEETS table, the predicate needs to be written in this way (basically, the predicate expression must match the GIN expression).



### Caution

A SQL query is expected to return search results where each search result item has a unique IA\_ROWID. It is possible to create a query in which multiple records with the same IA\_ROWID are returned, but this is not recommended. There are a number of known issues with duplicate IA\_ROWIDs:

1. When selecting one of the search result items in the search result list, all other search result items with the same IA\_ROWID are selected.
2. When exporting the search results, attached content may be missing.
3. When using the search result to apply retention, the search result number of the original search may not match the number of records where retention is applied.

### 6.10.2.1 Creating a multi-language SQL query

This is a continuation of [Storing multi-language extracted text](#) and [Multi-language custom indexes](#).

If your search supports search by language, you would need to define a SQL query per language, as your query must use the column name with the proper language extension. You could create a search configuration per language, which requires not only a separate SQL query, but also a ResultMaster and an XForm, as well. This leads to a lot of duplication.

To avoid this duplication, OpenText Information Archive supports SQL queries that can be parametrized by language. For this a search parameter with the name `language` can be used.

Use a language prefix to indicate a column name must be replaced by the language-specific column name before the query is executed. For this purpose, OpenText Information Archive uses the value of the `$language/code` parameter.

To parametrize a query, existing queries might rely SQL functions like the following example:

 **Example 6-1: Legacy SQL function**

```
CREATE FUNCTION ORDER_MANAGEMENT.getSearchConfig (lang text) RETURNS regconfig as $$  
SELECT CASE lang  
WHEN 'en' THEN ('english')::regconfig  
WHEN 'de' THEN ('german')::regconfig  
WHEN 'fr' THEN ('french')::regconfig  
WHEN 'nl' THEN ('dutch')::regconfig  
WHEN 'es' THEN ('spanish')::regconfig  
WHEN 'it' THEN ('italian')::regconfig  
ELSE ('simple')::regconfig END; $$ language sql immutable;
```

 These legacy functions, however, have been replaced by using `$language/name` instead.

In previous versions of OpenText Information Archive that supported multi-language applications, the language dropdown widget only submitted the language code using the `$language` variable. Any SQL queries that also needed the corresponding language name were forced to rely on a custom SQL function to convert the language code to the language name for subsequent use.

This language dropdown widget has now been updated to submit both language code and language name so that, within the SQL query, these can be referenced through `$language/code` and `$language/name`, respectively, and the use of this custom SQL function is no longer needed.

Existing multi-language SQL queries can be updated accordingly by replacing all references to the `$language` variable to `$language/code`. Additionally, invocations of the SQL function to convert language code to language name can now simply be replaced with `$language/name`.

For instance, if you have a full text condition in a query such as the following:

```
to_tsvector('english', FACTSHEETS.FACTSHEET_TEXT) @@  
to_tsquery('english', $partFACTSHEET)
```

Make this condition work for all languages by rewriting the condition to:

```
to_tsvector($language/name, language:FACTSHEETS.FACTSHEET_TEXT) @@  
to_tsquery($language/name, $partFACTSHEET)
```

However, for backwards compatibility reasons, the original \$language variable continues to provide the language code as before so existing SQL queries should continue to work unchanged, as well.

The example application OrderManagementMultilanguage contains examples of SQL queries that can be used with multiple languages.

### 6.10.3 Editing the SQL query for record lookups by ID

SQL query configuration is used by the system to run table application searches. Occasionally, when restoring a saved search, the system needs to look up individual records by their record IDs, and display the result like a regular search result. Along with each SQL query for a search composition, it is recommended that you provide a second SQL query, equivalent to the original in all aspects except that it selects the records by record IDs, not by search criteria.

This ID query is used when reloading saved searches and when expanding retention sets, hold sets, and purge lists. For more information about saved searches, see Section 2.7.2 “Creating a saved search” in *OpenText Information Archive - End User Guide (EARCORE-UGD)*. Learn more about retention sets, hold sets, and purge lists in the [Performing compliance tasks](#) chapter.

The following is an example, taken from the Baseball example application's Name Search. The search query is this:

```
SELECT IA_ROWID, LAST_ACTIVITY_DATE as "lastactivity", MILEAGE_BALANCE as "balance",  
MEMBERSHIP_STATUS as "status", CUST_ID as "customerId",  
MEMBER_SINCE_DATE as "membersince", LAST_NAME as "lastName", crypto:FIRST_NAME as  
"firstName" FROM TICKETS.CUSTOMER WHERE 1=1%  
AND FIRST_NAME = $crypto:firstName%% AND LAST_NAME = $lastName%
```

Its corresponding ID query is this:

```
SELECT IA_ROWID, LAST_ACTIVITY_DATE as "lastactivity", MILEAGE_BALANCE as "balance",  
MEMBERSHIP_STATUS as "status", CUST_ID as "customerId",  
MEMBER_SINCE_DATE as "membersince", LAST_NAME as "lastName", crypto:FIRST_NAME as  
"firstName" FROM TICKETS.CUSTOMER WHERE 1=1%  
AND IA_ROWID = any($ia_rowid)%
```

Essentially, we took the original query from the Query Editor and, in the ID Query Editor, updated the WHERE clause to the applicable ROWID.



**Tip:** Review the ID Query Editor tab for the example table applications, as they have the ID queries set up to demonstrate the feature.



**Note:** The system does not validate the query so test the SQL query is valid.

If you do not provide an ID query along with your search query, the system still makes a best effort to reload an associated saved search, or show an expanded retention set, hold set or purge list. However, there are limitations, and the saved search reload may fail, or the expanded view may look bad or show only record IDs.

Columns representing content or links will not be shown in expanded retention sets, hold sets or purge lists.

## 6.11 Setting the debug options for a search

A Developer can set debug options for a particular search, which enable the logging of search events for analysis. Debug options are useful in the following situations:

- *Search creation:* When creating the query portion of a search, a Developer can test and modify the query to optimize its use of indexes.
- *Troubleshooting:* If a search in production has become too slow (for example, when the size of data has grown substantially), a Developer can investigate whether the search is optimized. The Developer can turn on the debug options, ask an End User affected by the search performance to run the search, and then analyze the logs to identify a fix.

**!** **Important**

Debug options should only be set temporarily because they can affect search performance due to excessive logging output.

The debug settings are ignored by searches done by jobs such as the Apply Retention Rule to Records job.

When you enable debugging for a search, debug tracing is enabled. Debug options can be activated both in composition mode and from the **Search Listings** page.

Table searches are run by using SQL only, while SIP searches are partly run by using SQL and partly by Lucene. Therefore, table searches support only Debug Search SQL Phase, while SIP searches support both Debug Search SQL Phase and Debug Search Lucene Phase.

When selecting one or more options for the Debug Search SQL Phase, you must also select the checkbox next to **Debug Search SQL Phase**.

When debug is in play, a **Logging** tag is displayed on the **Search Listings** page, as well as in search composition.

---

### SQL Debug Options

Analyze/ analyze	Carry out the command and show the actual run times.
---------------------	------------------------------------------------------

<b>Verbose/verbose</b>	Displays additional information regarding the plan specifically, include the output column list for each node in the plan tree, schema-qualify table and function names, always label variables in expressions with their range table alias, and always print the name of each trigger for which statistics are displayed.
<b>Costs/costs</b>	Include information on the estimated startup and total cost of each plan node, as well as the estimated number of rows and the estimated width of each row.
<b>Buffers/buffers</b>	Include information on buffer usage, specifically include the number of shared blocks hits, reads and writes, the number of local blocks hits, reads and writes, and the number of temp blocks reads and writes, Shared blocks, local blocks and temp blocks contain tables and indexes, temporary tables and temporary indexes, and disk blocks used in sort and materialized plans, respectively. The number of blocks shown for an upper-level node includes those used by all its child nodes. In text format, only non-zero values are printed. This parameter may only be used with the ANALYZE parameter.
<b>Timing/timing</b>	Include actual startup time and time spent in each node in the output. The overhead of repeatedly reading the system clock can slow down the query significantly on some systems, so it may be useful to set this parameter to FALSE when only actual row counts, and not exact times, are needed. Run time of the entire statement is always measured, even when node-level timing is turned off with this option. This parameter may only be used when ANALYZE is also enabled.
<b>Summary/summary</b>	Include summary information (e.g., totaled timing information) after the query plan. Summary information is included by default when ANALYZE is used but otherwise is not included by default, but can be enabled using this option. Planning time in EXPLAIN EXECUTE includes the time required to fetch the plan from the cache and the time required for re-planning, if necessary.
<b>Format/format</b>	Choose the format of the debug output: TEXT, XML JSON or YML.

### Lucene Debug Options

<b>Query/query</b>	Display the structure of the original and the rewritten Lucene Query Tree.
--------------------	----------------------------------------------------------------------------

You can set the debug options for a search using either IA Shell or IA Web App.

For more information about setting the debug options for a search in IA Shell, see Section 1.7 “Logging” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*.

When you set debug options for a search in IA Web App, a **Logging** label appears next to the name of the search and on the search result page.

**To set the debug options using IA Web App:**

1. To set debug options during search creation, in the IA Web App, when you are composing the search, click the  icon next to the name of the search, and then click **Debug Options**.  
To set debug options for an existing search, in the IA Web App, go to the **Applications** tab and click the name of the application. Next to the search that you want to debug, click the context menu and select **Debug Options**.
2. Set the debug options that you want and click **Save**.
3. Run the search as a background request.
4. When the background request has completed, view the debug output by doing one of the following:
  - In IA Web App, on the **Background Requests** tab, click the **Completed** link for the search, and then click **Download diagnostic logs**.
  - In a text editor, open the <IA\_ROOT>/logs/iaserver/xquery-debug.log file.
5. Remove the debug options when they are no longer required.

## 6.12 Adding user context in custom search forms and search result presentations

Some items in a custom search form or custom search result presentation may be based on the current login user context. OpenText Information Archive exposes the user context, including the user name, groups, roles and language for customized search forms and result presentations.

### 6.12.1 Injecting user context XForm instance to a custom XForm

The following illustrates how to inject a user context XForm instance to custom XForm when rendering it at run time. The user context XForm instance has the following structure:

```
<instance xmlns="http://www.w3.org/2002/xforms" id="context">
  <context xmlns="">
    <userName>...</userName>
    <groupNames>
      <groupName>...</groupName>
      ...
    </groupNames>
    <roleNames>
      <roleName>...</roleName>
      ...
    </roleNames>
    <language>...</language>
  </context>
</instance>
```

## 6.12.2 Accessing user context in a custom XForm

To access the user context in a custom XForm, refer to the user context instance node value in the custom XForm XML. For example, define the following XForms elements in XForms model part:

```
<xforms:action ev:event="xforms-ready">
  <xforms:message>
    Hello, <xforms:output value="instance('context')/userName"/>
    with language <xforms:output value="instance('context')/language"/> !
  </xforms:message>
</xforms:action>
```

Login as sue@iacustomer.com and open the custom search form, a popup message dialog is displayed that includes the user name and current language value.

## 6.12.3 Adding user context APIs to IA Custom Presentation Service

Four APIs can be used to expose user information in IACustomPresentationService:

- `getUserName()`
- `getGroupNames()`
- `getRoleNames()`
- `getLanguage()`

## 6.12.4 Accessing user context in a custom presentation

In custom presentation of Angular JS, ingest the IACustomPresentationService to its controller, then call one of the APIs mentioned in the [previous section](#) to access the user context. For example:

```
angular.module('Search').registerController('CustomController',
['IACustomPresentationService', '$scope', '$timeout',
  function(IACustomPresentationService, $scope, $timeout) {
  var cc = this;

  cc.userName = IACustomPresentationService.getUserName();
  cc.groupNames = IACustomPresentationService.getGroupNames();
  cc.roleNames = IACustomPresentationService.getRoleNames();
  cc.language = IACustomPresentationService.getLanguage();

  ...
})();
```

## 6.13 Search troubleshooting

**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:**

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 [step 2](#) above).
- Try to reduce the amount of result columns (the more columns in the result list, the slower the search will be because of [step 2](#)).
- Review load balancing topology.
- Review external IT factors.

**SIP search is slow:**

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. OpenText Information Archive search is a two-tiered search. The first tier selects a subset of AIPs based on the defined partitioning keys. The second tier runs 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 runs 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 is 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.
- Enable DEBUG log level for all components (services level and UI level) and repeat the search. Find the trace that looks like the following in either the `ia.log` file or in the specific log associated with the background search:

```
2025-06-04 11:15:11.457 [47] [12c5d038-6f90-4b73-8b53-4c1c01e29fc7] INFO
search
    - SearchComposition ('PRIMARY')
'68715e99-bb8e-4041-b338-03c35db15f5b' has been executed in 4201 ms. 38
records
have been written in the cache. 10 records have been returned.
{"executionTimePhases" : [369,3758,0,74], "subPhasesDetails" :
{ "splitAndSort"
: 256, "cacheInit" : 4, "subQueries" : 2508, "cacheIndexation" : 615 },
"maxThreadNumber" : 2, "subQueryCount" : 10, "threads" : [{ "thread" : 85,
"subQueries" : [[725,1226],[28,96],[27,99],[33,99],[41,118]]},{ "thread" : 86,
"subQueries" : [[650,1251],[32,95],[35,86],[36,89],[33,105]]}],
"threadQuerySum" : 1640, "threadCacheSum" : 3264, "searchCompositionId" :
"68715e99-bb8e-4041-b338-03c35db15f5b", "executionTime": 4201,
"recordCacheCount": 38, "aipCount" : 10, "aiuCount" : 38, "async" : "false",
"libCount" : "10", "stats" : 0}
```

This trace provides details and timing information from the search execution. This information helps pinpoint the part of application configuration that needs to be improved. The following table provides more information about each property in the JSON structure provided by the trace. Most of the data are also available (under the same name unless stated otherwise) in the supplemental data of the audit generated by the search:

Property	Available in search audit	Multi-threaded only	Description
executionTimePhases	Yes	No	<p>Breakdown array of the durations in milliseconds of the four main steps of the SIP search in order:</p> <ol style="list-style-type: none"> <li>1. (<code>executionTime::sdpg</code> in <code>Search audit's supplemental data</code>): Duration of the AIP's selection in the system database.</li> <li>2. (<code>executionTime::query</code>): Duration of the AIU's selection from Lucene indexes. In a multi-threaded search: duration of the AIU's selection + writing of the results. Due to the nature of a multi-threaded search, separating both is not possible.</li> <li>3. (<code>executionTime::cache</code>): Duration taken to write the result to the <code>SearchResult</code> Lucene index. Always equal to 0 for a multi-threaded search (see the comment about multi-threaded searches in number 2).</li> <li>4. (<code>executionTime::page</code>)</li> </ol>

Property	Available in search audit	Multi-threaded only	Description
			Duration of the pagination (delivery of the results to the client). Always equal to 0 for background searches, as the delivery is done when consulting the background search and not during the execution.

Property	Available in search audit	Multi-threaded only	Description
subPhasesDetails	No	No	<p>Additional details about execution times in milliseconds for sub tasks from the AIU selections and Writing the results phases:</p> <ul style="list-style-type: none"> <li>- splitAndSort: Time passed to sort AIPs selected and split them by Lucene codecs and maximum number of libraries by split.</li> <li>- cacheInit: Time elapsed to create the search result library in case of a multi-threaded search (will always be 0 for single threaded search).</li> <li>- subQueries: Time passed to execute the Lucene queries on the AIUs.</li> <li>- cacheIndexation: Time taken to finalizing the Lucene index containing the search results. In a multi-threaded search: time to consolidate the results of all threads into one Lucene index.</li> </ul>

Property	Available in search audit	Multi-threaded only	Description
maxThreadNumber	Yes	Yes	Maximum number of parallel threads used to run this search.
subQueryCount	Yes	Yes	Total number of sub-queries executed during the search.
threads	No	Yes	Breakdown of subqueries execution runtime grouped by thread: <ul style="list-style-type: none"> <li>- <b>thread:</b> ID of the search thread.</li> <li>- <b>subQueries:</b> Breakdown array of the durations in milliseconds of steps of each sub-query executed by the search thread: <ul style="list-style-type: none"> <li>o Duration of the AIU's selection from Lucene indexes.</li> <li>o Duration taken to write the result to the SearchResult 's Lucene index.</li> </ul> </li> </ul>

Property	Available in search audit	Multi-threaded only	Description
threadQuerySum	Yes	Yes	Total time passed for the AIU's queries (sum of the first values in the previous arrays). This is the cumulative effort of each thread, so it could be superior to the total query time as threads are executed in parallel.
threadCacheSum	Yes	Yes	Total time passed writing the results in SearchResult's Lucene index (sum the second values in the previous arrays).
searchCompositionId	Yes	No	The ID of the searchComposition object used to run the search.
executionTime	Yes	No	Total execution time of the search.
recordCacheCount	Yes	No	Number of records written in the search result (number of results).
aipCount	Yes	No	Number of AIPs selected in the first phase of the search.
aiuCount	Yes	No	Number of AIUs in the scope of the search after the selection of AIPs.
async	Yes	No	Whether the search is a background search or not.
libCount	Yes	No	Number of Lucene libraries targeted by the AIU's query.
stats	No	No	Time passed on updating the search statistics.

- The AIPs are offline so it takes longer to bring them online.

#### Expression in unstructured content is not found during a search:

If a word or expression is not found in an unstructured content, it might be due to the fact that the extraction was not "as expected". When using the on premise deployment only, it is possible to use a script to directly extract a document and verify if the extraction is correct. The extraction is done using KView. Sample scripts are available at <IA\_ROOT>\lib\iaserver\keyview\javaapi\sample. To perform the extraction from one file to another, the FilterFileToFile script can be used (available in Linux or Windows). To learn how to set up the samples, review the documentation at <IA\_ROOT>\lib\iaserver\keyview\Guides\html\Filter\_24.1.0\_JavaProgramming\index.html in the Sample Programs chapter.

#### If the following message is found in the ia.logs file:

```
2020-02-11 14:24:05.283 [55] [a05b6e00-3252-47cf-b57d-215163a6b1fa]
WARN c.e.i.s.search.results.Column- Conflicting values for column Search By
Player Name/Decrypted First Name/Side Panel/Tab 1/debut/dataType: 'STRING'
vs 'DATETIME'
```

This message indicates that the search defined a column that does not match the underlying data model value. The simplest solution is to remove the column from the panel (the error message indicates that the side panel and tab, and then select from schema so that the type correctly matches the underlying data type).

For the example above, a developer should edit the **Search By Player Name** search to ensure that the correct set is selected. The name of the set in this case is **Decrypted First Name**. The developer would have to:

1. Click the **Result Detail** tab.
2. Remove the debut field.
3. Add the column back using the **Select from schema** button. This ensures that the datatype for the column aligns with expected value.



## Chapter 7

# Configuring the export of search results

## 7.1 Introduction

The OpenText Information Archive export mechanism enables export workflows 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 content.

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 default section, one row of data (selected row) is available for the export operation. From the main search result page, 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.

This mechanism is driven by one configuration component “Export Configuration”. The export configuration can be added at the tenant or application levels. A number of tenant export configurations are available out-of-the-box. Refer to [Default export configurations](#) for more information.

The OpenText Information Archive export mechanism enables export workflows 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 content. A number of common export configurations are available out-of-the-box.

Previously, the export mechanism of OpenText Information Archive used YAML files that were updated in IA Shell to configure how search results were exported. Now, the export mechanism relies on a CRUD operation to configure how search results are exported. Refer to [Creating export configurations at the tenant and application levels](#) for more information.

Three distinct export scenarios are supported:

This mechanism is driven by one configuration component “Export Configuration” to support three distinct export scenarios.

- Default: Exporting records based on search configuration (ResultMaster)
- XSLT: Exporting records based on XSL-T/XSL-FO transformation
- Custom: Exporting records based on custom JAVA implementation

For each export scenario, it is possible to decide the compression/packaging format, the input/output format, to include or not unstructured contents, the background priority, the record limit and synchronous export support.

### Export configuration

Field name	Type	Comment	Example
name	String	Internal name	
label	String	Name displayed	
description	String	Description	
tenant	Tenant Ref	Export configurations with tenant are shared by all applications	
application	Application Ref	Export configurations with application are available only for this application	
engine	Enum	Export framework implementation	Default (default), XProc
priority	Integer	Background request priority	100 by default
mimeType	String	MIME type returned with synchronous export	application/octet-stream by default
contentDisposition	Enum	Content Disposition returned with synchronous export	Inline, Attachment
sizeThreshold	Long	Max records support for instant download	50 by default
compression	Enum	Compression format	None, Zip, Gzip, Tar
inputFormat	Enum	Input format used during the export	ROW_COLUMN, ROW_COLUMN_WITH_RAW_XML
outputFormat	Enum	Output format after the transformation	CSV, TEXT, XML, JSON, HTML, PDF
includesContent	Boolean	Include or not unstructured contents	False by default
transformationType	Enum	Export scenario	Default, XSLT, Custom
transformationContent	String	XSLT/FO content	

Field name	Type	Comment	Example
recordLimit	Long	Maximum limit of records	0 by default (unlimited)
customExportHandler	String	Full name of custom handler implementation	com.acme.ia.extension.export.CustomSearchResultExportHandler
options	Map<String, String>	Options that could leverage by custom handler	
encryptedOptions	Map<String, String>	Sensible options that could leverage by custom handler	
deliveryType	Enum	Indicates whether the custom handler implementation delivers files by itself (NONE) or not (DEFAULT).	DEFAULT, NONE

➤ **Example 7-1: Example of declarative configuration**

```

tenant:
  name: INFOARCHIVE

exportConfigurations:
  - name: zip_csv
    label: Records Only as CSV (ZIP)
    description: zip envelope for csv export
    application: null
    engine: DEFAULT
    contentDisposition: ATTACHMENT
    mimeType: application/zip
    includesContent: false
    inputFormat: ROW_COLUMN
    outputFormat: CSV
    compression: ZIP
    sizeThreshold: 50
    tenant: INFOARCHIVE
    transformationType: DEFAULT
    recordLimit: 1000000
  
```



! **Important**

When selecting the outputFormat field, the PDF format causes performance issues if the number of results is too large. If you anticipate a large number of results, use a different format, such as XML or CVS.

## 7.1.1 Creating export configurations at the tenant and application levels

To configure how search results are exported at the tenant level, access the **Administration > Export Configurations** tab.

To configure how an application's search results are exported, access the **Export Configurations** tab in the desired application.

Existing export configurations, whether configured at the tenant or application level, appear in a table that includes:

- **Name:** The name of the configuration.
- The  button allows you to perform the following actions:
  - **Duplicate:** This allows you to duplicate the selected export configuration. Enter a **Name**, **Description** and click **OK**. The export configuration can then be edited and redesigned for its new purpose. For more information, refer to the procedure below about editing an export configuration.
  - **Delete:** If an export configuration has not been applied to a search, you are able to delete it from the list. When prompted whether you want the configuration to be deleted, click **Delete**.
- **Label:** Information that describes how the configuration exports the search results.
- The **In Use** column indicates whether each configuration has been applied to at least one search form.

Selecting a configuration's row allows you to see more information about each configuration in the inline panel.

Use the **Find Export Configuration** field to locate a specific export configuration.

### To create an export configuration:

1. Click the **+** (Create Export Configuration) button.
2. In the **Summary** tab, enter the following:
  - a. **Name:** The name of the configuration.
  - b. **Label:** Enter information that describes how the configuration exports the search results.
  - c. **Optional Description:** Enter a description of the export configuration.
  - d. **Input Format:** Select the input format to be used during the export:
    - **ROW\_COLUMN**
    - **ROW\_COLUMN\_WITH\_RAW\_XML**

- e. **Output Format:** Select the output format to be used after the transformation.
    - **Optional** Select the **Include Content** box if the export is to include unstructured content.
  - f. **Optional Mime Type:** Enter the MIME type to be returned with the synchronous export.
  - g. **Transformation Type:** Select the export scenario (default, XSLT, or custom).
3. The **Transformation** tab allows you to configure the export to perform a transformation based on XSLT. Refer to [Writing CUSTOM implementation to export search results](#) for more information.



**Tip:** To review an example of a transformation configuration:

1. Access the **Export Configurations** tab in the Baseball application.
2. Select the **searchByPlayerNameDetails** configuration and click the  button.
3. Navigate to the **Transformation** tab.
4. **Optional** The **Delivery** tab, Enter the following:
  - a. **Optional Compression:** Select the compression format for the export configuration (None, ZIP, GZIP, or TAR).
  - b. **Optional Record Limit:** Enter the maximum limit of records that can be returned in a synchronous export. By default, the value is 0.
  - c. **Optional Priority:** Enter the background request priority for the export configuration.
  - d. **Optional Size Threshold:** Enter the maximum number of records the export configuration will support for instant downloads
  - e. **Optional Content Disposition:** Select whether a synchronous export will be returned inline or with an attachment.
5. **Optional** The **Options** tab allows you to enter options that could be leveraged by the custom handler. Click **+ Add Option** and provide a **Key** and **Value**.  
Multiple options can be added by clicking **+ Add Option** again. However, if multiple options are being applied to the export configuration, the value entered in the **Key** field must be unique.
6. When done, click **Create**. The export configuration you created appears in the table in the **Export Configurations** tab.

#### To edit an export configuration:

1. Select the export configuration being edited and click the  button.

2. Make the desired changes to the export configuration.
3. When done, click **Save**.

### 7.1.2 Default export configurations

A number of common export configurations are available out-of-the-box. The list can be modified by editing the `first-time-setup/applications/Tenant/config/exports/configuration.yml` file.

```
gzip_json
  label: Records only as JSON (GZIP)
  engine: DEFAULT
  transformationType: DEFAULT
  inputFormat: ROW_COLUMN
  outputFormat: JSON
  includesContent: false
  compression: GZIP
  sizeThreshold: 50
  mimeType: application/gzip

gzip_json_with_content
  label: Records with Content as JSON (GZIP)
  engine: DEFAULT
  transformationType: DEFAULT
  inputFormat: ROW_COLUMN
  outputFormat: JSON
  includesContent: true
  compression: GZIP
  sizeThreshold: 50
  mimeType: application/gzip

zip_csv
  label: Records only as CSV (ZIP)
  engine: DEFAULT
  transformationType: DEFAULT
  inputFormat: ROW_COLUMN
  outputFormat: CSV
  includesContent: false
  compression: ZIP
  sizeThreshold: 50
```

```
mimeType: application/zip

zip_csv_with_content
    label: Records with Content as CSV (ZIP)
    engine: DEFAULT
    transformationType: DEFAULT
    inputFormat: ROW_COLUMN
    outputFormat: CSV
    includesContent: true
    compression: ZIP
    sizeThreshold: 50
    mimeType: application/zip

zip_raw_xml
    label: Records only as RAW XML (ZIP)
    engine: DEFAULT
    transformationType: DEFAULT
    inputFormat: ROW_COLUMN_WITH_RAW_XML
    outputFormat: XML
    includesContent: false
    compression: ZIP
    sizeThreshold: 50
    mimeType: application/zip

zip_raw_xml_with_content
    label: Records with Content as RAW XML (ZIP)
    engine: DEFAULT
    transformationType: DEFAULT
    inputFormat: ROW_COLUMN_WITH_RAW_XML
    outputFormat: XML
    includesContent: true
    compression: ZIP
    sizeThreshold: 50
    mimeType: application/zip

pdf
    label: Records only as PDF
    engine: DEFAULT
    transformationType: XSLT
    inputFormat: ROW_COLUMN
```

```
        outputFormat: PDF
        includesContent: false
        compression: NONE
        sizeThreshold: 50
        mimeType: application/pdf

zip_pdf_with_content
    label: Records only as PDF
    engine: DEFAULT
    transformationType: XSLT
    inputFormat: ROW_COLUMN
    outputFormat: PDF
    includesContent: false
    compression: NONE
    sizeThreshold: 50
    mimeType: application/pdf

zip_xml
    label: Records only as XML (ZIP)
    engine: DEFAULT
    transformationType: DEFAULT
    inputFormat: ROW_COLUMN
    outputFormat: XML
    includesContent: false
    compression: ZIP
    sizeThreshold: 50
    mimeType: application/zip

zip_xml_with_content
    label: Records with Content as XML (ZIP)
    engine: DEFAULT
    transformationType: DEFAULT
    inputFormat: ROW_COLUMN
    outputFormat: XML
    includesContent: true
    compression: ZIP
    sizeThreshold: 50
    mimeType: application/zip
```

---

All export configurations have the option `includeCompliance` that allows you to export compliance information. The following table describes information that will be exported depending on the option value.

<b>Compliance Property</b>	<b>includeCompliance</b>		<b>Description</b>
	<b>min</b>	<b>all</b>	
compliance_projectedDispositionDate	X	X	The date on which record is eligible for disposition.
compliance_underRetention		X	Boolean indicating whether a retention policy has been applied on a record or its parent (SIP, table, or application).
compliance_directlyUnderRetention		X	Boolean indicating whether a hold has been applied on record.
compliance_underHold	X	X	Boolean indicating whether a hold has been applied on a record or its parent (SIP, table, or application).
compliance_directlyUnderHold		X	Boolean indicating whether a hold has been applied on a record.
compliance_retentionPolicyNames		X	List of retention policy names (separated by comma).
compliance_holdNames		X	List of hold names (separated by comma).
compliance_createdDate		X	The date when compliance was applied to the record.

These properties are omitted from the export when their values are empty or set to false.

## 7.2 Writing XSLT to export search results

For a complex use case, it is possible to configure the export to perform a transformation based on XSLT. This transformation can be performed against ROW COLUMN or RAW XML input formats. The XSLT is set directly on the export configuration component with the transformationContent field. XSLT templates are available in the page displaying an export configuration to facilitate creation of new transformation content.

### 7.2.1 Default collection

To support a large volume of data, the transformation must be performed from a stream. This stream can be obtained from the default collection (collection()) and iterated with a for-each statement. The identity transformation is like the XSLT below:

```
<xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output indent="yes" method="xml" encoding="utf-8"/>
  <xsl:template match="/">
    <results>
      <xsl:for-each select="collection()">
        <xsl:copy-of select=". />
      </xsl:for-each>
    </results>
  </xsl:template>
</xsl:stylesheet>
```

#### Contextual information

During the transformation, contextual information (user name, application name, user groups, columns) is accessible from three distinct collections.

- **Context (collection('uri:context'))**

```
<context>
  <userName>sue@iacustomer.com</userName>
  <onBehalfOfUser/>
  <searchName>Date_Operator</searchName>
  <searchCompositionName>Set 1</searchCompositionName>
  <exportConfigurationName>raw_xml_pdf</exportConfigurationName>
  <exportConfigurationOptions>
    <option name="key1" value="value1"/>
  </exportConfigurationOptions>
  <applicationName>PhoneCalls</applicationName>
</context>
```

- **User Groups (collection('uri:user-groups'))**

```
<userGroups>
  <userGroup>GROUP_RETENTION_MANAGER</userGroup>
  <userGroup>GROUP_END_USER</userGroup>
  <userGroup>GROUP_DEVELOPER</userGroup>
  <userGroup>GROUP_IT_OWNER</userGroup>
  <userGroup>GROUP_BUSINESS_OWNER</userGroup>
  <userGroup>GROUP_ADMINISTRATOR</userGroup>
</userGroups>
```

- **Columns (collection('uri:columns'))**

```
<columns>
  <column name="id" label="ID"/>
```

```

<column name="CustomerID" label="Customer ID"/>
<column name="CustomerFirstName" label="First name"/>
<column name="CustomerLastName" label="Last name"/>
<column name="CallFromPhoneNumber" label="Call from"/>
<column name="CallToPhoneNumber" label="Call to"/>
<column name="CallStartDate" label="Call start"/>
<column name="CallEndDate" label="Call end"/>
<column name="Attachments::AttachmentName"
        label="Attachment Name"
        groupName="Attachments"
        groupLabel="Attachments"/>
<column name="Attachments::cid"
        label="Cid"
        groupName="Attachments"
        groupLabel="Attachments"/>
<column name="Attachments::CreatedBy"
        label="Created By"
        groupName="Attachments"
        groupLabel="Attachments"/>
<column name="Attachments::CreatedOnDate"
        label="Created On Date"
        groupName="Attachments"
        groupLabel="Attachments"/>
<column name="Attachments::FileName"
        label="File Name"
        groupName="Attachments"
        groupLabel="Attachments"/>
<column name="SentToArchiveDate" label="Sent to"/>
<column name="RepresentativeID" label="Representative ID"/>

```

The example below transforms a row column to CSV based on the column's contextual information:

```

<xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="text" encoding="utf-8"/>
    <xsl:param name="delim" select="',' "/>
    <xsl:param name="quote" select="''''"/>
    <xsl:param name="break" select="'
    '"/>
    <xsl:param name="columns" as="element()" select="collection('uri:columns')/columns"/>

    <xsl:template match="/">
        <!-- CSV Header -->
        <xsl:call-template name="header"/>
        <!-- CSV Rows -->
        <xsl:for-each select="collection()">
            <xsl:apply-templates select="$columns/column[not(@groupName)]" mode="row">
                <xsl:with-param name="row" select=". "/>
            </xsl:apply-templates>
            <xsl:value-of select="$break"/>
        </xsl:for-each>
    </xsl:template>

    <xsl:template name="header">
        <xsl:apply-templates select="$columns/column[not(@groupName)]" mode="header"/>
        <xsl:value-of select="$break"/>
    </xsl:template>

    <xsl:template match="column" mode="header">
        <xsl:value-of select="concat($quote, @label, $quote)" />
        <xsl:if test="following-sibling::*">
            <xsl:value-of select="$delim"/>
        </xsl:if>
    </xsl:template>

    <xsl:template match="column" mode="row">
        <xsl:param name="row"/>
        <xsl:variable name="name" select="@name"/>
        <xsl:value-of select="concat($quote, $row/column[@name=$name], $quote)" />
    </xsl:template>

```

```
<xsl:if test="following-sibling::*">
    <xsl:value-of select="$delim"/>
</xsl:if>
</xsl:template>

</xsl:stylesheet>
```

## 7.2.2 Get-Compliance

During the transformation, it is possible to call the function Get-Compliance by passing the record ID as a parameter to retrieve compliance information, such as `compliance_projectedDispositionDate`, `compliance_underRetention`, `compliance_directlyUnderRetention`, `compliance_underHold`, `compliance_directlyUnderHold`, `compliance_retentionPolicyNames`, `compliance_holdNames` and `compliance_createdDate`.

**!** **Important**

Customers creating searches for table names must not use column names that start with the `compliance_` prefix, as this naming convention is reserved for compliance fields only.

```
<xsl:stylesheet version="3.0" xmlns:ia-func="urn:x-emc:ia:functions" xmlns:xsl="http://
www.w3.org/1999/XSL/Transform">
    <xsl:output indent="yes" method="xml" encoding="utf-8"/>
    <xsl:template match="/">
        <results>
            <xsl:for-each select="collection()">
                <xsl:variable name="compliance" select="ia-func:get-compliance(./@id)/
compliance"/>
                <record id="./@id">
                    projectedDispositionDate="{$compliance/compliance_projectedDispositionDate}"
                    underRetention="{$compliance/compliance_underRetention}"
                    directlyUnderRetention="{$compliance/compliance_directlyUnderRetention}"
                    underHold="{$compliance/compliance_underHold}"
                    directlyUnderHold="{$compliance/compliance_directlyUnderHold}"
                    retentionPolicyNames="{$compliance/compliance_retentionPolicyNames}"
                    holdNames="{$compliance/compliance_holdNames}"
                    createdDate="{$compliance/compliance_createdDate}">
                    <xsl:copy-of select="*"/>
                </record>
            </xsl:for-each>
        </results>
    </xsl:template>
</xsl:stylesheet>
```

## 7.2.3 PDF support

**!** **Important**

When output format is PDF, the transformation is followed by a rendering step with Apache FOP.

In this case, the XSLT must generate a XSL-FO format. When no XSLT content is provided a default XSLT is used. To limit memory footprint, it's highly recommended to create one page-sequence per record to force Apache FOP to flush the result and to not keep all the document in memory. We also recommend to not display the total pages number for the same reason.

Base-14 fonts are natively supported, and custom TrueType fonts (TTF) are supported by adding font files in specific folder:

- When font files are deployed under <IA\_ROOT>/lib/iaserver/external/fonts/embedded/, fonts are embedded in generated PDF.
- When font files are deployed under <IA\_ROOT>/lib/iaserver/external/fonts/, fonts are only referenced in generated PDF. In this case, it is important to be in control of the target environment where the produced document is consumed (*i.e.*, the necessary fonts have to be installed there).

When the font is not specified or is missing, the font is mapped to the Base-14 font Times. If no glyph can be found for a given character, FOP will issue a warning and use the glyph for # (if available) instead.

### Default PDF XSLT

```
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:fo="http://www.w3.org/1999/XSL/
Format">
  <xsl:output indent="no" method="xml" encoding="utf-8"/>
  <xsl:attribute-set name="h2">
    <xsl:attribute name="font-family">Helvetica</xsl:attribute>
    <xsl:attribute name="font-size">12pt</xsl:attribute>
    <xsl:attribute name="font-weight">bold</xsl:attribute>
    <xsl:attribute name="text-align">left</xsl:attribute>
  </xsl:attribute-set>
  <xsl:attribute-set name="table">
    <xsl:attribute name="font-family">Helvetica</xsl:attribute>
    <xsl:attribute name="font-size">10pt</xsl:attribute>
    <xsl:attribute name="text-align">left</xsl:attribute>
    <xsl:attribute name="font-weight">normal</xsl:attribute>
    <xsl:attribute name="table-layout">fixed</xsl:attribute>
    <xsl:attribute name="border-collapse">collapse</xsl:attribute>
    <xsl:attribute name="width">17cm</xsl:attribute>
  </xsl:attribute-set>
  <xsl:attribute-set name="label">
    <xsl:attribute name="font-family">Helvetica</xsl:attribute>
    <xsl:attribute name="font-size">10pt</xsl:attribute>
    <xsl:attribute name="text-align">left</xsl:attribute>
    <xsl:attribute name="font-weight">bold</xsl:attribute>
    <xsl:attribute name="padding-top">0.2cm</xsl:attribute>
    <xsl:attribute name="column-width">4cm</xsl:attribute>
  </xsl:attribute-set>
  <xsl:attribute-set name="value">
    <xsl:attribute name="font-family">Helvetica</xsl:attribute>
    <xsl:attribute name="font-size">10pt</xsl:attribute>
    <xsl:attribute name="text-align">left</xsl:attribute>
    <xsl:attribute name="font-weight">normal</xsl:attribute>
    <xsl:attribute name="padding-top">0.2cm</xsl:attribute>
    <xsl:attribute name="padding-left">0.2cm</xsl:attribute>
    <xsl:attribute name="column-width">13cm</xsl:attribute>
  </xsl:attribute-set>
  <xsl:attribute-set name="page">
    <xsl:attribute name="margin-right">2cm</xsl:attribute>
    <xsl:attribute name="margin-left">2cm</xsl:attribute>
    <xsl:attribute name="margin-bottom">1cm</xsl:attribute>
    <xsl:attribute name="margin-top">1cm</xsl:attribute>
    <xsl:attribute name="font-family">Helvetica</xsl:attribute>
    <xsl:attribute name="page-height">29.7cm</xsl:attribute>
    <xsl:attribute name="page-width">21cm</xsl:attribute>
  </xsl:attribute-set>
  <xsl:template match="/">
    <xsl:variable name="columns" as="element()" select="collection('uri:columns')/
columns"/>
```

```
<xsl:variable name="context" as="element()" select="collection('uri:context')/context"/>
<!--
<xsl:variable name="includeCompliance" select="$context/exportConfigurationOptions/
option[@name='includeCompliance'][1]/@value"/>
-->
<fo:root>
<fo:layout-master-set>
<fo:simple-page-master xsl:use-attribute-sets="page" master-name="main">
<fo:region-body margin-bottom="1.5cm" margin-top="1.5cm"/>
<fo:region-before extent="1.5cm"/>
<fo:region-after extent="1.5cm"/>
</fo:simple-page-master>
</fo:layout-master-set>
<!-- use default collection to stream the records -->
<xsl:for-each select="collection()">
<!-- create one page-sequence per record to limit memory footprint -->
<fo:page-sequence master-reference="main">
<fo:static-content flow-name="xsl-region-before">
<xsl:call-template name="header"/>
</fo:static-content>
<fo:static-content flow-name="xsl-region-after">
<xsl:call-template name="footer">
<xsl:with-param name="applicationName" select="$context/
applicationName"/>
<xsl:with-param name="searchName" select="$context/searchName"/>
</xsl:call-template>
</fo:static-content>
<fo:flow flow-name="xsl-region-body">
<xsl:call-template name="table">
<xsl:with-param name="columns" select="$columns" as="element()"/>
</xsl:call-template>
</fo:flow>
</fo:page-sequence>
</xsl:for-each>
</fo:root>
</xsl:template>

<xsl:template name="header">
<fo:block>
<fo:block xsl:use-attribute-sets="h2">Record #<xsl:value-of select="position()"/>
</fo:block>
<fo:leader leader-pattern="rule" leader-length="100%"/>
</fo:block>
</xsl:template>

<xsl:template name="footer">
<xsl:param name="applicationName"/>
<xsl:param name="searchName"/>
<fo:block>
<fo:leader leader-pattern="rule" leader-length="100%"/>
<fo:block text-align-last="justify">
<fo:block>
<xsl:value-of select="$applicationName"/> > <xsl:value-of
select="$searchName"/>
<fo:leader leader-pattern="space"/>
<fo:page-number/>
</fo:block>
</fo:block>
</fo:block>
</xsl:template>

<xsl:template name="table">
<xsl:param name="columns" as="element()"/>
<fo:table xsl:use-attribute-sets="table">
<fo:table-column xsl:use-attribute-sets="label"/>
<fo:table-column xsl:use-attribute-sets="value"/>
<fo:table-body>
<xsl:apply-templates select="column">
<xsl:with-param name="columns" select="$columns" as="element()"/>
```

```

        </xsl:apply-templates>
        </fo:table-body>
    </fo:table>
</xsl:template>

<xsl:template match="column">
    <xsl:param name="columns" as="element()" />
    <xsl:variable name="name" select="@name" />
    <xsl:choose>
        <xsl:when test="exists(row)">
            <xsl:apply-templates select="row" mode="nested">
                <xsl:with-param name="columns" select="$columns" as="element()" />
                <xsl:with-param name="columnName" select="$name" as="xs:string" />
            </xsl:apply-templates>
        </xsl:when>
        <xsl:otherwise>
            <fo:table-row>
                <fo:table-cell xsl:use-attribute-sets="label">
                    <fo:block-container overflow="hidden">
                        <fo:block>
                            <xsl:value-of select="$columns/column[@name=$name][1]/@label" />
                        </fo:block>
                    </fo:block-container>
                </fo:table-cell>
                <fo:table-cell xsl:use-attribute-sets="value">
                    <fo:block>
                        <xsl:value-of select="." />
                    </fo:block>
                </fo:table-cell>
            </fo:table-row>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

<xsl:template match="row" mode="nested">
    <xsl:param name="columns" as="element()" />
    <xsl:param name="columnName" as="xs:string" />
    <fo:table-row>
        <fo:table-cell xsl:use-attribute-sets="label">
            <fo:block-container overflow="hidden">
                <fo:block>
                    <xsl:value-of select="$columns/column[@groupName=$columnName][1]/
@groupLabel" />
                </fo:block>
            </fo:block-container>
        </fo:table-cell>
        <fo:table-cell xsl:use-attribute-sets="value">
            <fo:block>
                </fo:block>
            </fo:table-cell>
        </fo:table-row>
    <xsl:apply-templates select="column" mode="nested">
        <xsl:with-param name="columns" select="$columns" as="element()" />
    </xsl:apply-templates>
</xsl:template>

<xsl:template match="column" mode="nested">
    <xsl:param name="columns" as="element()" />
    <xsl:variable name="name" select="@name" />
    <fo:table-row>
        <fo:table-cell xsl:use-attribute-sets="label" padding-left="0.5cm">
            <fo:block>
                <xsl:value-of select="$columns/column[@name=$name][1]/@label" />
            </fo:block>
        </fo:table-cell>
        <fo:table-cell xsl:use-attribute-sets="value">
            <fo:block>
                <xsl:value-of select="." />
            </fo:block>
        </fo:table-cell>
    </fo:table-row>
</xsl:template>

```

```
</xsl:template>
</xsl:stylesheet>
```

The default PDF XSLT rendering compliance information is available in the `export_default_pdf.xsl` file located in the `doc` directory.

## 7.3 Writing CUSTOM implementation to export search results

When DEFAULT and XSLT transformation mode cannot cover one export/delivery use case, it is possible to define a custom logic. To do that, it is necessary to write a class that implements the public interface `com.emc.ia.export.handler`.

`SearchResultExportHandler`. This class must be declared with the field `customExportHandler`, as illustrated below:

```
exportConfiguration:
  name: custom-export-configuration
  transformationType: CUSTOM
  customExportHandler: com.acme.ia.extension.export.CustomSearchResultExportHandler
  ...
```

### SearchResultExportHandler public API

The interface can be found at `$IA_HOME/lib/iaserver/infoarchive-public-api-xx.x.jar`.

`com.emc.ia.export.handler.SearchReusltExportHandler.java`

```
/*
 * Copyright (c) OpenText Corporation. All Rights Reserved.
 * OpenText Confidential: Restricted Internal Distribution
 */
package com.emc.ia.export.handler;

import java.io.IOException;
import java.nio.file.Path;
import java.util.Iterator;
import java.util.List;

public interface SearchResultExportHandler {

    void setContext(Context context);

    List<Path> doExport(Iterator<? extends ResultRow> nodes, ExportType exportType)
        throws IOException;

    enum InputFormat {
        ROW_COLUMN,
        ROW_COLUMN_WITH_RAW_XML
    }

    enum OutputFormat {
        XML,
        CSV,
        HTML,
        TEXT,
        PDF,
        JSON,
        PARQUET,
        AVRO,
        BINARY
    }
}
```

```

enum Compression {
    NONE,
    ZIP,
    GZIP,
    TAR
}

enum ExportType {
    SYNCHRONOUS,
    ASYNCHRONOUS
}

enum DeliveryType {
    DEFAULT,
    NONE
}
}

```

**com.emc.ia.export.handler.Context**

```

/*
 * Copyright (c) OpenText Corporation. All Rights Reserved.
 * OpenText Confidential: Restricted Internal Distribution
 */
package com.emc.ia.export.handler;

import java.io.IOException;
import java.io.InputStream;
import java.nio.file.Path;
import java.util.Map;
import java.util.Set;
import java.util.UUID;

import org.slf4j.Logger;
import org.w3c.dom.Node;

import com.emc.ia.export.handler.SearchResultExportHandler.Compression;
import com.emc.ia.export.handler.SearchResultExportHandler.InputFormat;
import com.emc.ia.export.handler.SearchResultExportHandler.OutputFormat;

public interface Context {

    String getApplicationName();

    String getSearchName();

    String getSearchCompositionName();

    String getExportConfigurationName();

    InputFormat getInputFormat();

    OutputFormat getOutputFormat();

    Compression getCompression();

    String getContentDisposition();

    String getMimeType();

    SearchResultExportHandler.DeliveryType getDeliveryType();

    Map<String, String> getExportConfigurationOptions();

    Map<String, String> getExportConfigurationEncryptedOptions();

    boolean withCis();

    String getTransformationContent();
}

```

```
long getRecordLimit();
Logger getLogger(Class<?> clazz);
Path getWorkingDir();
Path getContentWorkingDir();
UUID getSearchResultId();
UUID getOrderItemId();
String getUserName();
Set<String> getUserGroups();
String getOnBehalfOfUser();
String getColumnLabel(String columnName);
String getColumnDataType(String columnName);
Path downloadCi(String cid) throws IOException;
InputStream readCi(String cid) throws IOException;
String decryptValue(String string);
Node getRawXml(ResultRow resultRow);

/**
 * Returning various compliance information about record
(compliance_projectedDispositionDate (OffsetDateTime),
compliance_underRetention (Boolean), compliance_directlyUnderRetention,
compliance_underHold (Boolean),
compliance_directlyUnderHold (Boolean), compliance_holdNames (Set),
compliance_retentionPolicyNames (Set),
compliance_createdDate (OffsetDateTime)).
 * @param recordId record identifier
 * @return compliance information
 */
Map<String, Object> getCompliance(String recordId);

void setRecordCount(long count);
void setExportSize(long size);
void setPartial(boolean partial);
boolean isCanceled();
}
```

### More details about SearchResultExportHandler

Code source and configuration can be found into the distribution: <IA\_ROOT>/extensions/iaserver/custom-export-configuration.

```
version: 1.0.0

tenant:
  name: INFOARCHIVE
  configure: use existing

exportConfiguration:
  name: ftp-export-configuration
  label: FTP Export Configuration
  description: Export Search Result to CSV through FTP
  application: null
```

```

tenant: INFOARCHIVE
engine: DEFAULT
contentDisposition: ATTACHMENT
mimeType: text/csv
includesContent: false
inputFormat: ROW_COLUMN
outputFormat: CSV
compression: NONE
sizeThreshold: 50
transformationType: CUSTOM
customExportHandler: com.emc.ia.extension.export.FtpSearchResultExportHandler
encryptedOptions:
    user: ${ftp.username}
    host: ${ftp.host:localhost}
    port: ${ftp.port:21}
    password: ${ftp.password}

```

The following is a Java code example:

```

/*
 * Copyright (c) OpenText Corporation. All Rights Reserved.
 * OpenText Confidential: Restricted Internal Distribution
 */
package com.emc.ia.extension.export;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.StringWriter;
import java.io.Writer;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.time.OffsetDateTime;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.function.Function;
import java.util.regex.Pattern;

import org.apache.commons.csv.CSVFormat;
import org.apache.commons.csv.CSVPrinter;
import org.apache.commons.io.IOUtils;
import org.apache.commons.net.ftp.FTP;
import org.apache.commons.net.ftp.FTPClient;

import com.emc.ia.export.handler.Context;
import com.emc.ia.export.handler.ResultColumn;
import com.emc.ia.export.handler.ResultRow;
import com.emc.ia.export.handler.SearchResultExportHandler;

public class FtpSearchResultExportHandler implements SearchResultExportHandler {

    private static final DateTimeFormatter FILE_DATE_TIME_FORMATTER =
        DateTimeFormatter.ofPattern("yyyyMMddHHmmssSSS");

    private static final Pattern FILENAME_PATTERN = Pattern.compile("[^a-zA-Z0-9-_.]");

    private static final int CHECK_CANCEL_STATUS_EVERY = 1000;

    private FTPClient ftpClient;
    private Context context;
    private Path outputPath;

    @Override
    public void setContext(Context context) {

```

```
        this.context = context;
    }

    @Override
    public List<Path> doExport(Iterator<? extends ResultRow> rows,
                               SearchResultExportHandler.ExportType exportType) throws IOException {

        try {
            // Init output path and FTP Client
            init();
            // Write CSV
            writeCsv(rows);
            // Upload CSV
            upload();
        } finally {
            if (ftpClient.isConnected()) {
                ftpClient.disconnect();
            }
        }

        return Collections.emptyList();
    }

    private void upload() throws IOException {
        String fileName = outputPath.getFileName().toString();
        try (OutputStream outputStream = ftpClient.storeFileStream(fileName);
             InputStream inputStream = Files.newInputStream(outputPath)) {
            IOUtils.copy(inputStream, outputStream);
        }
        if (!ftpClient.completePendingCommand()) {
            throw new IOException(String.format("Problem uploading file %s", fileName));
        }
        context.getLogger(this.getClass()).info("The file '{}' was successfully delivered through FTP",
  fileName);
    }

    private void writeCsv(Iterator<? extends ResultRow> rows) throws IOException {
        try (OutputStream outputStream = Files.newOutputStream(
             outputPath); CSVPrinter csvWriter = getCSVWriter(
             new OutputStreamWriter(outputStream, StandardCharsets.UTF_8))) {
            boolean first = true;

            long recordCount = 0L;
            long recordLimit = context.getRecordLimit();
            boolean noLimit = recordLimit <= 0;
            while (rows.hasNext() && (noLimit || recordCount < recordLimit) && (recordCount % CHECK_CANCEL_STATUS_EVERY != 0 || !context.isCanceled())) {
                ResultRow resultRow = rows.next();
                writeResultRow(csvWriter, resultRow, first);
                first = false;
                recordCount++;
            }
            context.setRecordCount(recordCount);
            context.setPartial(rows.hasNext());
        }
        context.setExportSize(outputPath.toFile().length());
    }

    private void writeResultRow(CSVPrinter csvWriter, ResultRow resultRow, boolean header)
        throws IOException {
        if (header) {
            writeResultRow(csvWriter, resultRow, column ->
context.getColumnLabel(column.getName()));
        }
        writeResultRow(csvWriter, resultRow, this::getResultSetValue);
    }

    private void writeResultRow(CSVPrinter writer, ResultRow resultRow,
```

```

        Function<ResultColumn, String> valueFunction) throws IOException {
    List<String> cells = new ArrayList<>();

    resultRow.getColumns().stream().filter(ResultColumn::isExportable).forEach(resultColumn -> {
        String value = valueFunction.apply(resultColumn);
        cells.add(value == null ? "" : value);
    });

    writer.printRecord(cells);
}

private String getResultColumnValue(ResultColumn column) {
    if (column.isComposite()) {
        StringWriter stringWriter = new StringWriter();
        try (CSVPrinter writer = getCSVWriter(stringWriter)) {
            for (ResultRow resultRow : column.getRows()) {
                writeResultRow(writer, resultRow, this::getResultColumnValue);
            }
        } catch (IOException e) {
            // Never actually thrown, but just in case
            throw new RuntimeException(e);
        }
        return stringWriter.toString();
    }

    if (column.isEncrypted()) {
        return context.decryptValue(column.getValue());
    } else {
        return column.getValue();
    }
}

private void init() throws IOException {
    // Define CSV path
    Path workingPath = context.getWorkingDir();
    String outputFileName = String.format("%s-%s-%s.csv", context.getSearchResultId(),
   context.getSearchName(), OffsetDateTime.now().format(FILE_DATE_TIME_FORMATTER));
    this.outputPath =
    Files.createFile(workingPath.resolve(toSafeFileName(outputFileName)));

    // Init FTP Client
    this.ftpClient = new FTPClient();
    Map<String, String> encryptedOptions =
    context.getExportConfigurationEncryptedOptions();
    String user = encryptedOptions.get("user");
    String host = encryptedOptions.get("host");
    String port = encryptedOptions.get("port");
    String pwd = encryptedOptions.get("password");
    try {
        ftpClient.connect(host, Integer.parseInt(port));
        boolean isLoggedIn = ftpClient.login(user, pwd);
        if (isLoggedIn) {
            ftpClient.enterLocalPassiveMode();
            ftpClient.setFileType(FTP.BINARY_FILE_TYPE);
            ftpClient.makeDirectory("export");
            if (!ftpClient.changeWorkingDirectory("export")) {
                throw new RuntimeException("Cannot change the working directory to "
   + ftpClient.printWorkingDirectory() + "export");
            }
        } else {
            throw new RuntimeException(String.format("Wrong credentials for user %s.", user));
        }
    } catch (IOException e) {
        throw new RuntimeException("Failed to initialize the Ftp Client for " + host + ':' +
        port, e);
    }
}

private static String toSafeFileName(String name) {
}

```

```
        return FILENAME_PATTERN.matcher(name).replaceAll("_").replaceAll("__+", "_").trim();
    }

    private static CSVPrinter getCSVWriter(Writer writer) throws IOException {
        return new CSVPrinter(writer, CSVFormat.DEFAULT.builder().build());
    }
}
```

## Chapter 8

# Declarative configuration

## 8.1 Introduction

Up to this point, configuration of storage, applications, searches and other configurable elements, has been explained mainly through the IA Web App interface. However, once you are familiar with IA concepts, you will most likely prefer to configure such things using a textual format, with all your configuration in one or a set of human-readable files, that you can edit and upload, and also obtain by exporting existing storages, applications, searches and more.

OpenText Information Archive provides this through declarative configuration (DC), a YAML-based format for configuration of all OpenText Information Archive elements.

The Declarative Configuration format for OpenText Information Archive is explained in detail on the GitHub website.

## 8.2 YAML-based format

DC is presented as set of YAML (<http://www.yaml.org/spec/1.2/spec.html>)-based files that:

- Make it easier to set up an application.
- Support migration between environments (for example, migrating from the staging to production environment).

The YAML-based declarative configuration files are easier to use, as you simply state what you want, not how to get there. 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.

When manually creating content in YAML files to define configuration objects, it is your responsibility to explicitly specify any references to required resources.

## 8.3 DC variants

### Single YAML file

DC can be presented as a single YAML file that describes a full list of configuration objects, including all resources and types.

A single YAML file is convenient and recommended when:

- You are working with a single resource up to ten resources.
- It is required to have a single container for all a resource's types.
- It is required to install or update a single resource in OpenText Information Archive.

This type of configuration has its own advantages and disadvantages:

---

#### Advantages

- The single YAML file is easy to operate on local disk.
- Easy to edit references and resources with any text editor.

---

#### Disadvantages

- The YAML file looks like a huge container for all of a resource's types.
- It does not allow for a quick understanding of what resources the single YAML file contains.
- There is a limit on the size of the YAML file. Splitting the file into different files helps you stay below that limit.

---

### Module structure of YAML files

While a single YAML file is difficult to work with when there are a lot of resources of different types, then there is a possibility to create a module structure of nested YAML files. This can be achieved with `include` statement. Every module is represented with its own YAML file and can refer to any number of other relevant and required YAML files.

Structured YAML files by module are convenient and recommended to work with when creating a complicated configuration that consists of more than 10 resources of different types which need to be separated by users with different roles.

When using a DC module structure, it is recommended to split modules in the following structure:

---

### Administration Module

The module contains all of the resources related to the definition of storage systems, databases, and encryption objects. The module corresponds to the resources that are configured by an Administrator.

---

### Compliance Module

The module contains all of the resources related to the definition of retention policies, holds and rules. The module corresponds to the resources that are configured by a Retention Manager.

---

### Application Module

The module contains all of the resources related to the definition of applications. The module corresponds to the resources that are configured by a Developer.



**Note:** This module contains a reference to the administration module and compliance module.

---

The following are the advantages and disadvantages of the DC module structure:

---

#### Advantages

- The structure provides a decent overview of existing parts of the configuration.
- Every module corresponds to a particular user role.
- Every module is optional and independent (except for reference names).
- The format is used when exporting configuration from OpenText Information Archive.
- The format is used in sample applications.

---

#### Disadvantage

- There are several files of local disks in a nested folder structure, which has disadvantages. For example, it is more difficult to navigate to a specific file or copy and paste information into a particular file.

---

### 8.3.1 ZIP archive

To simplify working with configurations that consist of several YAML files (or modules), then a ZIP archive with configuration is supported. ZIP archives are used for uploading/downloading the configurations over the REST.

## 8.4 Import DC configuration

OpenText Information Archive allows the opportunity to import declarative configuration that contains any resource definition. Even a single resource can be defined in a DC (YAML) file and imported to the system.

The import operation can run with any DC configuration variant:

- Single YAML file,
- Folder with a set of nested and structured YAML files,
- ZIP archive with YAML files.

When importing the configuration, the resources defined in YAML files 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 it is possible to call the import configuration functionality in OpenText Information Archive:

---

### REST API Server

OpenText Information Archive provided REST API that can be consumed by any REST client. Refer to *OpenText Information Archive REST API Developer's Guide* on support.opentext.com (<https://support.opentext.com/>) for more information. REST accepts only single YAML files or ZIP archive configuration formats.

---

### IA Shell

The import commands in IA Shell allow for flexible configuration import. There is a possibility to use all possible DC formats and scopes for configuration (for more information, see Section 2.10.7 "import" in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)* and Section 2.10.8 "import-search" in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*).

---

### IA Web App

User can import configuration using the IA Web App from the application listing page, search listing page, and holding listing page. The only format for import is ZIP, as it the most convenient format. If more flexible functionality is required, use IA Shell or the REST API.

Regardless of where the configuration has been imported, a notification message informs the user about the created/skipped resources. It is worth preserving this information, at least temporarily, during configuration debugging on the development environment.



**Note:** Import of configuration is done under the user with a specified role (Administrator/Developer/Retention Manager). It is important to remember that a particular role has access only its specific resources. If the role does not have access to a resource, then an “Access Denied” error is reported during the import operation. There is a superuser role (sue@iacustomer.com), however, that can import all the resources from all the categories in one shot.

## 8.5 Export DC configuration

### 8.5.1 Export levels

OpenText Information Archive provides the capability to export the configuration of:

---

#### System

The configuration level that allows the exportation of all existing tenants.

---

#### Tenant

The configuration level that allows the exportation of all existing applications under that tenant.

---

#### Application

The configuration level that allows the exportation of all existing holdings under that application and other relevant application configuration resources.

---

#### Holdings

The configuration level for a particular holding to export.

---

#### Searches

The configuration level for a particular search to export.

### 8.5.2 Export content and formats

The number of artifacts inside the exported configuration is different and depends on two factors:

- The export-level (refer to [Export levels](#) for more information).
- The property that is passed to export: `create`, `use existing`, `create or update`, and `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.

Export can be done into one of the following formats (refer to [DC configuration variants](#) for more information):

- In either a ZIP archive with module structure of YAML files; or
- A single YAML format.

When exporting to a ZIP format, the archive contains a set of configuration resources in YAML format. It includes the following module(s) and related information:

- **Administration**

- Data nodes, databases
  - Various storage, credentials and encryption objects

- **Compliance**

- Retentions, holds and rules

- **Application**

- Data model files

For SIP applications, this includes the holding that specifies the schema for the records, but not the records themselves.

For table applications, this exports the table database, table schema, and tables, but none of the records.

- Search definitions
  - Result list custom presentations
  - Result list export definition



**Note:** When exporting to ZIP, the type of content being exported depends on the user's role. This is only applicable from when done at the application level. For example:

- An Administrator only exports an Administration module.
- A Retention Manager only exports a Compliance module.
- A Developer only exports application-specific resources.
- A super user exports all of the modules.

Holds are not exported when exporting an application

When exporting to a single YAML file format, the exported file does not contain dedicated resources or module files but, instead, the application, administration and compliance configurations are presented as one large YAML file.



**Note:** The application state is not listed in the exported file.

### 8.5.3 Security export aspects

For security reasons, an exported configuration never contains credentials. It may contain system-specific settings that vary from system to system, and must be updated before you import it into another system. Prior to importing exported configuration to another environment, be sure to:

- Update system-related settings and credentials in the `administration-config/configuration.yml` file.
- Update references to administration resources in the `application-config/configuration.yml` file.

### 8.5.4 Export API

The following list outlines the places where to call the export configuration functionality in OpenText Information Archive:

---

#### REST API Server

OpenText Information Archive provided REST API that can be consumed by any REST client. Refer to *OpenText Information Archive REST API Developer's Guide* on support.opentext.com (<https://support.opentext.com/>) for more information.

---

#### IA Shell

The 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 (for more information, see Section 2.10.5 "export" in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)* and Section 2.10.6 "export-search" in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*).

---

#### IA Web App

User can export an application, search, and holding. There are the restrictions from the user interface on export for applications and holdings: only create or update the scope of the 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 export is ZIP, as it the most convenient format. If more flexible functionality is required, use IA Shell or the REST API.



**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 incorrect.

## 8.6 Logging

When operating with DC (either export or import), the actions are logged in a dedicated log file located (<IA\_ROOT>/logs/iaserver/dc.log). This log file helps you to investigate and analyze the activity. If an exception or error has occurred, it is logged in both log files: (<IA\_ROOT>/logs/iaserver/dc.log) and (<IA\_ROOT>/logs/iaserver/errors.log).



**Tip:** If there is an error or exception, view both log files in DEBUG mode.

## 8.7 Supporting old 'S3Glacier' declarative configurations

Previously, there used to be four store types related to Amazon S3:

- S3
- S3PROXY
- S3GLACIER
- S3GLACIERPROXY

Currently, however, only the S3 and S3PROXY store types are supported. While the S3GLACIER and S3GLACIERPROXY configurations (storage system and store) are loading, the system converts S3GLACIER to S3 and S3GLACIERPROXY converts to S3PROXY, respectively. This way, the system is completing the migration/transformation of old S3GLACIER and S3GLACIERPROXY configurations to S3 or S3PROXY. Therefore, the same UI handles both S3/S3PROXY configurations, as well as old S3GLACIER/S3GLACIERPROXY configurations.

As of this release, however, there is only one store type (S3).

## 8.8 Example applications

As this is probably best learned by example, all of our example applications are configured using declarative configuration format for you to browse through, edit and upload to test different variations. This chapter explains certain aspects in a bit more detail.

The sample declarative configuration applications can be found in the <IA\_ROOT>\examples\applications\<APPLICATION\_NAME> directory.

Every application consists of three parts:

- The applications configuration files in YAML (<http://www.yaml.org/spec/1.2/spec.html>) format, which is exactly called declarative configuration.
- The sample data (SIPs or tables) to ingest into an application after installation.

- The IA Shell installation script that contains a list of commands that need to run for application installation and data ingestion.

### 8.8.1 Example application declarative configurations

The declarative configuration files can be found in the <IA\_ROOT> \examples \applications\<APPLICATION\_NAME>\config directory.

The folder consists of three optional modules and one main `configuration.yml` file that joins the modules together with the *include* statement.

The modules are related to the following resource types:

- Administration resources
- Compliance resources
- Application resources

In turn, every module contains its own `configuration.yml` file with metadata describing the configuration objects. Certain content is put in external files that is referenced from the YAML files to make that content easier to edit.

Every 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 configuration under that folder is installed with IA Shell.

### 8.8.2 Locating OpenText Information Archive applications outside of the examples folder structure

You might not want to locate your OpenText Information Archive application in the <IA\_ROOT>/examples folder structure. For example, its data might be too large to be located on the same computer as OpenText Information Archive components.

Applications consist of three important parts:

- The declarative configuration
- The folder with business data that is ingested
- The installation script with IA Shell commands

You can use any location for each of these three parts, provided you configure your application accordingly.

*Before you begin:*

- Install IA Shell on the computer where you want to run the install script and ingest the data.

- Configure IA Shell and connect to OpenText Information Archive.
- Make sure that the computer that hosts IA Shell can access the declarative configuration files and business data to be ingested. You can use a network filesystem.

For more information about installing and configuring IA Shell, see Section 1 “Overview” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*.

#### To locate applications outside of the examples folder structure:

1. In a text editor, open the application’s `install.bat` (Windows) or `install.sh` (Linux) file.
2. Make sure that the file has the correct path to IA Shell.
3. In a text editor, open the application’s `install.iashell` file.
4. Make sure that the following is true:
  - The `import` command must refer to the directory with the declarative configuration files. For example:

```
import "C:\MigrationTool\CustomerApplication\config"
```
  - The `ingest` command must refer to the folder that contains the business data to be ingested. For example:

```
ingest applications/CustomerApplication --from
"\server01\share_name\CustomerApplication\data"
```
5. Run the application’s `install.bat` (Windows) or `install.sh` (Linux) file.

### 8.8.3 Example application data

The sample data files (SIPs or tables) to ingest into application after installation can be found in the `<IA_ROOT>\examples\applications\<APPLICATION_NAME>\data` directory.

### 8.8.4 IA Shell installation script

There is an IA Shell installation script can be found in: `<IA_ROOT>\examples\applications\<APPLICATION_NAME>\iashell.install`.

The script contains a list of commands to run when installing the application and ingesting the data.

There are steps for:

- Establishing a connection with OpenText Information Archive.
- Installing DC configuration.
- Ingesting the sample data.

- Additional commands, if required, for running indexing jobs or installing additional configurations.

For more information, see the *OpenText Information Archive Shell Guide (EARCORE-ARE)*.



**Note:** It is possible to use any location for application sources, data files to be ingested, and installation scripts.

In other words, you can place the artifacts in any location.

To allow any location of IA Shell and data files, the following are required:

- IA Shell should be available on the PC, from where the application installation and data ingestion goes.
- IA Shell should be properly configured with files in the <IA\_ROOT>/config/iashell directory.
- The folder with application configuration sources and business data to be ingested must be reachable from the PC. A shared network location is allowed.

## 8.8.5 Learning tools

### 8.8.5.1 SIP

DC for SIP applications is best learned by example using what is provided in the example applications or generated by the Holding Wizard.

### 8.8.5.2 Table

DC for table archiving is best learned by example using what is provided in the example applications. As far as DC for table archiving is concerned, this chapter focuses on details that may be easily overlooked.

### 8.8.5.3 Declarative configuration and table metadata

Earlier in this guide, we explained table structure configuration through metadata (XML) files, which can be uploaded to a database resource in a table application.

There is a way to link to table metadata files from a DC YAML file so that you do not have to upload the metadata separately afterwards, but make it part of the DC upload.

In the Baseball sample application, we see the following:

```
database:
  name: Baseball-sql-db
  caseSensitive: false
  ciStore: ${store.name:default-store}
  defaultSchema: BASEBALL
  indexingOnIngest: false
  locale: en-US
  managedItemStore: ${store.name:default-store}
```

```
metadata:  
- resource: '*.xml'  
validatingOnIngest: true
```

This marks the configuration of a database object named “Baseball-sql-db” in the Baseball application.

The following section that the XML files in the same folder as this configuration YAML file represent the table metadata for this database. When you upload the DC, the table metadata will automatically be uploaded to the database resource:

```
metadata:  
- resource: '*.xml'
```

For more information, see Section 2.8.10 “ingest-table-metadata” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*

#### 8.8.5.4 XML namespaces in table DC

The following entry in the configuration.yml file for the Baseball application means that prefixes ia and table encountered in any XML and query in the included configuration files will be resolved to their corresponding URLs. These namespace declarations prevent such files from becoming cluttered with big namespace URLs:

```
namespaces:  
- prefix: ia  
uri: urn:x-emc:ia:schema:fn  
- prefix: table  
uri: urn:x-emc:ia:schema:table
```

#### 8.8.5.5 Configuring a table search

A user has the ability to specify a table reference in a search designed for a table application. This reference is required if you want to be able to apply retention to the rows of the table using a search.

The following code is a fragment of the configuration.yml file. The table reference is specified on the final line (in this example, the reference is accountReceivables:

```
searches:  
- name: AccountReceivables Search aic: null  
configure: ${configureSearches:  
description: Search on various AccountReceivables attributes query: null  
schema: ar  
state: ${search.state:published} table: accountReceivables
```

The table reference indicates which table the records reside in.

If one of the fields directly maps to one of the columns in that table, by specifying the table reference, an binding field becomes available. When viewing the records in a purge list, these binding fields are how the system can show the fields of the record. Calculated fields or fields referring to other tables cannot be shown when viewing the record in the purge list.

Associating a search with a table also allows for better formatting if, by setting the binding, the user wants to associate date fields to the in-line or detail panels.

### 8.8.6 Other

Refer to the Declarative configuration format for OpenText Information Archive ([https://github.com/Enterprise-Content-Management/infoarchive-sip-sdk/wiki/Declarative- Configuration](https://github.com/Enterprise-Content-Management/infoarchive-sip-sdk/wiki/Declarative-Configuration)) for further information about using DC to configure your system.



## Chapter 9

# Performing compliance tasks

*OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)* explains the conceptual differences between retention policies and holds (for more information, see Section 9 “Compliance – General concepts” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*). It also explains the retention lifecycle, which culminates in the disposition of data, once the required aging has been met.

This chapter outlines the tasks that a user with the Retention Manager role can perform on ingested data. You can use the **Compliance** tab in IA Web App to work with retention policies and holds, verify that data is protected, and dispose of data. You can also perform compliance tasks on data via a specific application.

## 9.1 Viewing, creating, and deleting a retention policy

The **Compliance > Retention Policies** tab allows the Retention Manager to:

- View the details of a retention policy
- Create and delete a retention policy
- Modify a retention policy (see [Changing a retention policy](#))



**Note:** If an administrator has set group permissions on a retention policy, you must be in at least one of those groups otherwise the retention policy will not be listed in this section (meaning you cannot view, edit, delete or apply that retention policy). For more information on setting group permissions on a retention policy see *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

### 9.1.1 Viewing retention policies

Each retention policy is displayed in a table with the following columns:

---

#### Policy Name

The unique name of the retention policy.

---

#### Policy Category

The retention policy's category, if any.

---

#### Aging Strategy

The policy's retention type: **Fixed Date**, **Duration**, **Event**, **Mixed**. For related fields, see *To create a retention policy* below.

---

#### In use

Whether the retention policy is in use.

**Last Modified Date**

The date and time on which the retention policy was last modified.

**Status**

Whether the retention policy's disposition processing is enabled (**Active**) or disabled (**Disabled**).

A side panel contains the following information related to the selected policy:

<b>Policy Name</b>	The unique name of the retention policy.
<b>Description</b>	A description of the retention policy (optional).
<b>Policy Category</b>	The retention policy's category (optional).
<b>Aging Strategy</b>	The policy's retention type: <b>Fixed Date</b> , <b>Duration</b> , <b>Event</b> , <b>Mixed</b> . For related fields, refer to <a href="#">Creating a retention policy</a> .
<b>Status</b>	Whether the retention policy's disposition processing is enabled ( <b>Active</b> ) or disabled ( <b>Disabled</b> ).
<b>Approved Date</b>	The date on which the retention policy was approved.
<b>Policy Approver</b>	The user who approved the retention policy.
<b>Notes</b>	Any additional information provided that can aid the Retention Manager (optional).
<b>In Use</b>	Whether the retention policy is in use.
<b>Created By</b>	The user who created the retention policy. If this retention policy was imported from configuration, this value will be the user that imported the configuration.
<b>Created Date</b>	The date the retention policy was created (only date is shown).
<b>Modified By</b>	The user who last modified the retention policy.
<b>Modified Date</b>	The date and time on which the retention policy was last modified.
<b>Custom Attributes</b>	A list of custom attributes specified to further define the application. These values can only be set when importing a retention policy from configuration.

Some customers may have many retention policies for content that spawns different geographical regions with different regulations. From the **Retention Policies** tab, three filters have been added that can be used in conjunction with each other:

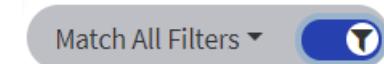
- The first filter can find a policy and filters (case-insensitive) text found either in the name of the policy or its description. This filter behaves similar to other filters that can be found on the **Application** listing page.
- The second filter allows filtering based on the retention policy. The default value is to show all retention policies for all categories.
- The third filter allows filtering based on the aging strategy. The possible values are:

- All (no filter)
- Fixed Date
- Duration
- Event
- Mixed

#### To apply column-based filters to the retention policy list:

You can apply one or more column-based filters to the list of retention policies to narrow results.

1. Turn the switch on.



You can combine multiple search criteria and use logical operators (**Match All Filters** or **Match Any Filter**) or combine multiple filters.

An additional row under the list header is displayed that includes at least one filter icon. The icon indicates that the column can be filtered.

2. Click the filter icon or anywhere in the list header cell.
3. In the popup window, select the operator to apply to the filter.

These operators are available based on the data type associated with the column. For example, the **Policy Name** column has the **Exact Match**, **Begins with**, **Ends with**, **Contains**, and **Not equal To** operators available while the **Last Modified Date** column has the **Between**, **Before**, **On or Before**, **After**, **On or After**, and **Not On** operators available.

4. Enter the filter criteria and click **GO** or press **ENTER**.
- The filter is also run when you switch to **Match All Filters/Match Any Filters** on the filter switch.

5. Repeat steps 2–4 if you want to apply multiple filters to the list.

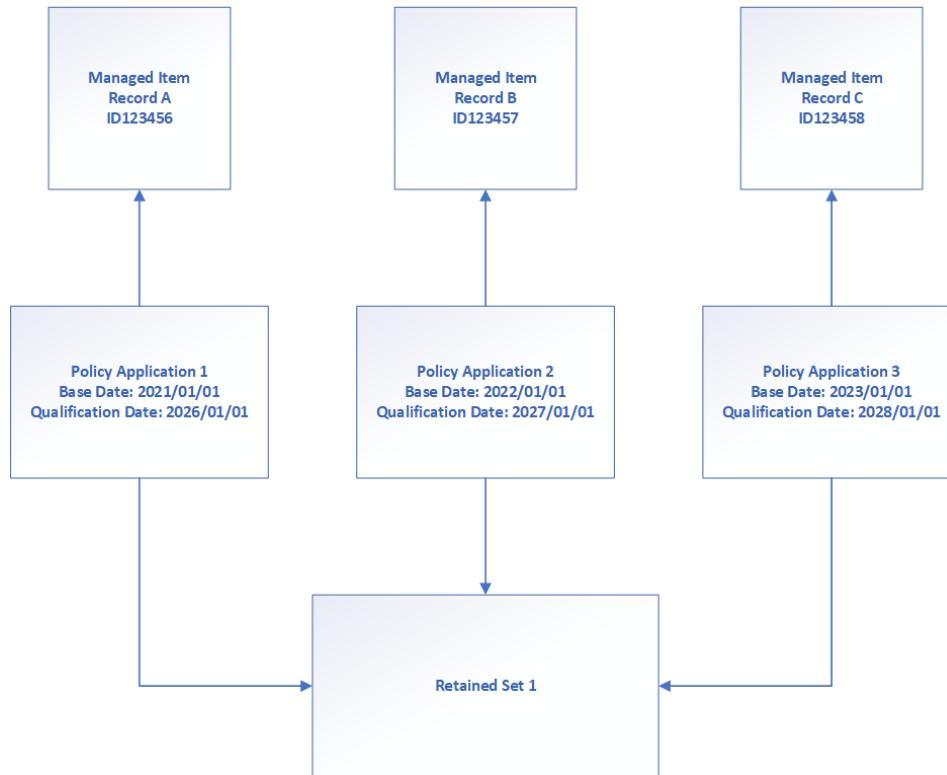
Once a filter is applied, you can remove it by clicking **X**.

## 9.1.2 Common use cases for retention

The following are some typical examples of using retention and describes what will be created in the system when a policy is applied. These use cases give a representation of the data model.

### Duration policy – Five years

The operation is to apply a policy called Retain\_5\_years to the Record A, B and C. We will use individual retention and a base date of 2001/01/01 for Record A, 2002/01/01 for Record B and 2003/01/01 for Record C.



### Event policy – Age three years after account closed

The operation is to apply an event policy to a customer's bank statements. All statements for a customer need to be kept for two years after the account is closed. An event policy called Retain\_For\_3\_Years\_From\_Close will be applied to statements 1 and 2 using customer CUST01 as the context. The event will be triggered on 2008/12/31.

If your retention policies change, you have the following options:

- If the duration, cutoff, or "retain until" date is changed, the retention policy can be modified.

- If the existing records are to be grandfathered, meaning the original qualification dates are fine, you can choose to requalify the existing policy applications.
- It is possible to create a new retention policy and use the new policy going forward. If retention was configured on the holding (for applying retention at ingestion), there is a current limitation that the retention information cannot be changed on the holding once it is in use. This limitation may be removed in future releases.

However, if you are using package or table retention, and the storage system is retention aware, it is advisable to not requalify, as the hardware will still protect the data until the original date is met. Some storage systems may not support increasing the retention period. If retention is applied to the records directly because individual dates are not pushed to the hardware, it is safe to requalify.

When running the Requalification job, it is recommended to either suspend the Disposition job schedule or disable disposition processing on the retention policy (the option is given when modifying a retention policy in use).

### Example 9-1: Apply three-year event policy to trades

All trades for a customer would age three years based on the date of the last trade.

There is an attribute on records that determine if this is the last trade for the customer and if the event needs to be triggered. The records also contain the customer ID, which will be used to group customer records and give OpenText Information Archive the ability to age them all based on the last trade date.

Create the following Event policy:

- Name: Trades
- Aging Strategy: Event
- Retains for: 3 Years
- Condition: Date of Last Trade

Within the rule, you would set the context to the CustomerID field of the trades. This groups all trades for the customer. If the last trades (LastTrade) field is 'Yes', set the event date and trigger the event.

The AIU rule would look like the following:

```
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;
rule "Apply policy to trades" when
    $aiuRecordBean:AiuRecordBean(); then
        ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();
        $applyRetentionBean.setRetentionPolicy("Trades3Years");

$applyRetentionBean.setEventDate(getTradeDate($aiuRecordBean.getRecordRows().get("LastTra
```

```

de"),
    $aiuRecordBean.getRecordRows().get("TradeDate"));

$applyRetentionBean.setEventContext($aiuRecordBean.getRecordRows().get( "CustomerID"));

$applyRetentionBean.setTriggerEvent(isLastTrade($aiuRecordBean.getRecordRows().get
("LastTrade"));
    $applyRetentionBean.setIndividualRetention(true);
    // do not create retained sets per customer
    // $applyRetentionBean.setRetainedSetName("Trades3Years : " +
    // $aiuRecordBean.getRecordRows().get("CustomerID"));
    $applyRetentionBean.setId($aiuRecordBean.getId());

        insert($applyRetentionBean);
end

function boolean isLastTrade(String lastTrade){
    if(lastTrade.equals("Yes"))
        return true;
    else
        return false;
}

function String getTradeEventDate(String lastTrade, String eventDate){
    if(isLastTrade(lastTrade))
        return eventDate;
    else
        return null;
}

```

This example can either have the event triggered (based on a value in the last trade field) or to not trigger the event for the application of an event policy.

The Trigger Event Rule job could have already triggered the event for the condition and context, and, when the Event policy is applied using the event date, the qualification date will be calculated after the Process Retention Events job is run.

The table row rule would look like the following:

```

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.TableRecordBean;
rule "Apply policy to trades" when
    $tableRecordBean:TableRecordBean(); then
        ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();
        $applyRetentionBean.setRetentionPolicy("Trades3Years");

        $applyRetentionBean.setEventContext($tableRecordBean.getRecordRows().get
("CustomerID"));
        $tableRecordBean.getRecordRows().get("TradeDate"));

$applyRetentionBean.setTriggerEvent(isLastTrade($tableRecordBean.getRecordRows().get
("LastTrade"));
    // example of using a function, if trigger event is not set to true, event date
is
    // ignored
    $applyRetentionBean.setEventDate(getTradeEventDate(
    $tableRecordBean.getRecordRows().get("LastTrade"),
    $applyRetentionBean.setIndividualRetention(true);

    // do not create a retained set per customer id
    // $applyRetentionBean.setRetainedSetName("Trades3Years : " +
    // $tableRecordBean.getRecordRows().get("CustomerID"));

```

```

$applyRetentionBean.setId($tableRecordBean.getId());
insert($applyRetentionBean);
end

function boolean isLastTrade(String lastTrade){
    if(lastTrade.equals("Yes"))
        return true;
    else
        return false;
}

function String getTradeEventDate(String lastTrade, String eventDate){
    if(isLastTrade(lastTrade))
    return eventDate;
else
return null;
}

```

Within the rule, there are two functions that check if the trade is the last trade. If it is, the set trigger flag is set to true, and the event date is set to the trade date. Since we cannot guarantee the order of the trades being returned from the search, the event may be triggered, but additional records can be added to the existing set. It is recommended to not define the name of the retained set per rule. Instead, either allow the system to create the set or use the name of the retention policy.



#### ➤ Example 9-2: Apply three-year event policy based on region

Within a global company, credit card report retention policies vary between jurisdictions. However, the system needs to apply the correct policy depending on where the customer is located. The account closing is stored in an external field and the system can call to see if the event needs to be triggered. The statement contains a field that identifies the region where the card is from. The rule can check this field and identify the proper retention policy to apply.

This can be modeled in two different ways:

1. Ensure that the retention policies are named the same as the region. Within the rule, set the policy name based on the region from the statement date.

We have three regions:

- North America
- Europe
- Asia

We have three policies created:

<b>Name:</b> North America <b>Aging Strategy:</b> Event <b>Retain for:</b> 3 years	<b>Name:</b> Europe <b>Aging Strategy:</b> Event <b>Retains for:</b> 5 years	<b>Name:</b> Asia <b>Aging Strategy:</b> Event <b>Retain for:</b> 2 years
------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------	---------------------------------------------------------------------------------

The following would be the AIU rule (the policy is named for the region):

```
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;

import com.mypackage.POJO_Class;

rule "Apply policy to credit card reports"
when
    $aiuRecordBean:AiuRecordBean();
then
    ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();
    $applyRetentionBean.setRetentionPolicy($aiuRecordBean.getRecordRows().get("Region"));

$applyRetentionBean.setEventDate(getEventDate($aiuRecordBean.getRecordRows().get("CustomerID"), $aiuRecordBean.getRecordRows().get("CloseDate")));
    $applyRetentionBean.setEventContext($aiuRecordBean.getRecordRows().get("CustomerID"));

$applyRetentionBean.setTriggerEvent(isAccountClosed($aiuRecordBean.getRecordRows().get("CustomerID")));
    $applyRetentionBean.setIndividualRetention(true);
    $applyRetentionBean.setRetainedSetName($aiuRecordBean.getRecordRows().get("Region"));
    $applyRetentionBean.setId($aiuRecordBean.getId()); insert($applyRetentionBean);
end

function String getEventDate(String customerId, String eventDate){
    if(isAccountClosed(customerId))
        return eventDate;
    else
        return null;
}

function Boolean isAccountClosed(String customerId) {
    POJO_Class myClass = new POJO_Class();
    return myClass.isAccountClosed(customerID);
}
```

The following would be the table row rule (policy is named for the region):

```
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.TableRecordBean;

import com.mypackage.POJO_Class;
rule "Apply policy to credit card reports" when
    $stableRecordBean:TableRecordBean(); then
    ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();

$applyRetentionBean.setRetentionPolicy($tableRecordBean.getRecordRows().get( "Region" ));

$applyRetentionBean.setEventDate(getEventDate($tableRecordBean.getRecordRows().get( "CustomerID"), $tableRecordBean.getRecordRows().get("CloseDate")));

$applyRetentionBean.setEventContext($tableRecordBean.getRecordRows().get( "CustomerID"));
    $applyRetentionBean.setTriggerEvent(isAccountClosed(
        $stableRecordBean.getRecordRows().get("CustomerID")));
    $applyRetentionBean.setIndividualRetention(true);

$applyRetentionBean.setRetainedSetName($tableRecordBean.getRecordRows().get("Region" ));
```

```

        $applyRetentionBean.setId($tableRecordBean.getId());
        insert($applyRetentionBean);
    end

    function String getEventDate(String customerId, String eventDate){
        if(isAccountClosed(customerId))
            return eventDate;
        else
            return null;
    }

    function Boolean isAccountClosed(String customerId) {
        POJO_Class myClass = new POJO_Class();
        return myClass.isAccountClosed(customerID);
    }
}

```

- Evaluate each statement and have a rule for each region. The policies can be named anything if the rule sets the correct policy.

We have three regions:

- North America
- Europe
- Asia

We have three policies created:

<b>Name:</b> North America	<b>Name:</b> Europe	<b>Name:</b> Asia
<b>Aging Strategy:</b> Event	<b>Aging Strategy:</b> Event	<b>Aging Strategy:</b> Event
<b>Retain for:</b> 3 years	<b>Retains for:</b> 5 years	<b>Retain for:</b> 2 years

The following is the AIU rule (rule for each region):

```

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;

import com.mypackage.POJO_Class;
rule "Apply policy to credit card reports North America" when
    $aiuRecordBean:AiuRecordBean();

eval($aiuRecordBean.getRecordRows().get("Region").equals("NorthAmerica"))

then
    ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();
    $applyRetentionBean.setRetentionPolicy("North America 3 Years");

$applyRetentionBean.setEventDate(getEventDate($aiuRecordBean.getRecordRows().
get    ("CustomerID"), $aiuRecordBean.getRecordRows().get("CloseDate")));

$applyRetentionBean.setEventContext($aiuRecordBean.getRecordRows().get("CustomerID")
);
    $applyRetentionBean.setTriggerEvent(isAccountClosed(
$aiuRecordBean.getRecordRows().get("CustomerID"));
$applyRetentionBean.setIndividualRetention(true);

$applyRetentionBean.setRetainedSetName($aiuRecordBean.getRecordRows().get("Region"))
;
    $applyRetentionBean.setId($aiuRecordBean.getId());
}

```

```

        insert($applyRetentionBean);
    end

rule "Apply policy to credit card reports Europe" when
    $aiuRecordBean:AiuRecordBean();
eval($aiuRecordBean.getRecordRows().get("Region").equals("Europe"))
then
    ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();
    $applyRetentionBean.setRetentionPolicy("Europe 5 Years");

$applyRetentionBean.setEventDate(getEventDate($aiuRecordBean.getRecordRows().
get("CustomerID"), $aiuRecordBean.getRecordRows().get("CloseDate")));

$applyRetentionBean.setEventContext($aiuRecordBean.getRecordRows().get("CustomerID")
);
    $applyRetentionBean.setTriggerEvent(isAccountClosed($aiuRecordBean.
getRecordRows().get("CustomerID")));
    $applyRetentionBean.setIndividualRetention(true);

$applyRetentionBean.setRetainedSetName($aiuRecordBean.getRecordRows().get("Region"))
;
    $applyRetentionBean.setId($aiuRecordBean.getId());
    insert($applyRetentionBean);
end

rule "Apply policy to credit card reports Asia" when
    $aiuRecordBean:AiuRecordBean();
eval($aiuRecordBean.getRecordRows().get("Region").equals("Asia"))
then
    ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();
    $applyRetentionBean.setRetentionPolicy("Asia 2 Years");
$applyRetentionBean.setEventDate(getEventDate($aiuRecordBean.getRecordRows().get("Cu
s
tomerID"), $aiuRecordBean.getRecordRows().get("CloseDate")));

$applyRetentionBean.setEventContext($aiuRecordBean.getRecordRows().get("CustomerID")
);

$applyRetentionBean.setTriggerEvent(isAccountClosed($aiuRecordBean.getRecordRows()
().get("CustomerID")));
    $applyRetentionBean.setIndividualRetention(true);

$applyRetentionBean.setRetainedSetName($aiuRecordBean.getRecordRows().get("Region"))
;
    $applyRetentionBean.setId($aiuRecordBean.getId());
insert($applyRetentionBean);
end

function String getEventDate(String customerId, String eventDate){
    if(isAccountClosed(customerId))
        return eventDate;
    else
        return null;
}

function Boolean isAccountClosed(String customerId) {
    POJO_Class myClass = new POJO_Class();
    return myClass.isAccountClosed(customerID);
}

```



### 9.1.3 Creating a retention policy

**To create a retention policy:**

1. On the **Compliance > Retention Policies** tab, click **+**.
2. Enter the following information:

---

**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**

Select the retention type for the policy being created. If you select:

- **Fixed Date**, enter the date after which the item can be disposed.
- **Duration**, 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. For retention policies that use a cutoff, the calculation is based on UTC (Coordinated Universal Time) midnight.



**Caution**

Customers using PowerScale 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 PowerScale ensures that data is kept as long as the originally stated date.

To resolve the issue, change the retention policy back to its original duration (increasing the duration or moving the retention to a later date).

- **Event**, 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 [Event-based retention](#) for further information.

**Mixed**, 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.

A date must be within the following range: 1000-01-01 –2999-12-31.

#### Approved Date

Enter the date that the retention policy was approved for use (optional). There is no business logic associated with this date.

#### 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**.

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 holding 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).

If a retention policy is in use, and its **Aging Strategy** is set to **Duration**, is the **Retain For** information is updated, one of two confirmation dialog messages may be displayed. The dialog displayed depends on whether disposition processing has already been disabled:

- If disposition processing has not been disabled, the change can either be:
  - **Retroactive (action required)**: If selected, disposition is disabled. To ensure the update is retroactive for existing applications, run the **Requalification job**. When the job is complete, enable disposition for the updated retention policy (see **Re-enabling disposition processing**).
  - **Apply forward**: If selected, the changes made to the retention policy only impact future applications of the policy. Items already impacted by the policy will be aged according to the previous settings.
- If disposition processing was previously disabled for the policy, run the **Requalification job** to ensure the update is retroactive for existing applications. If the **Requalification job** is not run, aging changes will only impact future

applications of the policy. When the job is complete, enable disposition for the updated retention policy.

### 9.1.4 Deleting a retention policy

**To delete a retention policy:**

1. In the **Compliance > Retention Policies** tab, click **X** for the retention policy being deleted.
2. When prompted to verify that you want to delete the selected retention policy, click **Delete**.

## 9.2 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, 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.

### 9.2.1 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 records at ingestion using a duration retention policy. It is possible to configure logic using the holding wizard to control which records should have policy applied.

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.

When applying retention or holds to table records, a search must be used. The search must define exactly one table to protect.

## 9.2.2 Hardware retention

### 9.2.2.1 Retention and the different store types

The table indicates the different supported storage and write once and read many (WORM) functionalities. Supports hardware protection means that the hardware will not allow the content to be either deleted or modified until the date pushed to the hardware has expired.

Retention dates are not pushed when using granular disposition. Retention periods cannot be shortened, only lengthened.

Storage type	Supports hardware protection (WORM)
File System	No
PowerScale (with SmartLock)	Yes
Dell EMC CAS Elastic Cloud Storage	Yes
<b>Cloud Storage</b>	
Pure Storage FlashBlade	Yes
Amazon S3 Storage	Yes
Dell EMC Elastic Cloud Storage (ECS)	Yes
PowerScale OneFS S3	Yes
Scality RING	Yes
Google Cloud Storage	Yes

Storage type	Supports hardware protection (WORM)
Microsoft Azure Blob Storage	No
NetApp StorageGRID	No
<b>OpenText Product Integration</b>	
Archive Center	Yes
Core Archive	Yes

Retention dates are not pushed when using granular disposition.

Retention periods cannot be shortened, only lengthened.

#### Holds and different storage types

OpenText Information Archive does not push hold information to storage.

#### Dell EMC Elastic Cloud Storage (ECS)

OpenText Information Archive can store all unstructured content (CI container) in a cheaper object storage option, such as ECS.

Integration with ECS storage is based on Dell EMC ECS SDK. The ECS APIs are located here:

- <https://github.com/EMCECS/ecs-object-client-java>
- <https://github.com/EMCECS/ecs-object-client-java/releases>

ECS supports retention support at the bucket level, as well as the retention on the object stored in the bucket.

The bucket retention period can only be changed by the ECS Management API via the ECS Management Service or coding to the ECS Management API.

The object retention period can only be set by the ECS S3 API.

#### Package Based Retention

OpenText Information Archive supports retention at various levels (for example, application, package and record). If ECS is used in the deployment, and if application- or record-based retention is used, retention information is not pushed to the storage. If ECS is used in the deployment, and if the package-based retention is utilized, then, by default, retention information is also pushed to the storage.

The scenario can be illustrated with an example:

AIP (package) may contain 1,000 AIUs (records) for bank accounts. And each record may have associated account statements in the PDF format. The structured data representing account information is stored in the XML repository. OpenText Information Archive creates a binary object combining 1,000 PDFs and stores it on ECS (or any other supported storage).

In the case of package-based retention, the retention information is pushed to the storage level, providing the hardware-level protection so physical

tampering of information can be avoided. In this scenario, retention information is known at the software layer (OpenText Information Archive) as well as at the hardware layer (ECS).

It also means, once the retention is pushed to the hardware, no modification to the binary object is possible, as ECS will not allow any modification on the object during the retention duration.

#### Record Based Retention

If there is a desire to use record-based retention and ECS storage is used, retention information is only managed at the software layer. In such a scenario, ECS does not have retention information on the binary object.

#### ECS Compliance Mode

If ECS is used in the compliance mode, then ECS enforces the retention requirement by needing retention on the bucket. Retention on the bucket means all objects in the bucket have one retention period. If bucket retention is active (not elapsed), writing the object under the bucket with retention information generates an error.

To avoid the error, OpenText Information Archive recommends that initial retention period on the bucket can be set to 1 second duration. It means the bucket retention duration will be expired after the creation of the bucket, allowing OpenText Information Archive to write information under it (for example, binary objects for each AIP can be stored underneath with retention information).

It does not mean the bucket can be deleted, as ECS will protect binary objects that have retention on it. ECS will not allow bucket deletion if the object inside has retention on it.

The connection to storage can be configured to use a proxy, if required.

---

#### Custom

Custom implementations need to implement the ContentStoreRetention interface and determine if they support pushing and getting hardware retention dates. For more information, see Section 2.2.10 “Custom storage” in *OpenText Information Archive - Administration Help (EARCORE-H-AGD)*.

### 9.2.2.2 When are dates pushed to the hardware?

How dates are pushed to the hardware depends on the data to which retention is being applied. 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. Furthermore, retention is only pushed to hardware for packages when it is in the committed phase.

Type	Hardware protection	Normal method retention is applied
Application	No	Manually, only via the web application

Type	Hardware protection	Normal method retention is applied
Packages	Yes	Automatically during ingestion (either by retention class or via default retention policy on application)
Tables	Yes	Manually, usually by the web application but can be done via job
Records	No	Via job or can be configured using the holding wizard

When applying multiple retention policies, if a shorter retention policy is applied, a new date will not be pushed. Dates can only be pushed further into the future. ECS does not support changing retention so do not apply multiple retention policies to the same table/package.

Dates are not pushed to hardware for granular disposition.

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 storage costs.

### Unstructured content and disposition

OpenText Information Archive can dispose unstructured content from the binary object (CI Container) if the package is being pruned. It is possible only if the retention date is not pushed to the hardware.

Supports partial update means that the storage allows the content to be updated if a date has not been pushed to the hardware. It is meant only to clear out certain sections of the CI container during disposition. Content for records that have not been disposed are left alone, hence the term partial update.

Storage type	Supports partial update on unstructured content (CI)
File System	Yes
PowerScale (with SmartLock)	Yes
Dell EMC CAS Elastic Cloud Storage	Yes
<b>Cloud Storage</b>	
Pure Storage FlashBlade	No
Amazon S3 Storage	No
Dell EMC Elastic Cloud Storage (ECS)	No
Google Cloud Storage	No

<b>Storage type</b>	<b>Supports partial update on unstructured content (CI)</b>
Microsoft Azure Blob Storage	No
NetApp StorageGRID	No
PowerScale OneFS S3	No
Scality RING	No
<b>OpenText Product Integration</b>	
Archive Center	No
Core Archive	No

### Hardware retention support considerations

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 the system and other structured data. 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 has expired. In this case, it is only possible to extend the retention period.

If your storage supports hardware retention, do not set default retention at the bucket level. However, if you plan to use Dell EMC Elastic Cloud Storage with compliance mode, see [Retention and the different store types](#).

#### 9.2.2.3 Changing retention policies based on storage

The following illustrates the rules for changing retention policies depending on the content store:

##### Pure Storage FlashBlade

You cannot reduce the retention period if it is applied to tables or packages.

##### File store

There are no restrictions.

##### Dell EMC CAS Elastic Cloud Storage

You cannot change the retention period if applied to tables or packages until the existing retention has elapsed.

##### Isilon (SmartLock)

You cannot reduce the retention period if it is applied to tables or packages.

##### Amazon S3 Storage

You cannot reduce the retention period if it is applied to tables or packages.

##### PowerScale OneFS S3

You cannot reduce the retention period if it is applied to tables or packages.

##### Scality RING

You cannot reduce the retention period if it is applied to tables or packages.

---

**Google Cloud Storage**

You cannot reduce the retention period if it is applied to tables or packages.

---

**NetApp StorageGrid**

Does not push retention to hardware.

OpenText Information Archive does not push hardware dates if the retention policy is applied to records (table rows or AIUs).

Refer to [When are dates pushed to the hardware?](#) for further information.

### 9.2.3 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 (PowerScale or Dell EMC CAS Elastic Cloud Storage). For PowerScale or Dell EMC CAS Elastic Cloud Storage, dates can only be pushed further into the future.

The ability to apply multiple policies is not supported by Dell EMC Elastic Cloud Storage.

OpenText Information Archive 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.

#### 9.2.4 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
Application with one retention policy applied to it: <ul style="list-style-type: none"><li>• Duration: 1 day</li><li>• Base Date: January 1, 2017</li></ul>	All rows have the same retention policy applied to them and, therefore, have the same base date  The retention policy 1 is processed on January 1, 2017. The application and all of its rows are disposed, along with all supporting tables.
Application 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>– Duration: 1 day</li><li>– Fixed Date: January 2, 2017</li></ul></li></ul>	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.  Retention Policy 1 is processed on January 1, 2017 but the application will not be eligible for inclusion in a purge list.  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.

Scenario	What Happens
<p>Application with a retention policy applied to it.</p> <p>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>
<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>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>

## 9.2.5 Table applications

If you are applying retention or holds in table application, it is recommended to specify a query ID for any searches that could be used for saved search or compliance. When creating or editing a table search, use the ID Query Editor to specify the ID, which is used for reloading saved searches and displaying records in purge lists, retained sets, and hold sets.

If you do not specify the query ID, only the fields available for the primary table of the search are used.

When using the query ID, any columns that match the following are not included when expanding a hold set, retained set, or purge list:

- Compliance columns
- Fields using CID (content)
- Hidden fields

For more information, see [Using the ID Query Editor](#) and [Enriching saved searches with compliance](#).

### 9.2.6 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. The package will still be eligible for inclusion in a purge list, however, the AIP may be pruned instead of disposed if

any of its records are not eligible for disposition. Records are put in separate purge lists 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.

#### 9.2.6.1 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).

### 9.3 Mechanisms for applying retention

The following table outlines which retention policy types can be applied using the different mechanisms:

Policy Types	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
Duration	Yes	Yes	Yes	Yes
Fixed Date	Yes	Yes	Yes	Yes
Event Based	Yes *	No	Yes	No
Mixed	Yes	No	Yes	No

\* Applying an event-based or mixed retention policy to applications, tables or AIPs is not supported.

#### 9.3.1 Using the Application Info tab

The **Application Info** tab allows the Retention Manager to apply or remove retention policies/holds from an application.

The Business Owner role can also access the **Application Info** tab but cannot perform any actions other than reviewing the details of the selected application.

### 9.3.1.1 Applying a retention policy to an application from the Application Info tab

The **Application Info** tab allows the Retention Manager to apply or remove retention policies/holds from an application. The Business Owner role can also access the **Application Info** tab but cannot perform any actions other than review the details of the selected application.

If multiple retention policies are applied to an application, all policies must be honored.

1. Select the desired application.
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 selecting a **Duration** retention policy, select a **Base Date** from the calendar picker.



**Note:** The **Base Date** no longer defaults to today's date.

5. Click **Next**.
6. 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**.
7. Review the information you have entered. When satisfied that the information is correct, click **Finish**.

### 9.3.2 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 <IA\_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, run 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.

2. 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.

3. Click **Apply Hold**.

A three-step wizard allows you to choose a hold and enter the details for the hold set being created.

4. In the **Choose Hold** step:

- a. You can use a filter to limit which holds are shown in the list. The text can match anything in either the name or description. There is also a second filter for limiting the hold to a particular type (either legal or permanent).

Select the hold that you wish to use (the first entry is automatically selected). The next button will be disabled if no hold was selected.

- b. Click **Next**.

5. In the **Specify Hold Set** tab:

- a. Enter a name for the hold set that you are creating.

- b. **Optional** 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.

### 9.3.3 Applying 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.

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 the Administrator, duplicate the `Apply Retention Policy to Records` job and make the following changes:
  - a. Give the job a unique name.
  - b. In **Applications**, select the Baseball application.
  - c. Ensure the **Schedule Mode** is set to Manual.
  - d. Set the following properties:

---

**Search Name**

Enter Search By Player Name.

---

**Search Set**

Enter Decrypted First Name.

---

**Search Criteria File**

Enter the following:

```
<data><firstName>Don</firstName><lastName></lastName></data>
```

The **Search Criteria File** could have also been a file location on the server that has this content. For more information, see the example in Section 3.7.3 “`Apply Retention Policy To Records` job” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

---

**Retention Policy Name**

Enter `Baseball-FixedDateInPast`.

---

**Context Type**

Ensure the value for this property is set to `attribute`. This attribute is used to determine how to set the base date but since we used a Fixed Retention policy, does not matter.

The **Trigger** property only matters for Event and Mixed Retention Policies.

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 recommended that you clone the job and set to manual so you can run it when desired. Normally, disposition is done on a schedule.

When the job has completed, the MASTER table will have those records removed.

#### 9.3.4 Applying a retention policy to an AIP

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. Select the AIP the retention policy is being applied to.
2. Click the action button 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. If selecting a **Duration** retention policy, select a **Base Date** from the calendar picker.



**Note:** The **Base Date** no longer defaults to today's date.

6. Click **Next**.
7. Enter the following information and click **Next**:
  - **Retention Set Name:** Enter a unique name for the policy.

- Enter a **Description** for the policy.
8. Review the information you have entered. When satisfied that the information is correct, click **Finish**.

You should receive the following message: **Retention policy applied successfully**.

### 9.3.5 Using jobs to apply retention

There are two jobs that apply retention to objects in the system:

When applying retention or hold to records, a library store is required.

- For SIP applications, a library store is already required to store the structured data for the application. You can either create a dedicated library store for granular retention or can use the same store as the structured data. Each holding's retention store can be set using the IA Web App interface via the Edit Retention and navigate to the **Retention Store** tab.
- For table applications, a library store is required to store the granular retention, as a library store is not used to store structured data. Each database's retention store can be set using the IA Web App interface via the Edit Stores.

If the list of stores is empty, create a store for the retention of type Library.

- The **Apply Retention Policy to Records** job
- The **Apply Retention Rule to Records** job

#### 9.3.5.1 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.

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.

The `searchCriteriaFile` property is required to provide values for all input fields for the query. If one or more input fields are missing, the job will fail and an error message is issued.

Set the following properties:

**Search Name**

This is the name of the search to run. The search must be defined in the application and must be ready.

**Search Set**

Search composition within the search to use.

**Search Criteria File**

This is a path to a criteria file that contains what you want to narrow the results to. For more information, see the example in Section 3.7.3 “Apply Retention Policy To Records job” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

**Retention Policy Name**

All four types of retention policies are supported. Depending on the type of policy, other job properties might have to be set.

**Retention Date Attribute**

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

**Context Type**

Used for Event policies. There are two possible values:

- **attribute:** The job gets the context from an attribute in the data. The attribute is set in the context field.
- **fixed:** The job uses the value in this field as the context for the events.

For example, if you want to have all records for an employee age for 5 years after the employee has left the company, make the context the employee ID, since it will be the same for all documents. OpenText Information Archive 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).

**Context**

The value is dependent on the **Context Type** 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.

**Trigger Check Attribute**

The job can 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, `hasLeft = true`). The attribute is `hasLeft` and the value is `true`. In this case, the **Trigger Check Value** 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.

**Trigger Check Value**

This is value to check against for whether the event should be triggered.

**Trigger Date Attribute**

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 `terminationDate` that contains the date the employee left the company. Putting `terminationDate` in the **Trigger Date Attribute** property would instruct the job to fetch the trigger date from the `terminationDate` attribute.

**Trigger**

Event policies can be triggered using the records data. If selected, the job attempts to trigger the event policy that was applied to the records.

### 9.3.5.2 Using the Apply Retention Rule to Records 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. Runs 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 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>
```

### 9.3.6 Restricting the apply retention and hold jobs to only evaluate records ingested since the last run for SIP applications

OpenText Information Archive allows you to apply retention to AIP applications using a job. This section illustrates how to configure these jobs to apply only to new records and avoid having to process existing records.

The job definition records when the last time the job ran against each application. If a job definition is duplicated, the system will clear the last successful run information. Furthermore, if an application is deleted, the last successful run information will be removed from the job definition, if present.

The system records when specific AIPs were ingested. This is the criteria that is used to exclude the AIPs from processing.

### 9.3.7 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 binding (option only available for table searches). This option is only possible if the search has defined both a schema and a 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 example above was meant for debugging to determine why none of the rules were matching and to ensure that only one hold was applied.

### 9.3.7.1 Viewing an application's rules

The **Rules** tab allows Retention Managers and Developers to view rule information that has been set up for a particular application. Select an application and click **Rules** to access the tab. The Patent and PhoneCallsGranular sample applications contain pre-loaded rules.

Any rules set up for the application are displayed in a table that includes:

- The name of the rule.
- The type of rule, which may include Apply Hold, Apply Retention and Trigger Event.
- The date the rule was loaded into the application.
- The date the rule was last modified.

Use the **Find rule by name** field to locate a specific rule by entering a name and clicking Enter. To only view rules of a specific type, select a **Rule Type** from the list.

Select a rule in the table to view the rule information in a panel on the right side of the page.

### 9.3.7.2 Running rules to apply retention for an application

The following procedure is an example of how to run a job that uses rules to apply retention to one of the sample applications.

1. Install the sample PhoneCallsGranular application.

This application creates two retention policies, a Mixed and an Event policy type.

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. Login to the IA Web App as an Administrator.
3. In the **Administration > Jobs** tab, edit the Apply Retention Rule to Records job.
4. Set **Applications** to PhoneCallsGranular.
5. Update the following properties:

---

**Search Name**

Enter FirstName\_Operator. Ensure the search status is set to Ready.

---

**Search Set**

Enter Set 1.

---

**Search Criteria File**

Enter <data></data>.

This property can be used to narrow the criteria of which records to evaluate the rule against. For more information, see the example in Section 3.7.3 “Apply Retention Policy To Records job” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

#### Rule Name

Enter the name of the rule. This property allows you to limit the job to only run one rule.

#### Developer Mode

Select to test match without performing action; otherwise, leave blank.

Rules are found in the PhoneCallsGranular and Patent sample applications’ compliance-config folders. The rules are not compiled and 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.

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 IDHi Markto 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 a 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 the Requalification job:

- 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.

### 9.3.7.3 Using the rule engine to apply a retention policy or hold to records

Customers can use the rules engine to apply 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 running a search.

Rules are stored as SDX objects in the system. Currently, the only method to create the rules is via REST API. 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 OpenText Information Archive 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 *OpenText Information Archive REST API Developer's Guide* on support.opentext.com (<https://support.opentext.com/>) to learn how to set up a rule.

1. In the **Administration > Jobs** tab, click  and select **Edit job** for the **Apply Retention Policy to Records** job.
2. Select the desired application from the **Applications** list.



**Note:** Because searches are specific to an application, you cannot apply the job to more than one application.

In the **Properties** section, update the **Search Name**, **Search Set**, and **Search Criteria File** properties. The **Search Criteria File** allows the user to narrow the parameters of the search (for example, search for specific customer IDs). For more information, see the example in Section 3.7.3 "Apply Retention Policy To Records job" in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

3. Save and run the Apply Retention Policy to Records job.

The new retained sets appear in the **Retention Sets** tab for the selected application.

### ➡ Example 9-3: Apply 120 day policy to all trades

You want to apply a 120-day retention policy to all trades that are created in the system. Using a date in the data of the record that matches the created date in the system. You feel that the policy will not change so using individual retention is fine.

Create the following Duration policy:

- Name: Trade120
- Aging Strategy: Duration
- Retain for: 120 days

To organize the records, you could create a retained set per day. The AIU rule would look like the following:

```
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 trades"
when
    $aiuRecordBean:AiuRecordBean();
then
    ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();
    $applyRetentionBean.setRetentionPolicy("Trades120");

    $applyRetentionBean.setBaseDate($aiuRecordBean.getRecordRows().get("CreatedDate"));
    $applyRetentionBean.setIndividualRetention(true);
    $applyRetentionBean.setRetainedSetName("Trades120");
    $applyRetentionBean.setId($aiuRecordBean.getId());

    insert($applyRetentionBean);
end
```

In this example, the name of the retained set is set to the name for all the records that have the same policy, since the individual retention flag is set to true, and each record will have its own base date.

The table row rule would look like the following:

```
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.TableRecordBean;

//declare any global variables here
rule "Apply policy to trades" when
    $tableRecordBean:TableRecordBean(); then
        ApplyRetentionBean $applyRetentionBean = new ApplyRetentionBean();
        $applyRetentionBean.setRetentionPolicy("Trades120");
        $applyRetentionBean.setBaseDate($tableRecordBean.getRecordRows().get(
            ("CreatedDate")));
        $applyRetentionBean.setIndividualRetention(true);
        $applyRetentionBean.setRetainedSetName("Trades120");
```

```
$applyRetentionBean.setId($tableRecordBean.getId());  
  
insert($applyRetentionBean);  
end
```

Within the rule, the retained set name is set to the policy plus the date. When OpenText Information Archive goes to apply the policy, if the retained set already exists, it will put the trade in the existing set.



#### 9.3.7.4 Getting started with rules

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 run on a scheduled basis.
- Does not require any precompiling of Java 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?

IA Shell can also be used to create a rule (see Section 3.21 “Creating rules” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)* for more information).

#### 9.3.7.5 Jobs using rules

The following jobs use rules:

- Apply Retention Rule to Records processes the results of a search and can apply retention to the records.
- Apply Hold Rule to Records processes the results of a search and can apply a hold to the records.
- Trigger Event Rule allows you to set the fulfillment date for events.

For applications that have rules, the rules can be viewed from the application’s **Rules** tab. The PhoneCallsGranular sample application contains examples of rules for each type.

### 9.3.7.6 Example of granular retention

The following procedure is meant to outline how granular retention works. The following scenario assumes that the PhoneCallsGranular sample application has been installed. The path you choose can be the location that you decided to install.

1. As an Administrator, access the **Jobs** tab to edit the Apply Retention Rule to Records job:
  - a. Set **Schedule Mode** to **Manual**.
  - b. Select **PhoneCallsGranular** from the **Applications** list. The job must be scoped to an application that includes at least one defined rule.
  - c. Update the following properties:

---

#### Search Name

Enter FirstName\_Operator. The value entered could be any of the searches defined for the application, as long as the search status is Ready. The field is case-sensitive.

---

#### Search Set

Enter Set 1. The value entered needs to match the name of a search set.

---

#### Search Criteria File

Enter <data></data>. The value is the query parameters for the search. In this scenario, no criteria is used because we want the rule to evaluate all records. If your search set specified mandatory parameters, you will need to specify the parameters. For more information, see the example in Section 3.7.3 “Apply Retention Policy To Records job” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*..

---

#### Rule Name

Leave blank. If you only want one rule to run, use this job.

---

#### Developer Mode

Do not select the **Developer Mode** box. If selected, the jobs tests if the rule is working as expected and does not apply the policy to the record.

2. 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 ran successfully.

If you are running the server manually, you can also see information about the job.

Ignore the message about the delete temporary file. Notice that some of the records have policy A or B applied.

3. As the Retention Manager, view the details for the two retention policies in the **Compliance > Retention Policies** tab. 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.

4. As an Administrator, scope the Trigger Event Rule job to the PhoneCallsGranular application. Run the Trigger Event Rule job.
  5. Edit the Process Retention Events job:
    - a. From the **Applications** list, select the application that was used to trigger the events.

Since the events that you triggered are associated with more than one retention policy, leave the **Retention Policy Name** property blank (the default). We could have set the field to one of the **Policy A** or **Policy B** retention policies to restrict which policies are evaluated to improve job performance but, in this case, multiple policies are referencing the event.
    - b. Run the Process Retention Events job.
  6. Edit the Generate Purge Candidate List job by updating the Schedule By field to **Manual**. Run the Generate Purge Candidate List job.
  7. As a Retention Manager, access the **Purge Lists** tab for the PhoneCallsGranular application. Expand the purge list and export it.
    - a. 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.

- b. Since we know that we have the correct records, approve the purge list.

8. As an Administrator, edit the Dispose Purge Candidate List job and ensure that the **Schedule Mode** is set to **Manual**. Run the Dispose Purge Candidate List job.

9. As a Retention Manager, access the **Purge Lists** tab for the PhoneCallsGranular application. Expand the purge list.

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 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.

- a. Navigate to the **Packages** tab. Notice that two records have been removed from the first package.
- b. Click the package to see logs about the disposition that was done on the package.

Rules can be found in the following directory: <IA\_ROOT>/examples/applications/PhoneCallsGranular/config/compliance-config/rules. For the Patent application, the rules can be found in the compliance-config/rules directory.

### 9.3.7.7 Rule files

The rules are created in a separate file <filename>.DRL, which is a text file using the Drools format. You must generate your own files and upload them into OpenText Information Archive using IA Shell (declarative configuration). The Patent and PhoneCallsGranular applications provide examples of how to install rules using declarative configuration

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.

Rules are always associated with an application. When running a job, only rules associated with that application and the type associated with that job are run. For example, if the Apply Retention Rule to Records job is run for the PhoneCallsGranular application, only the Apply Retention rules are used. Meanwhile, the rules for the Patent application are ignored.

### 9.3.7.8 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 run.

The following is a sample rule file for the PhoneCallsGranular example application. There are two rules contained in this rule file. The first rule checks if the customer ID is 00103 or 000147, then it applies Policy C. The second rule checks if the customer is 00391, then apply Policy D. To view the rule, refer to the `rule-applyRetentionRule.drl` file located in the `<IA_ROOT>/examples/applications/PhoneCallsGranular/config/rules` directory.

If the `when` clause evaluates to `true`, the `then` clause will run. An `apply retention` result bean is created, populated with the relevant information and then returned to the rule service.

### 9.3.7.9 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 OpenText Information Archive 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 are 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:

---

**RetentionPolicy**

Name of the retention policy to apply to this record.

---

**BaseDate**

For duration-based policies (Duration and Mixed), this is the date the retention system will use to start aging from for this record.

---

**EventDate**

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.

---

**TriggerEvent**

Trigger the event-based on the context specified in the Context field.

---

**EventContext**

Context for event-based policies. Allows for grouping of records and all records using this context and policy will use the same event date.

---

**Id**

This is the AIU or table row ID. This value must be set from the `AiuRecordBean`/`TableRowRecordBean` : `$applyRetentionBean.setId($aiuRecordBean.getId())`.

---

**NewRetainedSet**

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.

---

**RetainedSetName**

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.

---

**ReplaceRetention**

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.

---

**IndividualRetention**

Indicates to the retention system to age this record individually with its own date.

---

**retentionPolicyToRemove**

If specified, indicates the name of the retention policy to remove. Expected to at least be defined (but is not considered an error if it is not applied).

## 9.4 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. Note that it is possible to configure the job to only evaluate newly ingested records since the last time the job ran.

If applying retention during ingestion, use a duration retention policy (configurable via the Holding Wizard).

Using the Holding Wizard is certainly simpler than using rules. Rules allow more flexibility and must be used if a duration retention policy will not suffice, or for more complicated base date calculations.

### 9.4.1 Applying granular retention using the holding wizard

While you can apply record-based retention via the rules-based retention mechanism, it is a complex procedure, as a rule must 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.

## 9.4.2 Updating configuration to apply retention classes dynamically on records

If you need to apply different retention classes to records based on values from those records, update the holding configuration generated by the Holding Wizard.

1. Within the PDI configuration, locate the XQuery that selects the AIU records in which the AIU retention will be applied:
2. Update the return statement of the query to add the **retentionPolicy** attribute.

```
return <aiu retentionPolicy=< retentionPolicyName >" id="${aiu/@ia:id}"  
baseRetentionDate="${aiu/n:CallStartDate}"/>
```
3. The value of the **retentionPolicy** attribute should be the name of the retention class to apply. If the attribute is not there, or the value is blank, the retention class defined in `duration.retention.policy.name` will be applied. The `duration.retention.policy.name` value must always be set with a valid retention policy.
4. Any retention policy that will be returned by this XQuery must be created prior to ingestion.

## 9.5 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. Also refer to the following sections:

- Viewing items in a retained set
- Verifying using the application's Tables tab
- Verifying using the application's Packages tab
- Verifying retention and holds in search results

### 9.5.1 Viewing items in a retained set

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:

---

**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.

---

**Earliest Qualification Date**

The earliest date the retention set qualifies for disposition. The qualification date may be null if event-based retention was applied.

In the **Retention Sets** tab, click the arrow to the left of the retention set name 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.

#### To export items in a retention set:

1. Check the boxes for the items being exported or click **Select All** to select all of the items contained in the retention set.
2. Click **Export Options**.  
Exporting retention set data is an asynchronous request.
3. In the **Export in Background** pop-up, update the default **Request Name** information, if desired.
4. Click **Start Background Request**.  
Navigate to the **Background Requests** tab to access the exported information.

### 9.5.2 Verifying that an item is protected

Depending on the type of item, the procedure is different:

- Application
- Table
- Package
- Record

#### 9.5.2.1 Verifying using the application's Tables tab

If the application is a table archive, the application's **Tables** tab provides an overview on whether the table is under retention or hold, or if any of its records are under retention or hold.

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 under retention, the table itself is not considered under retention.

**To view if a retention or hold is applied to a table at the row level:**

1. Select the table in the tab.
2. Make sure the details panel is open.
3. Click the details panel **Compliance** tab.

The tab provides information about retention and holds that are inherited from the application, applied directly to the table, or applied directly to any of the table's records.

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.

### 9.5.2.2 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, or any records contained within the package.

To view retention directly applied to a specific package, click the package name.

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.

**To view if a retention or hold is applied to a package at the row level:**

1. Select the package in the tab.
2. Make sure the details panel is open.
3. Click the details panel **Compliance** tab.

The tab provides information about retention and holds that are inherited from the application, applied directly to the package, or applied directly to any of the package's records.

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.

### 9.5.2.3 Verifying retention and holds in search results

To verify retention or holds in a set of search results:

1. Run the desired search.
2. Select a row in the set of search results.
3. Click the  button on the right side of the screen. This allows you to view compliance-related information about the selected row of search results. The information displayed depends on how the results were configured by the search designer.

## 9.6 Event-based retention

### 9.6.1 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.

### 9.6.2 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.

The following property can be configured:

#### Trigger File

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 trigger 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>
```

#### 9.6.2.1 Populating event dates for the Trigger Event Policy job

An .xml file is used to populate event dates for the Trigger Event Policy job:

 **Example 9-4: 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>
```



The following provides further information about what to include in the XML file:

**context**

Enter the context for the event (for example, employee number).

**triggerdate**

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.

**condition**

Enter the name of the condition, which must match the condition specified on the retention policy. The value is case sensitive.

### 9.6.3 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).

### 9.6.4 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. Whatever triggers the event might come from an external system (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;
import org.slf4j.Logger;

//declare any global variables here
global Logger logger;

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);
    logger.info( "Trigger Event for AIU Records");
end
```

## 9.6.5 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.

## 9.7 SIP retention

### 9.7.1 How does refactoring work?

The following use cases 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 backup is taken. No backup is taken, however, if the package is marked for disposition and is waiting for a confirmation.

There are options on the holding under retention that can be set to reduce the amount of space used by the CI container after refactoring in the Disposition Policy section.

### 9.7.2 Purge confirmations

It is possible to define on the holding a confirmation of the purge ([Edit Confirmation](#)).

In the advanced settings, it is possible to specify a query for the AIUs to define which fields are included in the confirmation. This can be used as a certificate of destruction for proof of exactly which records are disposed. The purge confirmation is generated by the Disposition job. There is no need to run the Confirmation job.



#### Caution

If you want confirmations for the other package states, you must run the Confirmation job before running Disposition.

## 9.8 Determining if a retention policy is in use

A policy is in use when it is applied to an object. A policy is not considered in use if it is only referred to in a job configuration. If a retention policy is removed, 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 information button. The Default retention policy is displayed.

## 9.9 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

When changing a retention policy that is in use, decide whether to disable disposition processing for the retention policy. If nothing is chosen, the default is to disable the retention policy.

1. In the **Compliance > Retention Policies** tab, select the policy being edited and click the pencil button.

2. Edit the fields, as desired.
3. Click **Save**.

A pop-up is displayed asking how you would like to treat existing and future qualification dates for items impacted by the policy being updated. Select one of the following options:

- **Retroactive (action required):** Select this option to disable disposition and have the existing retention applications be updated to use the new rules. To make the change retroactive for existing applications, run the Requalification job. After job has finished, re-enable disposition for this policy (See [Re-enabling disposition processing](#)).
- **Apply forward:** Select this option to ensure that changes to the retention policy will only impact future applications of the policy, and items already impacted by the policy will be aged according to previous settings.

4. Click **Continue**.

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 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.

The customer can also change the:

- Approved Date
- Policy Approver
- Notes

#### To edit a retention policy:

1. In the **Compliance > Retention Policies** tab, select the policy being edited and click the **Pencil** icon.
2. Edit the fields, as desired. For further field information, see [Creating a retention policy](#).
3. Click **Save**.

When changing a retention policy, existing objects will continue to age with the settings on the retention policy at the time of application.

## 9.9.1 Removing retention

### 9.9.1.1 Removing retention from an application from the Application Info tab

The **Application Info** tab allows the Retention Manager to apply or remove retention policies/holds from an application. The Business Owner role can also access the **Application Info** tab but cannot perform any actions other than review the details of the selected application.

**To remove retention from an application:**

1. Select the application.
2. In the **Application Info** tab, click  for the retention set that you want to remove.
3. When prompted to verify that you want to remove the selected hold set, click **Remove**.

### 9.9.1.2 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 properties can be configured:

---

#### Retention Policy Name

Name of retention policy that will be removed from items. The retention policy must be defined, and the name is case-sensitive.

---

#### Type

Use this Optional field to specify which type the policy will be removed from. Possible values include:

- application
- aip
- aiu
- table
- row

Row refers to a table row.

---

## 9.9.2 Re-enabling disposition processing

When modifying a retention policy's dates to be retroactive, it is necessary to run the **Requalification job**. Before running the job, the policy's disposition processing needs to be disabled. After the job is done, the policy's disposition processing needs to be re-enabled.

**To determine if a policy's disposition processing is disabled:**

1. Select the policy in the **Compliance Retention Policies** tab.
2. Look at the policy's **Status** column in the table. If the policy's disposition processing is disabled, the **Status** is set to **Disabled** and a warning appears at the top of the tab's side panel.

**To re-enable a policy's disposition processing:**

1. Select the policy and click the **Pencil** icon on the upper-right corner of the side panel.
2. Scroll down to the bottom of the **Edit Retention Policy** form.
3. Click the **Disable Disposition** check box to remove the check mark.
4. Click **Save**.

**To determine which retention policies have disposition processing disabled:**

- Click the **Status** header to sort the column in reverse alphabetical order.  
The retention policies with disposition disabled will appear at the top of the table.

## 9.9.3 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 has one mandatory property, **Retention Policy Name**. Enter the name of the retention policy that changed.

## 9.10 Disposition flow

Disposition is the final phase of a three-step lifecycle. Those phases are:

- Apply retention policies to item.
- Wait for the required regulated time that the objects need to be retained.
- Dispose of the objects after approval is given.

The two main jobs for the dispose phase are:

- 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 (see Section 3.7.16 “Generate Purge Candidate List job” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*).
- Dispose Purge Candidate List: This job disposes of items in approved purge candidate lists (see Section 3.7.14 “Dispose Purge Candidate List job” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*).

After a purge candidate list is generated, and before it is processed by the Dispose Purge Candidate List job, it must be approved.

Below we provide details on the workings of these jobs and the disposition process as a whole.

The general flow is:

- Generate the purge lists using the jobs.
- As part of the approval process, optionally, export the purge list and then approve it,
- Run the Dispose Purge Candidate List job,
- If disposing an application, run the Clean up Purge Candidate List and Applications job to finish the disposition.

### 9.10.1 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 during the next run of the Dispose Purge Candidate List job. If, in the meantime, a hold is placed on items in the list or put under a longer retention policy that has not 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.

When Retention Manager views the purge lists for an application, a list may contain a count of -1. What this means is that the count has not been initialized because the Generate Purge List job is still running. When the job has finished, the count will be set. If the Generate Purge List job fails for some reason, the job can either be retried or it can be run again, and the purge lists that had not be initialized will be cancelled. Running the Clean up Purge Candidate Lists and Applications job removes disposed and cancelled purge lists.

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.

When any of the following actions are undertaken, the Retention Manager must indicate the reason:

- Approving or rejecting a purge list,
- Cancelling the approval or rejection of a purge list.

The reason is then included in the audit for the action. The audit cancelling an approval or a rejection are mapped to a REVOKE audit event type.

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 through the Dispose Purge Candidate List job, and the items are purged, the Clean job needs to run 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.

## 9.10.2 Purge Lists

The Retention Manager can select an application and access the **Purge Lists** tab to view any purge lists that may have been generated.

The following table describes the fields for a purge list:

---

### Purge List Name

The name of the purge list.

---

### Type

Indicates where the retention was applied:

- Package
  - Table
  - Records
  - Application
- 

### Retention Policy

Name of the longest retention policy that was on the items.

---

### Items

Number of items in the purge list. For example, for the PhoneCalls sample application, 10 items are put into the purge list (each package has a different number of records contained within it).

---

### Created Date

Indicates the date the purge list was created.

---

### Status

Status of the purge candidate list:

- **Under Review:** The purge list has been created but not yet approved
  - **Approved**
  - **Rejected**
  - **Canceled**
  - **Disposed:**
  - **Disposing:** The Dispose Purge Candidate List job is currently processing the purge list
- 

When expanding the purge list, additional information is provided. For example, if a purge list contains packages, the following is displayed:

---

### Package Name

Indicates the name of the package.

---

---

**State**

Indicates the state of the package (typically, this would be Completed).

---

**Hold**

Indicates if the package is under hold. Will not be true if one of records is under hold, but the package is not under hold.

---

**Records**

Indicates the number of records in the package.

---

When viewing a purge list that has been disposed, if the items were disposed (and not pruned), they will not show up in the purge list. If the purge list contained packages that were pruned, the **Records** field indicates how many records remain in the package.

The available actions a Retention Manager can perform on a purge list depends on the Status of the list:

---

**Under Review**

You can approve or reject the purge candidate list.

---

**Approved**

You can cancel the approval of the purge candidate list.

---

**Rejected**

You can perform the following actions:

- **Cancel the Reject:** Moves the purge list back to under review, which allows you to review the list.
  - **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 can be applied to any items that should not be disposed so they are not included in the next generated purge list.
- 

**Cancelled**

There are no actions that can be performed

---

**Disposed**

There are no actions that can be performed

---

**To perform any of the above actions:**

1. Select the purge candidate list in the **Purge Lists** tab.
2. Click the right arrow and select the action you want to perform on the list.

### 9.10.3 Performing actions to a purge list

The **Purge Lists** tab allows the Retention Manager to perform the following actions on a purge list:

- Review a purge list
- Export information about what is in the purge list. Depending on where the retention was applied, there will be two options:
  - Export each table or package individually contained in a purge list for review purposes.
    - For packages, the ability to export is only possible if there is a transformation that specifies which fields to export. The PhoneCalls example application provides an example in the transformation-PhoneCallsTransformationCsv.xq file and demonstrates how to mask sensitive fields. This transformation cannot be configured through the holding wizard and can only be done via declarative configuration.  
When exporting a package, the search developer must define at search composition which fields are exported.
    - For tables, the search form indicates which fields to export.  
In the case of tables, the raw values of the fields are exported, and any values marked as encrypted will be exported.
  - Export a purge list that contains records.
    - For AIUs, the columns exported will be the same as the search that was used to apply retention.
    - For table records, the search will be used. However, any fields that do not have a binding set to the table will not be exported. Columns on other supporting tables will not be exported.

Refer to [Exporting purge lists](#) for more information.

- Approve a purge list
- Cancel a purge list
- Reject a purge list
- Cancel either the approval or rejection of a purge list (move the purge list back to in-review)

### 9.10.3.1 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.

It is possible to export purge lists that have been cancelled or fully disposed. The extension for exporting purge lists containing either tables or table records is .gz.

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 zipped 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).

### 9.10.3.2 Exporting packages or tables from a purge list

OpenText Information Archive supports export configurations for purge lists containing records as well as packages.



**Note:** Individual records not eligible for disposition (for example, records that are under a hold) will be excluded from the export. If you export a package or table from the purge list, any records under a hold (no longer eligible for disposition) will not be exported.

#### To export a package or table from a purge list:

1. Access the application's **Purge Lists** tab.
2. Expand one of the purge lists to view the packages or tables that are in the selected purge list.
3. For the package or table being exported, click and select **Prepare Export**.  
For a table application, the **Export in Background** dialog box is displayed. If desired, enter a **Request Name** or keep the name that has been automatically generated. Click **Start Background Request**. Navigate to the **Background Requests** tab to access the exported data.  
For a SIP application, the **Prepare Export** dialog box is displayed. Select the following options to prepare a package export:
  - Select the **Search Form**. The selected search form must have at least one export option configured for it. Otherwise, you will not be able to proceed,
  - Select the **Search Set**. If there is only one search set, the search set is displayed and this field cannot be updated.
  - Select the **Export Configuration**. If only one export option was configured for the search form, the option is displayed and cannot be updated.
  - If desired, enter a **Request Name** or keep the name that has been automatically generated.
4. Click **Prepare Export**. Navigate to the **Background Requests** tab to access the exported data.

### 9.10.3.3 Providing a reason and attaching approval documentation when approving a purge list

Retention Managers often must verify with the owners of data that the data can be disposed. Furthermore, a Retention Manager must obtain approval from the data owner to dispose the data.

Prior to approving a purge list, the Retention Manager can:

- Provide a reason why the data contained in a specific purge list is being deleted.
- Optionally, attach an approval document prior to approving the purge list. The file must be less than 50 megabytes in size. Otherwise, an error message is issued.

#### On the Purge Lists tab for a specific application:

1. Click the **Approve** button for a purge list. The Approve Purge List dialog is displayed, which indicates the name of the purge list, as well as the number of items contained in the list.
2. Enter the required information in the **Reason** field.
3. If any approval documentation is required (for example, an approval email), click **Upload file**. Attach the necessary document.
4. Click **Approve**.

The information entered by the Retention Manager is reflected in the system's audit records. Once the Archive Audits job has been run, a user can use the **Audit > Application Audit** to verify the approval information entered by the Retention Manager. Any attachments added during the disposition process can also be downloaded from the **Audit > Application Audit** results.

Furthermore, this information can also be exported. User can opt to export with content, meaning a file uploaded by the Retention Manager during the purge list approval process is also exported.

Currently, no system or tenant audits are available for this functionality, only application audits are available.

### 9.10.4 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
- job instances

**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. 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. For more information, see [Running the Clean Up Purge Candidate Lists and Applications job](#).



**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.

#### **9.10.4.1 SIP application disposition**

Previously, if you applied retention to a SIP application, the system would not create individual dispose audits per package. Also, the system would not generate confirmations per package if configured on the holding.

Now, the disposition of the SIP application is blocked for the following reasons:

- If you did a receive command using IA Shell and not followed up with an ingest command
- If you requested an invalidation for a package, but still have not ran the Invalidation job.

- If you are using AGGREGATE mode and there are any open aggregates (this can be fixed by requesting a closing of the aggregate from the Package tab, run the Close job, and then run the Clean job ).
- If any hold was applied to anything in the application (this could be via a saved search, legal matter, direct hold application to package or search results).
- If longer retention was applied to anything in the application.



**Note:** A package being offline will not impede disposition of a SIP application. If you are disposing an application and you do not need confirmations per package, this can be configured for the holding using **Edit Confirmation**.

Disposition of a SIP application stops if there is any non-expired hardware retention on any content in the application.

When disposition job finishes, an order item is created that does the SIP dispose application data. This order item may run longer than the job and should be checked.



**Tip:** If the status of the order item does not change, navigate away and view the job history again.

Assuming the order item did not run into any issues, running the Clean job and then running the Clean up Purge Candidate Lists and Applications job is required for the application to be removed.

If the order item to clean up the SIP application is not created, resolve the issue by viewing the composite diagnostics logs. Then, run the Generate Purge List job again, approve it, and run the Dispose Purge Candidate List job.

### 9.10.5 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 canceled. It also required to run this job to complete application disposition.

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.

## 9.10.6 AIP disposition

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.

### 9.10.6.1 Immediately disposing AIPs

In most cases, data is disposed through the standard retention lifecycle. However, there can be situations when an AIP should be deleted immediately. The workflow depends on whether the package has been committed and is in the `Completed` phase or not. The described steps can be applied for multiple AIPs.

If the package has not been committed, meaning is not in the `Completed` phase and retention is not active:

1. On the **Packages** tab, invalidate the AIP. The package will change to the `INV_WPROC` state.
2. On the **Jobs** tab, run the Invalidation job. The package will change to the `INV_WDEL` state.
3. Run the Clean job. The package will be then deleted.

If the package has been committed and is in the `Completed` phase:

1. If there is a hold applied against the AIP, from on the Hold Sets tab, remove the hold.
2. To remove the AIP from a search scope, on the **Packages** tab, invalidate the AIP. The package will change to the `INV_WPROC` state.
3. On the **Jobs** tab, run the Invalidation job. The package will change to the `INV_WDEL` state.
4. Verify that the AIP is associated with a retention policy that has a Qualification Date prior to the current date (the Base Date is 90 days before the current date, in the event that the retention range is 89 days). Otherwise, the AIP will not be added to a purge list.
5. On the **Jobs** tab, run the `GeneratePurgeCandidateList` job. A new purge list will be created on the **Purge Lists** tab of the application.

6. If the purge list is valid, **approve** the list for disposition.
7. Configure the Dispose Purge Candidate List job so it will be scoped to the application and run the job. The package's state will change to `WAIT_CONF`. Wait until the job is finished with the `SUCCESS` status.

 **Note:** If no confirmation is set for the holding and the library is not online or, if only SIP confirmation is required and the SIP XML is on Glacier, the next steps can be skipped, as confirmation and the final disposition will be performed immediately, and the package will be deleted.
8. Run the Dispose Purge Candidate List job. The package will be then deleted.

#### 9.10.6.2 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.

#### 9.10.6.3 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. Therefore, it is better to apply the retention to the AIP or the AIU (records) but not both.

### 9.10.7 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.

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.

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 delete the search if it is in use by a policy or hold application. You are allowed to change the fields that are shown.

### 9.10.7.1 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. Run 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, run the Generate Purge List job (either clone and change to manual or start schedule).

Navigate to the **Purge Lists** 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 from the IA Web App. However, it is possible to Prepare Export per table and then, once the export has been done, only records that will be disposed are included in the export.

Refer to [Performing actions to a purge list](#) and [Exporting purge lists](#) for more information.

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 recommended that you clone the job and set to manual so you can run it when desired. Normally, disposition is done on a schedule.

In this case, only the items under hold will remain in the MASTER table.

### 9.10.7.2 Table record disposition

If all the records in table are eligible for disposition and approval is given, when running the Dispose Purge Candidate List job, the records will be processed. The table will not be destroyed even if all 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 Dispose Purge Candidate List 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.

### 9.10.7.3 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.

### 9.10.7.4 Table disposition and storage footprint

After running disposition on a table application, many records may have been deleted from their PostgreSQL database. PostgreSQL will not actually return the corresponding disk space to the Operating System until a PostgreSQL DB Administrator runs VACUUM FULL. Until then, you will not see PostgreSQL's disk usage decrease after disposition. Refer to PostgreSQL documentation on how to use VACUUM FULL.



#### Caution

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. Therefore, it is better to apply the retention to the table or the table rows but not both.

## 9.11 Retention, holds and disposition for tables with foreign key constraints and extracted text

Tables can have foreign key constraints (refer to [Table and column constraints](#) for more information). If a table (for example, `LINEITEMS` in the OrderManagement sample application) has a foreign key constraint with cascading deletes to another (for example, `ORDERS`), this affects how you can apply retention or holds to records of that table, and how disposition behaves.

To stick with the example, you will no longer be able to apply retention or holds to records of the `LINEITEMS` table directly: you will see in UI/ReST, that those options are no longer available. Instead, you will only apply retention or holds to records in the `ORDERS` table, which from then on, will implicitly protect the `LINEITEMS` records referring to these `ORDERS` records. In other words:

- No `LINEITEMS` record will get disposed unless its referred `ORDERS` record gets disposed, and
- Once an `ORDERS` record gets disposed, all `LINEITEMS` records referring to it, are disposed, as well.

The latter is actually done by PostgreSQL itself, as it is part of the cascading deletes setting on the foreign key constraint.

Similar restrictions apply to tables that are configured as target tables for extracted text (see [Storing extracted text](#) for more information). You cannot apply retention or holds to them. Records will be disposed when the corresponding unstructured content gets disposed.

Here is what is blocked for a table with foreign key relations with cascading deletes, and for a table with extracted text:

- Apply retention option for that table as a whole
- Apply hold option for that table as a whole
- Apply hold option for records of a search result for that table
- Adding a hold to a saved search on that table

Here is what will (partially) fail for a table with foreign key relations with cascading deletes:

- Run apply retention job with a search on that table
- Adding a hold to a matter with a saved search on that table (it will complete with a warning that not all holds could be applied)

## 9.12 Records Report

The Records Report in the Reports sample application provides an overview about the records per application, along with some basic information about retention.

---

### Available Records

Indicates the number of records available for searching.

For table-based applications, it is the number of records.

For SIP-based applications, it is the number of records in packages in the Committed phase.



**Note:** There may be a discrepancy between the number of records listed on the **Packages** tab and the **Records Report**. The Packages tab does not count records in an open aggregate until it is closed. Conversely, the Records Report counts records in an open aggregate.

---

### Under Retention

Indicates the number of available records that are under retention. The total does not count records that were invalidated.

For SIP-based applications, records within packages that have been rejected or are being purged are excluded even if they are under retention.

---

### Not Under Retention

Indicates the number of available records that are not under retention.

For SIP-based applications, records that have not be committed are excluded from this calculation. Only records that are available are considered.

---

### In Purge List

Indicates the number of records in a purge list. The total includes canceled purge lists that have then been invalidated. This could be a larger number than the records under retention.

Indicates the number of records in a purge list, including records that are inheriting retention from the package or table.

---

### Under Hold

Indicates the number of records under a hold, including records that are inheriting a hold from the package or table.

---

On the side panel, the following information is available. For table applications, the counts will be zero after the container fields:

As Of	When the data was gathered.
-------	-----------------------------

<b>Containers</b>	The number of containers for the application that contain records eligible for search.  For a SIP application, indicates the number of committed packages.  For table-based applications, indicates the number of tables in the application
<b>Retained Containers</b>	Number of containers that have retention applied. This count excludes packages that have been invalidated.
<b>Uncommitted Packages</b>	Number of packages that do not have any issues but have not been committed. This includes packages in one of the following phases: <ul style="list-style-type: none"> <li>• None</li> <li>• Reception</li> <li>• Waiting Ingestion</li> <li>• Ingestion</li> <li>• Waiting Commit</li> </ul>
<b>Rejected packages</b>	Count of packages in the Reject phase.
<b>Invalid packages</b>	Count of packages in the Invalid phase.
<b>Invalid packages retained</b>	Count of packages in the Invalid phase under retention. These packages will have to go through disposition. To remove the packages, they will have to go through disposition.
<b>Packages in purge</b>	Count of packages in the Purge phase. These packages are being disposed.
<b>Uncommitted Records</b>	Count of records in uncommitted packages, including records in packages in one of the following phases: <ul style="list-style-type: none"> <li>• None</li> <li>• Reception</li> <li>• Waiting Ingestion</li> <li>• Ingestion</li> <li>• Waiting Commit</li> </ul>
<b>Invalid records</b>	Count of records in packages in the Invalid phase.
<b>Invalid records retained</b>	Count of records in packages in the Invalid phase that are under retention. Either the individual items or packages will have to go through disposition.
<b>Rejected records</b>	Count of records in packages in the Rejected phase.
<b>Records being disposed</b>	Count of records in packages in the Purge phase. This may not be seen often but can happen if confirmation for a purge requires information that is not available.

### 9.12.1 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 job runs. 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:

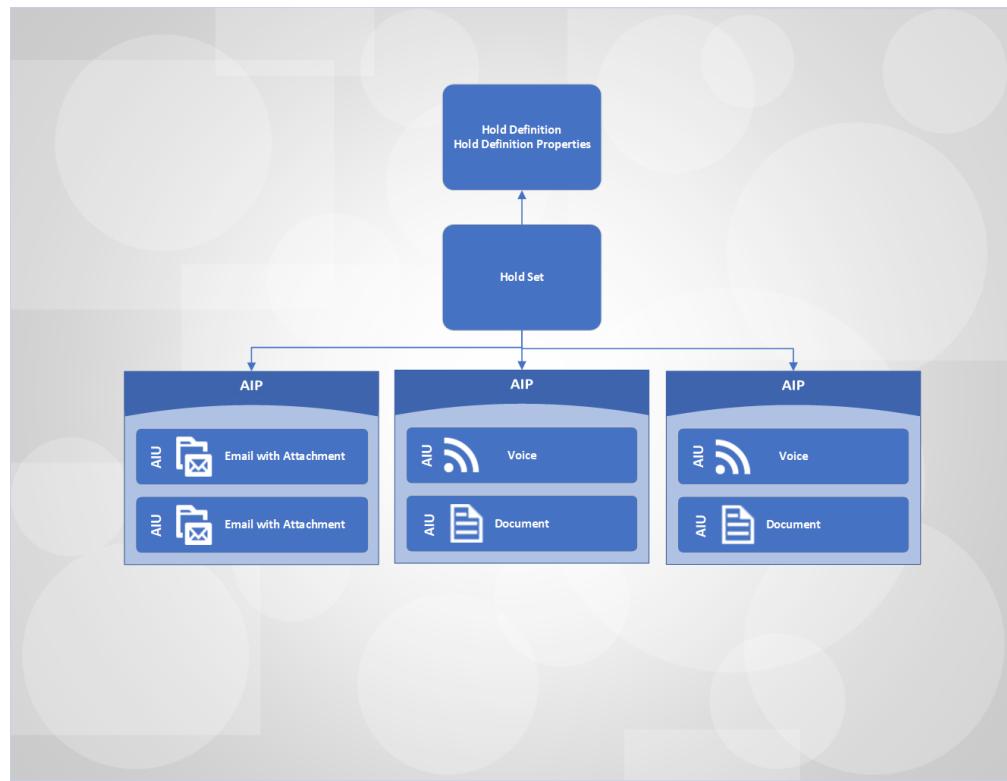
1. If the Refresh Metrics job has not run recently, run the job.
2. Select the **Reports** application.
3. In the list of search sets, select either:
  - **Overdue Dispositions:** Displays the total number of records that are overdue for disposition for at least six months for archived applications. 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.

## 9.13 Holds

OpenText Information Archive 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 several 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 bindings.
- 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.

### 9.13.1 Creating a hold

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:

---

**Hold Name**

The name of the hold.

---

**Description**

Provides a description of the hold.

---

**Type**

Indicates whether the hold is Legal or Permanent.

---

**Hold Approver**

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.

**To create a hold:**

1. On the **Compliance > Holds** tab, click +.
2. Enter the following information:

---

**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**.

**To edit a hold:**

1. In the **Compliance > Holds** tab, select the policy being edited and click the pencil button.
2. Edit the fields, as desired. For further field information, see [Creating a hold](#).
3. Click **Save**.

**To delete a hold:**

1. In the **Compliance > Holds** tab, click X for the hold being deleted.
2. When prompted to verify that you want to delete the selected hold, click **Delete**.

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:

**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:

- Legal
  - Permanent
- 

**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.

---

## 9.13.2 Applying a hold

There are different methods in which you can apply a hold to items:

- To the entire application.
- To a package.
- To a table.
- To a saved search.
- To a matter (converting to a legal matter).
- To individual records returned from a search result.
- To individual records using a job.
- To applying retention to search results (for more information, see [Applying retention to a table/applying a hold to records](#) section).

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.

### 9.13.2.1 Applying a hold to an application from the Application Info tab

The **Application Info** tab allows the Retention Manager to apply or remove retention policies/holds from an application. The Business Owner role can also access the **Application Info** tab but cannot perform any actions other than review the details of the selected application.

1. Select the desired application.
2. On the **Application Info** tab, click **Apply Hold**.
3. Select the hold 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. **Description:** Enter a description for the hold.
  - c. Click **Next**.
5. Review the information you have entered. When satisfied that the information is correct, click **Finish**.

### 9.13.2.2 Applying a hold to an AIP

Use the **Packages** tab to apply a hold to an AIP:

1. Select the AIP to which the hold is being applied.
2. Click the action button 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 background request completes.

### 9.13.2.3 Applying a hold to a table

Use the **Tables** tab to apply a hold to a table.

1. Select the table to which the hold is being applied.
2. Click the action button and select **Apply hold**.
3. Select the hold you want applied to the application and click **Next**.
4. Enter the following information:
  - **Hold Set Name:** Enter a unique name for the hold.
  - Enter a **Description** for the hold.
5. Click **Next**.
6. Review the information you have entered. When satisfied that the information is correct, click **Finish**.

This is an asynchronous operation. The table will not indicate that there is a hold policy applied to it until the background task completes.

### 9.13.2.4 Applying a hold to saved search results

A Retention Manager can apply a hold to the results of a saved search.

1. In the **Saved Searches** tab, click  for the search you are applying a hold to and select **Edit**.
  2. In the **Edit Saved Search** pop-up, click  in the **Holds** field.
  3. Select a hold from the list and click **Select**.
-  **Tip:** Use the **Search for a Hold** field to locate a specific hold.
4. Click **Save**.

A background request is created to apply the hold to the saved search.

### 9.13.3 Using legal matters to apply holds

OpenText Information Archive saved searches are typically grouped into matters for the collection and preservation of evidence in a legal investigation. In this case, holds are associated with a matter in addition to saved searches. This makes the matter a *legal matter*. In a legal matter, the associated holds are applied to each record in each of the saved searches of the matter and a hold set is created per saved search per hold.

#### 9.13.3.1 Creating legal matters

**To create a legal matter:**

1. Click the **Saved Searches** tab and then click the **Matters** sub-tab.
2. Click the **Create Matter** button on the **Matters** sub-tab.
3. In the **Create Matter** dialog, enter a **Name** for the matter.
4. **Optional** Enter a **Description** for the matter.
5. Click the **Select Holds** button next to the **Holds** field.
6. In the **Select Holds** dialog, select the holds you want to add to the matter and click **Select**. For help on creating holds, see “[Creating a hold](#)” on page 382.
7. Click the **Select Saved Searches** button next to the **Saved Searches** field.
8. In the **Select Saved Searches** dialog, select the saved searches you want to add to the matter and click **Select**. For help on creating saved searches, see Section 2.7.2 “[Creating a saved search](#)” in *OpenText Information Archive - End User Guide* (EARCORE-UGD).

- In the **Create Matter** dialog, click **Save** to have the selected holds applied to the records.



**Note:** A background request is created that appears in the **Background Requests** tab with a **Type of Update Holds on Matter**. While the background request is running, an icon appears next to the legal matter's **Name** on the **Matters** sub-tab. You can view details by clicking the icon.

- View the legal matter in the **Matters** sub-tab. For a description of the details on the **Matters** sub-tab, see Section 2.7.1.2 "The Matters sub-tab" in *OpenText Information Archive - End User Guide (EARCORE-UGD)*.



#### Important

Before proceeding to work with legal matters, read Section 2.7 "Creating and working with saved searches and matters" in *OpenText Information Archive - End User Guide (EARCORE-UGD)*.



**Note:** The background request can fail if one of its saved searches is still under construction, being rerun, or part of a legal matter for which there is currently a background request in process. To resolve a failure, wait for processing to finish and either retry the background request or edit the legal matter.



#### Important

When associating a saved search with a legal matter, only the records that were returned when the saved search was run will be put under hold. If you are doing live archiving, you may want to consider rerunning the search before associating it with the legal matter.

### 9.13.3.2 Working with legal matters

#### To filter matters in the Matters sub-tab by type:

Select one of the following options in the **Type** drop-down depending on what type of matters you want to view: **All**, **Matters**, **Legal Matters**.

#### To view matters in the Matters sub-tab only created or modified by you:

In the far-most right drop-down, select **Created by current user** or **Modified by current user**, as appropriate.

#### To view matters in the Matters sub-tab filtered by share status:

- In the **Share** drop-down list, select **All**, **Private**, **Shared by me**, or **Shared with me**, as appropriate.
- In the **Role** drop-down list, select **All**, **Editor**, or **Viewer**, as appropriate.

#### To view the saved searches associated with a legal matter:

- On the **Matters** sub-tab, click the **Expand** icon next to the matter's **Name**.
- To view a saved search in the **Saved Searches** sub-tab, click its **Name** in the expanded panel.

3. To view the saved search's results, click its name on the **Saved Searches** tab.

**!** **Important**

Retention Managers can see a saved search's results only if they are the creator of the saved search or the saved search is shared with them.

---

**To view the hold sets created by a legal matter:**

1. Click the **Applications** tab.
2. Select the application to which the legal matter's saved searches apply.
3. Click the **Hold Sets** sub-tab.
4. In the right-most filter drop-down, select **Show Hold Sets with Matters/Saved Searches Only**.

---

**To edit a legal matter:**

1. On the **Matters** sub-tab, select **Edit** in the context menu next to the **Name** of the legal matter you want to edit.
2. In the **Edit Matter** dialog, update the **Name** and **Description** as needed.
3. To remove a hold from the matter, click the **X** button next to the saved search's name in the **Holds** field.
4. To add a hold to the matter, click the **Select Holds** button next to the **Holds** field, select the hold in the **Update Holds** dialog, and then click **Update**.
5. To remove a saved search from the matter, click the **X** button next to the saved search's name in the **Saved Searches** field.
6. To add a saved search to the matter, click the **Select Saved Searches** button next to the **Saved Searches** field, select the search in the **Update Saved Searches** dialog, and then click **Update**.
7. In the **Edit Matter** dialog, click **Save**.



**Note:** If this operation results in holds being applied or removed from records, a background request is created to update the records referenced in the saved searches based on the new settings of the legal matter.

Editing a legal matter will be blocked or fail if there is a background request still running for the legal matter or if there is a background request running for a different legal matter that shares a saved search.

---

**To re-apply holds to a legal matter (in case of a suspected consistency/corruption issue):**

On the **Matters** sub-tab, select **Re-apply Holds** in the context menu next to the **Name** of the legal matter.

---

#### To delete a legal matter:

1. On the **Matters** sub-tab, edit the legal matter as above and remove all holds associated with the legal matter.  
This results in a background request that removes all holds from saved searches that were associated with the legal matter.
2. Wait for the background request to finish. When the background request finishes, the **Type** of the matter will change from **Legal Matter** to **Matter**.
3. On the **Matters** sub-tab, select **Delete** in the context menu next to the **Name** of the legal matter.
4. To delete saved searches associated with the matter, select the **Include Saved Searches** check box in the **Delete the Matter** dialog.  
Any saved searches that you do not have Edit access to or that are part of another legal matter will not be deleted and will be listed in the **Delete the Matter** dialog.
5. In the **Delete the Matter** dialog, click **Delete**.

---

#### 9.13.3.3 Sharing legal matters

Legal matters can be shared with groups and by access type: Viewer or Editor. For help with sharing legal matters, see Section 2.7.12 “Sharing saved searches and matters” in *OpenText Information Archive - End User Guide (EARCORE-UGD)*. Also see how to work with saved searches and matters in general by referring to Section 2.7 “Creating and working with saved searches and matters” in *OpenText Information Archive - End User Guide (EARCORE-UGD)*.

#### 9.13.3.4 Access and authorization rules for legal saved searches and matters

Section 2.7.13 “Access and authorization rules for saved searches and matters” in *OpenText Information Archive - End User Help (EARCORE-H-UGD)* explains access and authorization rules for non-legal saved searches and matters. Once a hold is added to a saved search, it becomes a legal saved search. Similarly, one a hold is applied to a matter, it becomes a legal matter and, in addition, all of its saved searches become legal saved searches.

For a saved search or matter to become legal, a user needs to apply the first hold. This user is always a Retention Manager.

Retention Managers can add holds to any non-legal saved search or matter they have edit rights to (meaning they created it or it was shared with them for editing).

Legal saved searches and matters have different access and authorization rules from non-legal ones. Basically, for any legal saved search or matter, all Retention Managers are Editors, whereas all other users who can access it are reduced to

viewing rights only. To better understand, refer to the general permission rules and the user types described in Section 2.7.13 “Access and authorization rules for saved searches and matters” in *OpenText Information Archive - End User Help (EARCORE-H-UGD)*.

To the list of general rules, the following is added:

- Legal saved searches and matters cannot be deleted. To delete them, first, they need to be made non-legal by removing all holds directly applied to them and for saved searches also indirectly applied to them through matters. The latter can be accomplished by removing the saved search from all its matters or removing all holds from all its matters.

To the list of four user types, the following fifth one is added:

- **Retention Manager:** A user who has the Retention Manager role and who is assumed to not also be the Creator or one of the sharees, and neither who has the Administrator role.

With these additions in mind, the access and authorization tables in Section 2.7.13 “Access and authorization rules for saved searches and matters” in *OpenText Information Archive - End User Help (EARCORE-H-UGD)* are amended as follows:

**Table 9-1: Legal saved searches**

	View	View search results	Edit*	Share	Remove items	Rerun	Delete
<b>Retention Manager</b>	Y	Y	Y	Y	Y	Y	N
<b>Creator</b>	Y	Y	N	N	N	N	N
<b>Viewer</b>	Y	Y	N	N	N	N	N
<b>Editor</b>	Y	Y	N	N	N	N	N
<b>Administrator</b>	Y	N	N	N	N	N	N

\*This includes adding/removing holds.



**Note:** With a legal saved search, any user who does not have the Retention Manager role is restricted to non-edit actions only.

**Table 9-2: Legal matters**

	View	Expand	Edit*	Share	Re-apply holds	Delete
<b>Retention Manager</b>	Y	Y	Y	Y	Y	N

	<b>View</b>	<b>Expand</b>	<b>Edit*</b>	<b>Share</b>	<b>Re-apply holds</b>	<b>Delete</b>
<b>Creator</b>	Y	Y	N	N	N	N
<b>Viewer</b>	Y	Y	N	N	N	N
<b>Editor</b>	Y	Y	N	N	N	N
<b>Administrator</b>	Y	Y	N	N	N	N

\*This includes adding/removing saved searches and holds.



**Note:** If a Retention Manager removes all holds from a legal saved search or matter, immediately all rules for the non-legal variants apply. In addition, the saved search or matter will remember the groups it was shared with for viewing/editing before any holds were added.

### 9.13.4 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 <IA\_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.

1. On the **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.



#### Caution

If you run 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 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.

2. Configure the job:
  - a. Select an application from the **Applications** list. To automatically apply a hold using a rule, the job must be scoped to a specific application.
  - b. Configure the following properties:

### Search Name and Search Set

Specify the name of the search and the search set (search composition). Once the job runs, the search results are passed into the rule, which automatically applied the hold.

### Search Criteria File

Specify information to narrow the scope of the search. You can either:

- Narrow the search scope within the rule itself by selecting criteria and doing evaluations based on the column data that is passed; or
- Enter the actual search criteria. For example:

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

For more information, see the example in Section 3.7.3 “Apply Retention Policy To Records job” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

### Rule Name

Specify the rule to be used for this job.

### Developer Mode

Select the **Developer Mode** box to test the rule. This outputs the results of the rule, but does not apply the hold to the object. Essentially, a test helps to ensure the data returned in the search is correct before actually running the rule.

Leave the **Developer Mode** blank if you do not want to test the rule.

3. Run the job. All the records returned in the selected search or search set will be put on hold. Navigate to the application’s **Hold Sets** tab to view the newly created hold set.

The rule itself looks like the apply retention rule.

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 **Search Name** and **Search Set** properties are set to a search with the status of Ready. Furthermore, the search criteria need 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).

1. In the **Jobs** tab, edit the Apply Hold Rule to Records job:
2. Select the **PhoneCallsGranular** application from the **Applications** list.
3. In the **Properties** section, enter the following:

**Search Name**

Enter FirstName\_Operator.

**Search Set**

Enter Set 1.

**Search Criteria File**

Enter <data></data>.

**Developer Mode**

Ensure that the **Developer Mode** box is not selected.

4. Click **Save**.
5. Run 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).

## 9.14 Removing holds

### 9.14.1 Viewing items in a hold set

When a hold is applied to records, the records are tracked as a hold set. The **Applications Hold Sets** sub-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:

**Hold Set Name**

The name of the hold set.

**Item Type**

Indicates the item that comprises the hold set (for example, application, package, table, etc.).

**Created Date**

The date the hold set was created.

**Associated Hold**

Indicates the hold applied to the hold set.

**Hold Type**

Indicates whether the hold is a legal or permanent hold.

**Items**

Indicates the number of items that are contained in the hold set.

**Review Date**

Indicates the date when the hold should be reviewed.

In the **Hold Sets** tab, click the arrow to the left of the hold set name you want to view. The information displayed depends on the Type of hold set:

**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.

- **To filter hold sets by associated hold:**

Click the **Associated Hold** drop-down and select the hold from the list.

- **To view hold sets *with* associated saved searches/matters:**

In the right-most filter drop-down, select **Show Hold Sets with Matters/Saved Searches only**.

- **To view hold sets *without* associated saved searches/matters:**

In the right-most filter drop-down, select **Show Hold Sets without Matters/Saved Searches**.

- **To export items in a hold set:**

1. Check the boxes for the items being exported or click **Select All** to select all the items contained in the hold set.

2. Click **Export Options**.

Exporting hold set data is an asynchronous request.

3. In the **Export in Background** pop-up, update the default **Request Name** information, if desired.

4. Click **Start Background Request**.

Navigate to the **Background Requests** tab to access the exported information.

## 9.14.2 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.

1. Select the application.
2. In the **Hold Sets** tab, click the action button.
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.

## 9.14.3 Removing a hold from an application from the Application Info tab

The **Application Info** tab allows the Retention Manager to apply or remove retention policies/holds from an application. The Business Owner role can also access the **Application Info** tab but cannot perform any actions other than review the details of the selected application.

### To remove a hold from an application:

1. Select the application.
2. In the **Application Info** tab, click  the hold set that you want to remove. The **Remove Hold** prompt is displayed.
3. In the **Reason** field, type a reason for the hold removal.
4. 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.

## 9.15 Running and viewing metrics with the compliance dashboard

Depending on your role, one of two possible dashboards is available to you:

- The administration dashboard
- The compliance dashboard

The compliance dashboard shows information about records, retention, holds and disposition. This dashboard is a good starting point for Retention Managers because you can perform many compliance tasks from this page.

To view the dashboard, Retention Managers can click the **Dashboard** tab. When Administrators click the **Dashboard** tab, they see the administration dashboard instead of the compliance dashboard. For more information about the administration dashboard.



**Note:** When you display the compliance dashboard, it may appear blank if the Refresh Metrics job needs to be run. To have the job run, contact your Administrator.

The compliance dashboard displays the following information:

- **Applications:** The number of applications, and how many are active archive applications versus decommissioned applications. Click **View** to go to the **Applications** listing page.
- **Retention Policies:** The total number of policies that have been created, and how many are currently in use. This section also shows how many of each type of retention policy has been created (fixed, duration, event, and mixed). Click **View** to go to the **Retention Policies** page.
- **Holds:** The total number of holds that have been created, and how many are currently in use. This section also shows how many of each type of hold has been created (legal or permanent). Click **View** to go to the **Holds** page.
- **Records Overview:** The total number of records that have been archived, and the state that they are in (under retention, under hold, the estimated records to dispose, and how many records are not under retention). You can hover your mouse over the bars in the graph to see exactly how many records are in each state.

### Under retention

The total number of records under retention. If a record has a retention policy applied more than once from different sources, it is counted once.

### Under hold

The total number of records under hold. If a record has any hold applied more than once from different sources (saved search, legal matter, inherited, or directly applied), it is counted once.

**Estimated records to dispose**

The total number of records in purge lists. If purge lists contain tables or packages, records contained inside are not part of this calculation. This also includes records in disposed purge lists where the records were not disposed (due to a hold).

**No retention**

Number of records that have no retention. Records are considered under retention if their container or the application is under retention.

- **Application Retention Coverage:** The total number of applications that have been created, and how many applications have full, partial, or no retention. To see a further breakdown of this information, click a section of the pie chart. For example, click the section of the chart for partial retention to view applications that have applied retention to some, but not all, of their records. Click the icon next to an application to perform some related tasks, such as viewing search forms or retention sets for the application. For detailed help, refer to [Viewing application retention coverage](#)
- **Application Purge Lists:** Select an application to view how many records are currently under review or require your attention. Click **View** to review the **Purge Lists** page for the selected application.
- **Upcoming for Disposal:** The record disposition projections for the next six months.



**Note:** The **As of** date and time on the upper-right corner of the dashboard refers to the date and time when the most recent Refresh Metrics job was run. The job affects the following dashboard panels: **Records Overview**, **Application Retention Coverage**, and **Upcoming for Disposal**. The information on the remaining panels is current as of the latest browser page load or refresh.

### 9.15.1 Viewing application retention coverage

The **Application Retention Coverage** panel on the compliance dashboard allows you to view which applications have full retention coverage, partial retention coverage, and no retention coverage. Applications with full retention coverage have retention applied to all its records. Applications with partial retention have retention applied to some of its records. And applications with no retention have retention applied to none of its records.

The panel's pie chart provides a visual breakdown of this information and allows you to view coverage information in the **Records Coverage** table to the right of the chart by clicking the corresponding section of the chart. In addition to displaying coverage information by application, the table provides access to retention-related tasks such as viewing an application's retention sets and holds. The table also allows you to drill down on an application to view a summary of its compliance with retention policies and holds.

**To view applications under a particular retention coverage status (full, partial, none):**

- Click the corresponding section of the pie chart.

You should now see the applicable applications and their coverage information in the **Records Coverage** table to the right. The table displays the following information for each application:

- The name of the application (**Application Name**)
- The number of the application's records that are retained (**Retained**)
- The total number of the application's records (**Total**)
- The percentage of the total records that are retained

**To update an application's retention compliance information:**

1. Click the application's icon located in the right-most column of the table.
2. In the list, click **Update**.

The update is run as a background request and you can view its status on the **Background Requests** page by clicking the **Background Requests** tab at the top of the page. For more details, see *To drill down and view a summary of an application's retention compliance* above.

**To drill down on an application and view a summary of its retention compliance:**

1. Click the application's link in the **Records Coverage** table.

If you have not updated the metrics for the application, data is not available. If no data is available, Click **Update**. This triggers a background request to update the application's compliance dashboard information. Once the job completes, navigate back to the application's compliance dashboard information to view the data.

Details of the application's retention compliance appear on the **Compliance Information** page. The page contains two panels. The **Effective Retention** panel displays the effective number of times each retention policy has been applied. The **Effective Holds** panel does the same for applicable holds.

2. To make sure you are viewing the latest information, click the **Update** button on the upper-right corner of the page.

Note that the page may appear empty if the update has never been run before by you or another Retention Manager. Note also that you can run the update directly from the **Records Coverage** table, as described below.

3. To include containers in the results of the **Retention Policies** panel, select the **Containers** check box in the upper-right corner of the panel.

For example, if retention is applied to all the packages, displaying containers will show the count of retained packages. Conversely, if retention is applied to

the records directly, the container count will be zero and the number of records that have retention applied will be shown.

**To view an application's search forms:**

1. Click the application's icon located in the right-most column of the table.
2. In the list, click **Search Forms**.

The search forms appear on the application's **Search Forms** tab. For more information on search forms, refer to [Search listing](#).

**To view an application's retention sets:**

1. Click the application's icon located in the right-most column of the table.
2. In the list, click **Retention Sets**.

The retention sets appear on the application's **Retention Sets** tab. For more information on retention sets, refer to [Applying retention to records](#).

**To view an application's hold sets:**

1. Click the application's icon located in the right-most column of the table.
2. In the list, click **Hold Sets**.

The hold sets appear on the application's **Hold Sets** tab. For more information on hold sets, refer to [Holds](#).

**To view an application's retention coverage:**

The Retention Manager can use the **Application Info** to review, apply, or remove retention against the selected application. The Business Owner role can also access the **Application Info** tab but cannot perform any actions other than review the details of the selected application.

1. Select the desired application.
2. Click **Application Info**.

## 9.15.2 Exporting data from the compliance dashboard to CSV format

You can use the **Export** button to export data from the compliance dashboard to CSV format (for example, for use in Microsoft Excel).

You download the exported data as a ZIP file, which contains three CSV files:

- `system-data-compliancetric.csv`: Contains system-level data that applies across applications (for example, how many records are under retention).
- `application-data-compliancetric.csv`: Contains data specific to individual applications (for example, how many records for an application have full retention).
- `application-summary-compliancetric.csv`: Contains data specific to individual applications (for example, retention or hold summary information for an application). The exported file only contains information if a Retention Manager requested an update of the compliance information for an application. This information may differ than data in the other exported file, as this information is updated on an ad hoc basis (refer to Viewing application retention coverage for more information).

---

### As Of

The date and time when the summary information in the row was last refreshed.

---

### Application

Name of the metric from the compliance dashboard.

---

### Type

Value for the metric.

---

### Name

Name of the retention policy or hold.

---

### Containers

Number of the containers.

---

### Records

Number of the records under the retention policy or hold.

---

### Applications

The total number of applications

---

### Active Archive

The number of active archiving applications

---

### Decommission

The number of application decommissioning applications

---

### Retention Policy

The total number of retention policies

**Retention Policy In Use**

The number of retention policies currently in use

**Fixed Retention Policy**

The number of retention policies with a fixed date aging strategy

**Duration Retention Policy**

The number of retention policies with a duration aging strategy

**Event Retention Policy**

The number of retention policies with an event aging strategy

**Mixed Retention Policy**

The number of retention policies with a mixed aging strategy

**Holds**

The total number of holds

**Holds In Use**

The number of holds currently in use

**Legal Holds**

The number of legal holds

**Permanent Holds**

The number of permanent holds

**Under Retention**

The number of records under retention

**No Retention**

The number of records not under retention

**Under Hold**

The number of records under hold

**Estimated Records to Dispose**

The total number of records to dispose across all applications, based on approved purge lists

**Upcoming for Disposal: <MONTH> <YEAR>**

The number of records that are eligible for disposition, regardless of whether they are in a purge list

The application-data-compliancemetric.csv file contains the following columns:

**Application Name**

The name of the application.

**Type**

Whether the application is an active archiving application (ACTIVE\_ARCHIVING) or application decommissioning application (APP\_DECOMM).

---

**Under Review (needs attention)**

The number of records in an application under review for disposition in purge lists. This is live data.

---

**Retained**

The number of records under retention for an application.

---

**Total**

The total number of records in an application.

---

Administrators can enable an audit that will allow them to see when the compliance dashboard was exported to CSV format and the audit was archived.

## 9.16 Enriching search results with compliance information

Retention managers have the ability to enrich a set of search results with compliance information (for more information, see [Adding columns](#)). This allows a Retention Manager to use column filtering and sorting to further analyze a search result.

1. Run the search. Assuming the search returned some results, the action will be available for retention managers.
2. Click **Enrich with Compliance**.
3. In the **Request Name** field, enter a name for the request that will run in the background.
4. Click **OK**.
5. Navigate to **Background Requests** and click **View** to review the updated search results. The compliance information is contained in the inline panel's **Compliance** tab.

The search results now include any additional columns and it is possible to see the date for when the search results were enriched. This information is at the time when the enrichment took place and only accurate at the time the enrichment was done. If it is known that compliance operations were done to the records, it is possible to re-enrich the search results.

It is also possible to use the column filters (as they are enabled automatically, if added) to filter the results. For more information, see Section 2.5.5 “Using a quick filter to narrow search results” in *OpenText Information Archive - End User Guide (EARCORE-UGD)*.

## 9.16.1 Example of how to update a sample application to use compliance columns

This section provides an example of how to configure a sample application's search results with enriched compliance information.

By default, retention manager permission was not provided for the CreditTresor application. Prior to completing the procedure below, ensure retention managers have permission for the CreditTresor application by performing one of the following:

- Disable the `permission.restrictRolesToApplicationGroups` using the **Global Settings** tab,
- Remove group permissions from the CreditTresor application using the **Permissions** tab, or
- Add the `GROUP_RETENTION_MANAGER` to the CreditTresor application using the **Permissions** tab.

This example assumes that in memory profile is enabled and the CreditTresor application has been installed.

### The Retention Manager should complete the following procedure:

1. In the CreditTresor application, run the **Find Customer by last name** search.
2. Once the results are displayed, click **Enrich with Compliance**.
3. **Optional** Enter a **Request Name**.
4. Click **OK**.
5. In the **Background Requests** tab, click **View** for the request you submitted. The compliance information is contained in the inline panel's **Compliance** tab.

## 9.16.2 Enriching saved searches with compliance

This section discusses saved searches and how to enrich them with compliance. For more information about saved searches, see Section 2.7 "Creating and working with saved searches and matters" in *OpenText Information Archive - End User Guide (EARCORE-UGD)*.

Retention Managers can enrich saved searches with compliance. The following scenarios are possible:

- A Retention Manager runs a search, enriches the results, and then creates a saved search.
- A Retention Manager creates a saved search, views the results, and then enriches the results with compliance.
- When another user saves a search and shares the results, the Retention Manager can view the results and enrich them with compliance.

Once a saved search is enriched, it remains enriched for any operation on the saved search, including adding or removing items or changing holds for the saved search, or adding or removing items for a legal matter. The system re-enriches the saved search if the search is either reloaded or rerun. If a saved search was created from an enriched search result, the **Create: Saved Search** audit indicates in the supplemental data if it was enriched and also includes information about any filter or criteria used for the saved search.

If a search result associated with saved search is enriched, the name of the saved search is available in the supplemental data fields of the audit.

If a saved search is shared with a Retention Manager, who subsequently enriches the search results, the information is available to anyone who has access to the search, along with the date when the search result was last enriched. This also includes being able to use advanced filtering on the compliance information. Advanced filters for compliance columns start with the `compliance_` prefix.

**!** **Important**

For both SIP and table applications, if a search result is enriched for compliance, you must rerun the search and then enrich the search result. The reason is because, on upgrade, any compliance columns defined for a search are automatically updated to add the `compliance_` prefix to avoid conflicts with other fields. If this is not done, the compliance fields will be blank and, if you use an advanced filter, it may not provide the expected results. After any change is made to a search, column filtering or advanced filtering on fields defined in the search will not work for the changed fields. The fix is to rerun the search so that the search result contains the required information.

You should also rerun any saved search that has compliance columns defined. If the saved search is part of a matter, rerunning the matter results in all saved searches contained in the matter to be rerun.

## 9.17 Exporting compliance information

### 9.17.1 Configuring an export configuration for compliance information

The following procedure is only supported for exporting search results containing records from SIP applications. To review the compliance information that is exported, see [Default export configurations](#).

Export configurations can be either be shared at the tenant-level or per application. You can edit or create an existing export configuration.

- For tenant-level export configurations, it can be found in the **Administration** tab's **Export Configuration** tab.
- For an application's export configuration, access the application and navigate to the **Export Configuration** tab.

To edit either a tenant-level or application-level export configuration, click the  icon in the side panel.

In the **Options** tab, see if there is an entry for `includeCompliance` in the **Key** column.

**If the `includeCompliance` option does not already exist:**

1. Click **+ Add Option**.
2. Enter the following:
  - a. For the **Key** field, enter `includeCompliance`.
  - b. For the **Value** field, either set to `all` or `min`. To learn which data is exported for these values, see [Creating export configurations at the tenant and application levels on page 279](#).
3. Click **Save**.

## 9.17.2 Exporting retention and hold sets

You cannot export a retention or hold set unless it contains records.

1. Depending on which data is being exported, navigate to the **Retention Sets** or **Hold Sets** tab.
2. Expand one of the retention or hold sets to view the items that are in the selected set.
3. Select the item or items that you want to export and click **Export Options**.
4. The **Export in Background** dialog box is displayed. If desired, enter a **Request Name** or keep the name that has been automatically generated. Click **Start Background Request**. Navigate to the **Background Requests** tab to access the exported data.

# 9.18 Audits

## 9.18.1 Using audits for compliance

OpenText Information Archive provides the following functionality for auditing:

- 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, which is typically defined in the supplemental data.

Changes made to a retention policy or hold create an audit at the tenant level. Any attributes that are updated are saved in the supplemental data of the audit.

Once a change is made, run the Archive Audits job. Then access the Audit sample application and run a Tenant Audit search. Select the applicable row to view the changes made to the retention policy or hold.

It is possible to view the previous attribute values by using the Tenant Audit search in the Audit sample application. Once the results are displayed, apply filters on the event type and event source. For example, to view all the changes to a retention policy, run a search scoping for a specific retention policy. Then apply a filter to the Event Source column to view the retention policy by its name. If you sort the Created Date column, you can see the history of each change to the retention policy and view its supplemental data.

### **9.18.1.1 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 event.

OpenText Information Archive includes a fixed list of audits stored in a constants file and uses a REST to retrieve the audits.

### **9.18.1.2 Application-specific audits**

OpenText Information Archive 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 is not set, then the audit is for all applications that include that 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.

### 9.18.1.3 Searching for audits

Audits are archived into the system as AIPs, which allow the customer to perform searches of the audit entries.

OpenText Information Archive provides three searches for searching system, tenant, and application audits.

The search forms have been enhanced to leverage the new filter functionality and the default sort is a descending sort on the created date.

### 9.18.1.4 Audit entries

Audits can be configured through the OpenText Information Archive 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	None
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.
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.

Name	Description	Notes
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.
Partial Dispose	Audit for if a resource is partial 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 back-up is taken. In the case of the table, the back up is done at the schema level. A back-up store must be defined for the application (the sample applications demonstrate how this is done). For each database or holding, a back-up 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	None

## 9.18.2 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 run a search against it. The objects are collected in a temporary storage until they are archived. The Archive Audit job must run 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.

For more information about configuring the parameters, see the Section 3.7.5 “Archive Audits job” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

When running this job, the audit for apply retention for the audits log will be archived in the next run.

This job can be scoped to an application, but it is recommended to keep it system-scoped, otherwise the system audits will not be archived.

## 9.18.3 Configuring Apache Kafka Producer Client

IA Server can be configured to be a real-time event producer for Kafka. It reacts on events immediately without waiting to archive the audits.

This feature could allow you to externally monitor OpenText Information Archive (for example, to know when a job fails, when a purge candidate list is awaiting for approval, or to track login failure).

### 9.18.3.1 Event format

The events are based on audit information. To generate an event, the corresponding audit must be enabled in OpenText Information Archive. The events are formatted based on the following table:

Field Name	Field Value	Type	Example
Topic	application.yml/\${kafka.producer.defaultTopic}	String	com.opentext.ia
Timestamp	\${audit.createdDate}	Timestamp	1673377348796 (2023-01-10T19:02:28.796+01:00)
Key	\${audit.id}	String	81259098-67b7-46b8-8a4d-356fb82ce878

Field Name	Field Value	Type	Example
Value	\$ {audit.supplementalData}	JSON	{           "searchId": "8589d25e-5fbb-42c4-b672-bab5ea6973df",           "searchName": "Invoice Number",           "searchType": "PRIMARY",           "queryId": "ddaa6485-510b-4870-bf2f-c83f398e2e6f",           "queryName": "Invoices-query",           "resultSchema": "urn:ea-samples:en:xsd:invoices.1.0",           "searchCompositionId": "90496bc5-0e02-4dcb-8602-aa93a1d21ec4",           "searchCompositionType": "PRIMARY",           "executionTime": "244",           "recordCount": "1",           "recordCacheCount": "1",           "async": "false",           "executionTime::sdp": "13",           "executionTime::query": "68",           "executionTime::cache": "68",           "executionTime::page": "94",           "aipCount": "2",           "aiuCount": "26",           "libCount": "2"         }

Field Name	Field Value	Type	Example
Headers *	createdBy	Bytes	emma@iacustomer.com
	eventType	Bytes	search
	eventName	Bytes	search
	eventSource	Bytes	Invoice Number
	auditedObjectId	Bytes	8589d25e-5fbb-42c4-b672-bab5ea6973df
	tenantId	Bytes	47003807-1386-4a79-81c8-7f8dc42ab714
	applicationName	Bytes	Invoices
	applicationId	Bytes	e60ec457-54c4-4fa3-a198-5f86fb5bb14b
	transactionId	Bytes	d5d21c7e-7a3f-4d5d-a1ef-775e072ae8fd

\* Using headers will allow fine filtering for customers.

### 9.18.3.2 Configuration

You need to update IA Server's application.yml file to declare a new profile and to provide connection settings.

Depending on your Kafka cluster/broker, the way to configure the settings may differ. It is important to know that OpenText Information Archive only supports these security protocols: PLAINTEXT, SSL (un-authenticated), SASL\_PLAINTEXT, SASL\_SSL, and these SASL mechanisms: PLAIN, SCRAM-SHA-256, SCRAM-SHA-512.

The following is a list of supported settings:

Key	Mandatory	Password	Comment	Example
kafka.producer.bootstrapServers	Yes		A list of host/port pairs to use for establishing the initial connection to the Kafka cluster.	localhost:9092
kafka.producer.defaultTopic	No		Topic name where to publish the events.	com.opentext.infoarchive (default)
kafka.producer.clientId	No		ID to pass to the IA Server when making requests. Used for server-side logging.	infoarchive-server-1 (default)
kafka.producer.securityProtocol	No		Protocol used to communicate with brokers. Valid values are PLAINTEXT, SSL, SASL_PLAINTEXT, SASL_SSL.	PLAINTEXT (default)
kafka.producer.sasl.mechanism	No		SASL mechanism used for client connections. Valid values are PLAIN, SCRAM-SHA-256, SCRAM-SHA-512.	PLAIN
kafka.producer.sasl.jaasConfig	No	Yes	JAAS login context parameters for SASL connections in the format used by JAAS configuration files.	org.apache.kafka.common.security.plain.PlainLoginModule required username="alice" password="alice-secret";

Key	Mandatory	Password	Comment	Example
kafka.producer.sasl.clientCallbackHandlerClass	No		The fully qualified name of a SASL client callback handler class that implements the AuthenticateCallbackHandler interface.	
<b>Additional settings</b>				
kafka.producer.lingerMs	No		The producer groups together any records that arrive in between request transmissions into a single batched request. Normally this occurs only under load when records arrive faster than they can be sent out. However in some circumstances the client may want to reduce the number of requests even under moderate load. This setting accomplishes this by adding a small amount of artificial delay—that is, rather than immediately sending out a record, the producer will wait for up to the given delay to allow other records to be sent so that the sends can be batched together.	
kafka.producer.bufferMemory	No		The total bytes of memory the producer can use to buffer records waiting to be sent to the server.	33554432 (default)

Key	Mandatory	Password	Comment	Example
kafka.producer.compressionType	No		The compression type for all data generated by the producer. The default is none (for example, no compression). Valid values are none, gzip, snappy, lz4, or zstd. Compression is of full batches of data, so the efficacy of batching will also impact the compression ratio (more batching means better compression).	none (default)
kafka.producer.retries	No		Setting a value greater than zero causes the client to resend any record whose send fails with a potentially transient error. Note that this retry is no different than if the client resent the record upon receiving the error.	0 (default)
kafka.producer.acks	No		The number of acknowledgments the producer requires the leader to have received before considering a request complete. [all, -1, 0, 1]	all (default)

Key	Mandatory	Password	Comment	Example
kafka.producer.batchSize	No		The producer will attempt to batch records together into fewer requests whenever multiple records are being sent to the same partition. This helps performance on both the client and the server. This configuration controls the default batch size in bytes.	16384 (default)
kafka.producer.ssl.keyPassword	No	Yes	The password of the private key in the key store file or the PEM key specified in ssl.keyStoreKey.	
kafka.producer.ssl.keyStoreCertificateChain	No		Certificate chain in the format specified by ssl.keyStoreType.	
kafka.producer.ssl.keyStoreKey	No		Private key in the format specified by ssl.keyStoreType	
kafka.producer.ssl.keyStoreLocation	No		The location of the key store file	
kafka.producer.ssl.keyStorePassword	No	Yes	The store password for the key store file	
kafka.producer.ssl.keyStoreType	No		The file format of the key store file	
kafka.producer.ssl.trustStoreCertificates	No		Trusted certificates in the format specified by ssl.trustStoreType	
kafka.producer.ssl.trustStoreLocation	No		The location of the trust store file.	
kafka.producer.ssl.trustStorePassword	No	Yes	The password for the trust store file.	

Key	Mandatory	Password	Comment	Example
kafka.producer.s sl.trustStoreType	No		The file format of the trust store file	

#### Local Kafka Server:

```
spring:
  profiles:
    include:
      - infoarchive.ias.profile.kafka
kafka:
  producer:
    bootstrapServers: localhost:9092
    defaultTopic: com.opentext.infoarchive
    clientId: infoarchive-server-1
    securityProtocol: PLAINTEXT
```

#### Azure Event Hubs:

```
spring:
  profiles:
    include:
      - infoarchive.ias.profile.kafka
kafka:
  producer:
    bootstrapServers: xxx.servicebus.windows.net:9093
    defaultTopic: com.opentext.infoarchive
    clientId: infoarchive-server-1
    securityProtocol: SASL_SSL
    sasl:
      mechanism: PLAIN
      jaasConfig: org.apache.kafka.common.security.plain.PlainLoginModule required
      username="$ConnectionString" password="Endpoint=sb://
      xxx.servicebus.windows.net/;SharedAccessKeyName=XXXXXXXXXX;SharedAccessKey=F/
      XXXXXXXXXXXXXXXX/XXXXXX/XXXXX=";
```

#### Confluent Cloud:

```
spring:
  profiles:
    include:
      - infoarchive.ias.profile.kafka
kafka:
  producer:
    bootstrapServers: xxx-xxxx.xxxxxxx.gcp.confluent.cloud:9092
    defaultTopic: com.opentext.infoarchive
    clientId: infoarchive-server-1
    securityProtocol: SASL_SSL
    sasl:
      jaasConfig: org.apache.kafka.common.security.plain.PlainLoginModule required
      username='XXXXXX' password='XXXXXXX';
      mechanism: PLAIN
```

### 9.18.3.3 Enabling the Kafka profile

The profile `infoarchive.ias.profile.kafka` must be declared in the IA Server's `application.yml`.

```
spring:  
  profiles:  
    include:  
      - infoarchive.ias.profile.kafka
```

When Kafka is enabled, the configuration will be tested during server startup.

`iaserver.log` (success):

```
Starting IA Server v23.2 using Java 11.0.12  
...  
The following profiles are active: infoarchive.ias.profile.kafka  
Enabled kafka handling  
...  
Verifying kafka connectivity  
...  
Started IA Server in 13.508 seconds (JVM running for 13.867)
```

If Kafka is not reachable, the credentials are not correct and/or the topic does not exist, the server will fail to start:

```
Starting IA Server v23.2 using Java 11.0.12  
...  
The following profiles are active: infoarchive.ias.profile.kafka  
...  
Enabled kafka handling  
...  
Verifying kafka connectivity  
Message with key '8e2661b3-3f08-4fc0-98f3-197f064ef2bf' for topic  
'com.opentext.infoarchive' was not sent due to an exception: Topic  
com.opentext.infoarchive not present in metadata after 60000 ms.  
Application run failed  
java.lang.IllegalStateException: Failed to connect to Kafka Server. Please check the  
corresponding configuration in application.yml  
    at com.emc.ia.kafka.KafkaService.startup(KafkaService.java:67)  
    at  
com.emc.ia.main.config.impl.KafkaConnectionChecker.onApplicationEvent(KafkaConnectionChec  
ker.java:37)  
    at  
com.emc.ia.main.config.impl.KafkaConnectionChecker.onApplicationEvent(KafkaConnectionChec  
ker.java:16)  
    at  
org.springframework.context.event.SimpleApplicationEventMulticaster.doInvokeListener(Simp  
leApplicationEventMulticaster.java:176)  
    at  
org.springframework.context.event.SimpleApplicationEventMulticaster.invokeListener(Simple  
ApplicationEventMulticaster.java:169)  
    at  
org.springframework.context.event.SimpleApplicationEventMulticaster.multicastEvent(Simple  
ApplicationEventMulticaster.java:143)  
    at  
org.springframework.context.support.AbstractApplicationContext.publishEvent(AbstractAppli  
cationContext.java:421)  
    at  
org.springframework.context.support.AbstractApplicationContext.publishEvent(AbstractAppli  
cationContext.java:378)  
    at  
org.springframework.context.support.AbstractApplicationContext.finishRefresh(AbstractAppl  
icationContext.java:938)  
    at  
org.springframework.context.support.AbstractApplicationContext.refresh(AbstractAppl  
icationContext.java:586)
```

```

at
org.springframework.boot.web.servlet.context.ServletWebServerApplicationContext.refresh(S
ervletWebServerApplicationContext.java:147)
    at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:731)
    at org.springframework.boot.SpringApplication.refreshContext(SpringApplication.java:
408)
        at org.springframework.boot.SpringApplication.run(SpringApplication.java:307)
        at
org.springframework.boot.builder.SpringApplicationBuilder.run(SpringApplicationBuilder.ja
va:164)
            at com.emc.ia.main.impl.Main.run(Main.java:44)
            at com.emc.ia.main.impl.Main.main(Main.java:28)
Caused by: org.springframework.kafka.KafkaException: Send failed; nested exception is
org.apache.kafka.common.errors.TimeoutException: Topic com.opentext.infoarchive not
present in metadata after 60000 ms.
        at org.springframework.kafka.core.KafkaTemplate.doSend(KafkaTemplate.java:666)
        at org.springframework.kafka.core.KafkaTemplate.send(KafkaTemplate.java:429)
        at com.emc.ia.kafka.KafkaService.startup(KafkaService.java:65)
        ... 16 common frames omitted
Caused by: org.apache.kafka.common.errors.TimeoutException: Topic
com.opentext.infoarchive not present in metadata after 60000 ms.

```

## 9.19 Caching in packages that contain records for retention sets and hold sets

The **Retention Sets** and **Hold Sets** tabs allow a Retention Manager to cache in packages that are part of a retention set or hold set. When you expand a particular retention set or hold set, a message appears if the set contains items that are offline and, therefore, cannot be viewed.

### To cache the records in:

1. Click the available link to cache the records in. This creates a background request.
2. On the **Background Requests** page, click **View** for the request that was created. The packages are now online and can be viewed.

If there are multiple pages of records and you attempt to navigate to the next page of records, some of these items might be offline. Repeat steps 1 and 2 to bring these records online.



**Tip:** It is possible to audit when a Retention Manager uses this feature. By default, this audit is not enabled. To enable the audit, for the event category named **Other**, select Cache In box for the Retained Set and/or Hold Set event types.

The audit includes two forms of supplemental data, a context that indicates the page and the page size, and another that indicates the packageCount

This audit is created when a request is made to cache a package in. Even if the request fails, for example, because the store is offline, the audit is still created. You can tell the operation was successful, you will see individual cache-in audits for the libraries that were cached in (those audits are, by default, enabled and could also be disabled).

## 9.20 Compliance troubleshooting

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".

While this message is rarely generated, if it is issued it means either:

- Another Retention Manager rejected and then cancelled the purge candidate list; or
- The Generate Purge List job is running too frequently.

### Why was a purge list not created?

- Check to see if the retention policies have been disabled for disposition processing.
- Check to see if the qualification date has been met.
- Check to see if the item has a hold directly applied (it will not be eligible). Also, check if the parent is under hold.
- Check to see if longer retention has been applied to the parent or application.

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.

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 exports the purge list (the package), the export will not contain any records that were in the invalidated package. Furthermore, if a confirmation was configured for PURGE on the holding, the confirmation will not include records that were invalidated.
- 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 runs 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.

If you have modified a retention policy and disabled disposition processing, records using that policy will not be put into purge lists until the retention policy is modified to re-enable disposition processing.

When applying hold or retention to records, ensure that the retention store is specified. Refer to [Using jobs to apply retention](#) for more information.

---

**Expected items are not included in a purge list.**

Check if the retention policy associated with the item(s) is disabled.

---

**Unable to dispose SIP application.**

Ensure that if a query is provided for the PURGE confirmation (defined on the holding) that it is correct; otherwise, failure to generate the confirmations for the packages will prevent the order item from disposing the package.

If there are any open aggregates, disposition of a SIP application will be skipped. To resolve this issue, request closing of the aggregates, run the Close job, and then run the Clean job.

If there are any packages that have been invalidated, but the Invalidation job has not run, this will prevent disposition. To resolve this issue, run the Invalidation job.

If there are any packages that have been received, but not yet ingested, either invalidate the package (and run the Invalidation job) or complete the ingestion procedure using either the IA Web App or IA Shell.

If there is any non-expired hardware retention on any content in the application, check the composite diagnostic logs for information about what the non-expired date was. Wait for the date to pass before trying the procedure again.

---



# Chapter 10

## Improving 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 derive from external factors rather than OpenText Information Archive 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.

## 10.1 Specifying the number of batches

While it was previously possible to configure a batch size for individual batch operation types in the `application.yml` file, now you specify the preferred number of batches (optionally per batch operation type) instead.

The batch size continues to be used internally, but instead of it being statically configured in the `application.yml` file, it is now dynamically calculated based on the total number of items for the operation and the preferred number of batches to create. A minimum batch size is ensured that is equal to the chunk size of the operation type to avoid multiple unnecessarily small batches instead of less larger ones.

This affects the `batch.nrOfBatches.*` settings in the **Global Settings** tab.

The purpose of this change is to significantly limit the number of batches created, especially for large operations. Consequently, this increases overall system responsiveness of/during batch runs, especially with multiple large batch operations running concurrently and makes identifying the appropriate configuration values easier. Typically, the number of batches running in parallel is a more understandable metric that is easier to predict system load than the size of individual batches with no control or insight in advance over how many would end up being created.

Unfortunately, just like for batch size, there is no golden rule as to what the ideal value is, since the total number of batches that can run in parallel is equal to the number of threads available for batch runs across the environment. A single OpenText Information Archive node does not have the necessary information to calculate this value and multiple batch operations could end up running in parallel, each of which would create up to the specified number of batches. The best value, therefore, depends on the environment and expected workload. Some experimentation might be required.

## 10.2 Improving the response time for a background request

In the IA Server, there is a default pool used to run background requests such as jobs, background searches, exports, and any other request stemming from manual interaction (request for a cache out of a library, *etc.*). All background requests use this same pool. This pool will run only background requests with a priority set to 0.

By default, there is also a second pool called “priority pool” that is used to run priority background requests. By default, the searches and export background requests are run with priority 100, and the priority pool considers the background request from 1 to 1,000. Only the search and export background are run by this pool and have more priority. This priority pool can be customized to prioritize specific background requests.

There is a priority order item pool run to improve the response time of background requests. A corresponding priority batch item pool run is also configurable.

The number of threads affected to the priority order item and respective priority batch item can be changed by updating the following properties of the IA Server's application.yml file:

- background.numberOfThreads.priorityOrderItems and, respectively,
- background.numberOfThreads.priorityBatchItems

This feature can be disabled by setting `numberOfThreads.priorityOrderItems` to 0.

It is also possible to customize the priority range of the priority pool. By default, the pool considers the range from 1 to infinite. Depending on what is required, it is possible to set a custom range by modifying the IA Server's application.yml file.

For example, if you need to prioritize the Clean job, you have to update the priority setting of the job (for the purposes of this scenario, let us choose 200). Once done, you can set the following:

- background.priorityRange.min:195
- background.priorityRange.max:205

In this case, only the background request with a range between 195 and 205 is run by the priority pool. The Clean job of priority 200 is the only job to run with this pool. Be aware that, in this example, the background searches are not run anymore in the priority pool because, by default, the priority is 100. You can also make an adjustment on the default priority search value by modifying the default value in global settings.

```
background:  
  priorityRange:  
    min: 195  
    max: 205  
  numberOfThreads:  
    orderItems: 10  
    # Additional threads to process priority order items only  
    priorityOrderItems: 5  
    batchItems: 10  
    priorityBatchItems: 5  
    jobItems: 5
```

## 10.3 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 OpenText Information Archive'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:

---

#### PDI file size

1. Might be the limit of the size of a document.
  2. Configure cache-out to keep only the amount of data you want.
- 

#### SIP file size

1. Can be limited by the customer environment, as the SIP might have to be copied through its network.
  2. The SIP file with structured content is less than 1 GB is a common case that uses batch ingestion.
  3. Ingestion time is not a criterion 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.
  4. 10 MB SIP is a small-sized SIP. For batch ingestion, 1 GB SIP for unstructured data and 100 MB SIP for structured data are not uncommon.
- 

#### Number of AIUs in a SIP

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 million 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 partitioning and indexes.
  2. Having too few records in each SIP item can slow down the search, and even having 10 to 40 is still too few. Several tens of thousands of AIUs are adequate for structured content. Hundreds of thousands are not uncommon for pure unstructured data. Having fewer records is only acceptable if doing synchronous ingestion and using aggregate mode, when the AIPs will be aggregated into larger ones after some time has elapsed.
- 

#### Number of SIP files in a holding

The only limitation is how long it will take to run the partitioning query. If indexes are created for 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 depend on the 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 how you are expecting SIPs to be sized.
- Ensure you use the optimal ingestion mode (private, aggregate, or pooled).
- Check the partitioning key strategy. OpenText Information Archive search is a two-tiered search. The first tier selects a subset of AIPs based on the defined partitioning keys. The second tier runs 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 runs 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 estimating the number of AIPs returned by the first tier for the given search criteria.

- Ensure that structured data encryption is not used unless needed for sensitive information.
- Avoid using full-text search unless it is required, as it has poor performance.
- To investigate search performance, enable the search debug option directly on the search. To investigate ingestion performance, set DEBUG to the logLevel field of the ingestNode holding configuration, and download the ingest.log attached to the AIP.
- The AIPs are offline so it takes longer to bring them online.

### 10.3.1 Improving ingestion speed

Ingestion speed depends on:

- The SIP size, including the percentages of structured and unstructured data
- Indexes
- Number of AIUs per SIP
- Network speed
- Target storage
- Working directory storage

The IA Server logs can be found in the <IA\_ROOT>/logs directory and can be used to determine how to improve system performance. For example:

- Only enable encryption on the holding, if required. Encryption can be set independently for metadata fields or for the unstructured content.
- If you do not require CI hashing, then remove/disable CI hashing configuration in the PDI file or use a faster hash algorithm instead of SHA3-224.

- Use a Solid State Drive (SSD) for the working directory (change the path in the application.yml file).

### 10.3.2 Prevent loading table data twice

It is possible to ingest the same table document multiple times by mistake.

To prevent this issue, you can add a unique document ID to each table document. During ingestion, the document ID of each table document is checked before being stored. If the document ID is already stored, an exception is thrown.

When ingesting a batch of documents, only table documents that have duplicate IDs will fail.

The unique ID of a table document must be stored in an attribute with the name documentId and namespace urn:x-emc:ia:schema:table on the document element. For example:

```
<TICKETS xmlns:table="urn:x-emc:ia:schema:table" table:documentId="1">
  <CUSTOMER>
    <ROW>
      ....
```

### 10.3.3 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.

## 10.4 Lucene libraries configuration cache

To improve search performance, it is possible to configure the IA Server to keep in memory Lucene library configuration objects to speed up the access to the Lucene segments on the disk. This cache is enabled by default for synchronous searches and disabled on loading for background searches. This means that background searches use the cache to speed up access to configuration objects of libraries already cached by a synchronous search, but will not load the uncached libraries they need into the cache.

To define maximum capacity, consider the memory allocated per IA Server and the average size of file descriptor of 50 KB (500 MB for 10,000 libraries).

Default values can be changed via the [Global Settings](#) tab.

Global Settings	Description	Default Value
search.lucene.cache.enabled	If true, enables caching of Lucene file descriptors.	true

Global Settings	Description	Default Value
search.lucene.cache.enabledAsync	If true, caches Lucene file descriptors during a background search when the caching is enabled.	false
search.lucene.cache.maximumCapacity	Number of Lucene libraries whose file descriptors should be put in the cache.	10,000
search.lucene.cache.refreshTimeout	Time in seconds to refresh cached Lucene file descriptors after last write.	60

## 10.4.1 Viewing Lucene cache statistics

You can configure the config/iaserver/application.yml to expose Lucene cache statistics:

```
spring:
  cloud:
    config:
      enabled: false
  jmx:
    enabled: true
```

Use a JMX client to view the current metric data. Lucene cache statistics are exposed through JMX as com.opentext.ia.management:name=LuceneIndexCache.

The statistics contain the following:

### AverageLoadPenalty

Returns the average number of nanoseconds spent loading new values. This is defined as `totalLoadTime / (loadSuccessCount + loadFailureCount)`.

### EvictionCount

Returns the number of times an entry has been evicted. This count does not include manual invalidations.

### HitCount

Returns the number of times cache lookup methods have returned a cached value.

### HitRate

Returns the ratio of cache requests that were hits. This is defined as `hitCount / requestCount`, or 1.0 when `requestCount` is set to 0.

### LoadCount

Returns the total number of times that cache lookup methods attempted to load new values. This includes both successful load operations and those that threw exceptions. For every cache entry that has been loaded for more than 60 seconds, the server quickly verifies that the version matches. If it does not match, a new Lucene file descriptor is opened to replace the outdated one. This check counts as a load even if the entry does not need to be updated.

**MissCount**

Returns the number of times cache lookup methods have returned an uncached (newly loaded) value, or null. Multiple concurrent calls to cache lookup methods on an absent value can result in multiple misses, all returning the results of a single cache load operation.

**MissRate**

Returns the ratio of cache requests that were misses. This is defined as `missCount / requestCount`, or 0.0 when `requestCount` is set to 0.

**RequestCount**

Returns the number of times cache lookup methods have returned either a cached or uncached value. This is defined as `hitCount + missCount`.

**TotalLoadTime**

Returns the total number of nanoseconds the cache has spent loading new values.

## 10.5 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.

If searches conducted in a table archive are slow, complete the following to improve table application search performance:

- Make sure indexes are in place;
- 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.

## 10.6 Importance of the IO performance of the storage systems

Our testing has revealed that the Lucene stack around 50% more IO intensive when compared to the legacy xDB stack. When you define the storage systems to store structured data (SIP only) and search result, some of the below are recommended:

- Use a dedicated storage device to handle Lucene segments
- Use a storage device that can support the additional IO requirements of the Lucene stack.
- Split the IO between search result and structured data by using distinct storage devices.

If you plan to use a shared directories (NFS), we recommend that you review the configuration and check the setting read-ahead (requests blocks). The default value differs for each OS. In our internal test on Red Hat Enterprise Linux (RHEL) 7.9, setting the NFS read-ahead to 64 KB provided optimal search performance. We recommend exercising caution when tuning the above setting. Changing this setting might result in performance degradation in some cases and, therefore, rigorous testing is required before arriving at an appropriate value.

On Centos/Redhat, it is possible to check and to update the read-ahead setting with these commands. \$MNT is a path to the mounting point:

```
# uname -r  
4.18.0-193.6.3.el8_2.x86_64  
  
# cat /sys/class/bdi/$(mountpoint -d $MNT)/read_ahead_kb  
15360  
  
# echo 64 > /sys/class/bdi/$(mountpoint -d $MNT)/read_ahead_kb  
# cat /sys/class/bdi/$(mountpoint -d $MNT)/read_ahead_kb  
64
```



## Chapter 11

# Appendix

### 11.1 Supported languages

The system supports the following languages:

ISO 639 language code	Language name	ISO 639 language code	Language name
AR	Arabic	ID	Indonesian
CA	Catalan	IT	Italian
DA	Danish	LT	Lithuanian
DE	German	NE	Nepali
EL	Greek	NL	Dutch
EN	English	NO	Norwegian
ES	Spanish	PT	Portuguese
EU	Basque	RO	Romanian
FI	Finnish	RU	Russian
FR	French	SR	Serbian
GA	Irish	SV	Swedish
HI	Hindi	TA	Tamil
HU	Hungarian	TR	Turkish
HY	Armenian	ZH	Chinese
JA	Japanese	KO	Korean

## 11.2 Supported file types for unstructured content text extraction

For text extraction, external tool OpenText File Content Extraction is used. File Content Extraction supports a large number of file types. For a complete list, see the *Supported Formats* appendix in File Content Extraction documentation.

The following is a list of file types supported in OpenText Information Archive:

Format name	Description	Media (MIME) type	Extensions	Metadata	Requires container support
ASCII_Text_Fmt	Plain Text file	text/plain	TXT		
Excel_2000_Fmt	Microsoft Excel 2000	application/x-ms-excel	XLS	✓	
Excel_95_Fmt	Microsoft Excel 95	application/x-ms-excel	XLS	✓	
Excel_97_Fmt	Microsoft Excel 97	application/x-ms-excel	XLS, XLR	✓	
Excel_Fmt	Microsoft Excel (up to version 5)	application/x-ms-excel	XLS	✓	
ICS_Fmt	Microsoft Outlook iCalendar file format	text/calendar	ICS, VCS	✓	✓
MIME_Fmt	MIME (EML / MBX email)	message/rfc822	EML, MBX	✓	✓
MS_Excel_2007_Fmt	Microsoft Excel 2007 XML	application/x-ms-excel07	XLSX, XLTX	✓	
MS_Excel_HTML_Fmt	Microsoft Excel HTML format		XLS, HTM	✓	
MS_Excel_Macro_2007_Fmt	Microsoft Excel Macro 2007 XML	application/x-ms-excel07m	XLSM, XLTM, XLAM	✓	
MS_Excel_XML_Fmt	Microsoft Excel 2003 XML	text/xml	XML	✓	
MS_Outlook_Fmt	Microsoft Outlook message	application/vnd.ms-outlook	MSG, OFT	✓	✓

<b>Format name</b>	<b>Description</b>	<b>Media (MIME) type</b>	<b>Extensions</b>	<b>Metadata</b>	<b>Requires container support</b>
MS_Outlook_PST_Fmt	Microsoft Outlook Personal Folders File (.pst)	application/vnd.ms-outlook-pst	PST	✓	✓
MS_PPT_2007_Fmt	Microsoft PowerPoint 2007 XML	application/x-ms-powerpoint07	PPTX, POTX, PPSX	✓	
MS_PPT_Macro_2007_Fmt	Microsoft PPT Macro 2007 XML	application/x-ms-powerpoint07m	PPTM, POTM, PPSM, PPAM	✓	
MS_Project_2000_Fmt	Microsoft Project 2000	application/vnd.ms-project	MPP	✓	
MS_Project_2007_Fmt	Microsoft Project 2007	application/vnd.ms-project	MPP	✓	
MS_Publisher_98_Fmt	Microsoft Publisher from version 98	application/x-mspublisher	PUB	✓	
MS_Visio_2013_Fmt	Microsoft Visio 2013	application/vnd.visio	VSDX, VSTX, VSSX	✓	
MS_Visio_2013_Macro_Fmt	Microsoft Visio 2013 macro	application/vnd.visio	VSDM, VSTM, VSSM	✓	
MS_Visio_Fmt	Microsoft Visio (up to version 11)	image/x-vsd	VSD	✓	
MS_Visio_XML_Fmt	Microsoft Visio 2003 XML	text/xml	VDX	✓	
MS_Word_2000_Fmt	Microsoft Word 2000	application/msword	DOC	✓	
MS_Word_2007_Flat_XML_Fmt	Microsoft Word 2007 XML - Flat xml	text/xml	XML	✓	
MS_Word_2007_Fmt	Microsoft Word 2007 XML - Docx	application/x-ms-word07	DOCX, DOTX	✓	

<b>Format name</b>	<b>Description</b>	<b>Media (MIME) type</b>	<b>Extensions</b>	<b>Metadata</b>	<b>Requires container support</b>
MS_Word_95_Fmt	Microsoft Word 95 (Word OLE format version 6,7)	application/msword	DOC	✓	
MS_Word_97_Fmt	Microsoft Word 97	application/msword	DOC, WPS, WBK	✓	
MS_Word_HTML_Fmt	Microsoft Word HTML format		DOC, HTM	✓	
MS_Word_Mac_4_Fmt	Microsoft Word for Macintosh (version 4,5)	application/msword	DOC	✓	
MS_Word_Macro_2007_Fmt	Microsoft Word Macro 2007 XML	application/x-ms-word07m	DOCM, DOTM	✓	
MS_Word_PC_Fmt	Microsoft Word for PC (up to version 6)	application/x-ms-wordpc	MW		
MS_Word_Win_Fmt	Microsoft Word for Windows (up to version 5)	application/msword	DOC, WPS		
MS_Word_XML_Fmt	Microsoft Word 2003 XML	text/xml	XML	✓	
MS_Works_DOS_WPFmt	Microsoft Works Word Processor for DOS	application/x-msworks	WPS		
MS_Works_Win_SS_Fmt	Microsoft Works Spreadsheet for Windows	application/x-msworks	WKS, S30, S40		
MS_Works_Win_WPFmt	Microsoft Works Word Processor for Windows (up to 2000)	application/x-msworks	WPS, W40		

<b>Format name</b>	<b>Description</b>	<b>Media (MIME) type</b>	<b>Extensions</b>	<b>Metadata</b>	<b>Requires container support</b>
MS_XPS_Fmt	Microsoft Open XML Paper Specification (XPS/OXPS)	application/vnd.ms-xpsdocument	XPS, OXPS		
OneNoteFmt	Microsoft OneNote Note Format	application/onenote	ONE		
PDF_Fmt	Adobe PDF (Portable Document Format)	application/pdf	PDF	✓	
PKZIP_Fmt	ZIP Archive	application/zip	ZIP, ZIPX		✓
Portfolio_PDF_FFmt	Portfolio PDF File	application/pdf	PDF	✓	
PowerPoint_2000_Fmt	Microsoft PowerPoint 2000	application/x-ms-powerpoint	PPT	✓	
PowerPoint_95_Fmt	Microsoft PowerPoint 95	application/x-ms-powerpoint	PPT	✓	
PowerPoint_97_Fmt	Microsoft PowerPoint 97	application/x-ms-powerpoint	PPT	✓	
PowerPoint_Win_Fmt	Microsoft PowerPoint PC (up to version 4)	application/x-ms-powerpoint	PPT	✓	
Unicode_Fmt	Unicode text file	text/plain	UNI		
VCF_Fmt	Microsoft Outlook vCard file format	text/vcard	VCF	✓	
XHTML_Fmt	XHTML	text/xhtml	XML, XHTML, XHT		
XML_Fmt	XML	text/xml	XML	✓	
Z7Z_Fmt	7-Zip archive (7z)	application/7z	7Z		✓

The Format name column contains the KeyView format name, which uniquely identifies every file format. For all file types, the Format name is added to the metadata.

Text extraction of a file does not depend on its extension.

Container files are optional. Container files are file types that can contain sub-files (for example, ZIP files and Outlook mail messages). File types such as PDFs and Microsoft Word documents may also contain sub-files. If container files are supported, the content of the sub-files can also be searched. Some file types like Outlook mail messages and ZIP files can only be searched when container files are supported.

All text extracted from the container file and its sub-files is stored as one single value. It is not possible to search the text of an attachment only.

Optical character recognition (OCR) is not supported.

## 11.3 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 supported:

Java Types Supported by the IA Server	Schema Types Supported
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 XSD types and OpenText Information Archive types:

Schema Type	IA Type	Description
ENTITIES	STRING	None
ENTITY	STRING	None
ID	STRING	A string that represents the ID attribute in XML (only used with schema attributes)
IDREF	STRING	A string that represents the IDREF attribute in XML (only used with schema attributes)

<b>Schema Type</b>	<b>IA Type</b>	<b>Description</b>
IDREFS	STRING	None
language	STRING	A string that contains a valid language id
Name	STRING	A string that contains a valid XML name
NCName	STRING	None
NMTOKEN	STRING	A string that represents the NMTOKEN attribute in XML (only used with schema attributes)
NMTOKENS	STRING	None
normalizedString	STRING	A string that does not contain line feeds, carriage returns, or tabs
QName	STRING	None
string	STRING	A string
token	STRING	A string that does not contain line feeds, carriage returns, tabs, leading or trailing spaces, or multiple spaces
date	DATE	Defines a date value
dateTime	DATETIME	Defines a date and time value
duration	STRING	Defines a time interval
gDay	INTEGER	Defines a part of a date - the day (DD)
gMonth	INTEGER	Defines a part of a date - the month (MM)
gMonthDay	STRING	Defines a part of a date - the month and day (MM-DD)
gYear	INTEGER	Defines a part of a date - the year (YYYY)
gYearMonth	STRING	Defines a part of a date - the year and month (YYYY-MM)
time	STRING	Defines a time value
byte	INTEGER	A signed 8-bit integer
decimal	BIGDECIMAL	A decimal value
int	INTEGER	A signed 32-bit integer
integer	BIGINTEGER	An integer value

<b>Schema Type</b>	<b>IA Type</b>	<b>Description</b>
long	LONG	A signed 64-bit integer
negativeInteger	BIGINTEGER	An integer containing only negative values (for example, ..., -2, -1)
nonNegativeInteger	BIGINTEGER	An integer containing only non-negative values (for example, 0, 1, 2, ...)
nonPositiveInteger	BIGINTEGER	An integer containing only non-positive values (for example, ..., -2, -1, 0)
positiveInteger	BIGINTEGER	An integer containing only positive values (for example, 1, 2, ...)
short	INTEGER	A signed 16-bit integer
unsignedLong	LONG	An unsigned 64-bit integer
unsignedInt	INTEGER	An unsigned 32-bit integer
unsignedShort	INTEGER	An unsigned 16-bit integer
unsignedByte	INTEGER	An unsigned 8-bit integer
anyURI	STRING	Used to specify a URI
base64Binary	STRING	Used to express binary-formatted data
boolean	STRING	Used to specify a true or false value
double	DOUBLE	A double-precision 64-bit IEEE 754 floating point.
float	FLOAT	A single-precision 32-bit IEEE 754 floating point.
hexBinary	STRING	Binary data types are used to express binary-formatted data
NOTATION	STRING	None
QName	STRING	None

## 11.4 Mapping table SQL data types to PostgreSQL data types

OpenText Information Archive SQL data types for table columns are defined in the metadata. As table structured data is stored in PostgreSQL, these SQL data types are mapped to PostgreSQL data types.

Values of data type VARCHAR, VARCHAR and NVARCHAR will be stored as text in PostgreSQL.

Values of data type DATETIME may have a time zone or not. As there is no corresponding data type in PostgreSQL, these values are stored as timestamp with timezone in PostgreSQL.

The following table contains the mapping from SQL data types to PostgreSQL data types:

OpenText Information Archive SQL Data Types	PostgreSQL Data Types
BOOLEAN	boolean
BIT	bit
CHAR	text
VARCHAR	text
VARCHAR2	text
NVARCHAR	text
TINYINT	smallint
SMALLINT	smallint
INT	integer
INT4	integer
INTEGER	integer
BIGINT	bigint
FLOAT	float
REAL	real
DOUBLE	double precision
NUMERIC	numeric
DECIMAL	decimal
TIMESTAMP	timestamp
DATETIME	Timestamp with timezone
DATE	date

OpenText Information Archive SQL Data Types	PostgreSQL Data Types
TIME	time

## 11.5 AIP

The information for this section has been split into two tables:

- The **first table** contains the following events: Receive, Ingest, Commit, Reject, and Invalid.
- The **second table** contains the following events: Rollback, Rebuild, Confirmation, and Confirmation\_Completed.

Supplemental Value	Type	Receive	Ingest	Commit	Reject	Invalid
name	STRING	X	X	X	X	X
externalId	STRING	X	X	X	X	X
dssId	STRING	X	X	X	X	X
dssHoldingName	STRING	X	X	X	X	X
dssPdiSchema	STRING	X	X	X	X	X
dssProductionDate	DATETIME	X	X	X	X	X
dssBaseRetentionDate	DATETIME	X	X	X	X	X
dssProducer	STRING	X	X	X	X	X
dssEntity	STRING	X	X	X	X	X
dssPriority	INTEGER	X	X	X	X	X
dssApplication	STRING	X	X	X	X	X
dssRetentionClass	STRING	X	X	X	X	X
sipProductionDate	DATETIME	X	X	X	X	X
sipSeqno	LONG	X	X	X	X	X
sipIsLast	STRING	X	X	X	X	X
sipAiuCount	LONG	X	X	X	X	X
returnCode	STRING	X	X	X	X	X
returnMsg	STRING	X	X	X	X	X
pdiFileSize	LONG	No	X	X	No	No
pdiValuesCharCount	LONG	No	X	X	No	No
ciCount	LONG	No	X	X	No	No
ciSize	LONG	No	X	X	No	No
originalCiSize	LONG	No	X	X	No	No

<b>Supplemental Value</b>	<b>Type</b>	<b>Receive</b>	<b>Ingest</b>	<b>Commit</b>	<b>Reject</b>	<b>Invalid</b>
ciFormat	STRING	No	X	X	No	No
stateCode	STRING	X	X	X	No	No
phaseCode	STRING	X	X	X	No	No
LibraryMode	STRING	No	X	X	No	No
pdiSchema	STRING	No	X	X	No	No
partOfAggregate	STRING	No	X	X	No	No
aggregateCiSeqno	LONG	No	X	X	No	No
aggregateAiuSeqno	LONG	No	X	X	No	No
pdiConfigName	STRING	No	X	X	No	No
priority	INTEGER	X	X	X	No	No
receiveStartDate	DATETIME	X	X	X	No	No
receiverNodeName	STRING	X	X	X	No	No
sipFileFormat	STRING	X	X	X	No	No
sipFileSize	STRING	X	X	X	No	No
sipFileHash	STRING	X	X	X	No	No
ingestWaitStartDate	DATETIME	No	X	X	No	No
ingestStartDate	DATETIME	No	X	X	No	No
ingestNodeName	STRING	No	X	X	No	No
ingestConfigName	STRING	No	X	X	No	No
commitWaitStartDate	DATETIME	No	X	X	No	No
commitSync	STRING	No	X	X	No	No
commitDate	DATETIME	No	X	X	No	No
configurationHash	STRING	No	X	X	No	No
libraryId	STRING	No	X	X	No	No
storageRetentionDate	DATETIME	No	No	X	No	No
rejectDate	DATETIME	No	No	No	X	No
rejectCategory	STRING	No	No	No	X	No
rejectDescription	STRING	No	No	No	X	No
rejectUserName	STRING	No	No	No	X	No
invalidDate	DATETIME	No	No	No	No	X
invalidCategory	STRING	No	No	No	No	X
invalidDescription	STRING	No	No	No	No	X
invalidUserName	STRING	No	No	No	No	X

<b>Supplemental Value</b>	<b>Type</b>	<b>Receive</b>	<b>Ingest</b>	<b>Commit</b>	<b>Reject</b>	<b>Invalid</b>
confirmationName	STRING	No	No	No	No	No
confirmationId	STRING	No	No	No	No	No
confirmationType	STRING	No	No	No	No	No
confirmationReceive-Date	DATETIME	No	No	No	No	No
confirmationAvailable-Date	DATETIME	No	No	No	No	No
confirmationStorage-Date	DATETIME	No	No	No	No	No
confirmationPurge-Date	DATETIME	No	No	No	No	No
confirmationReject-Date	DATETIME	No	No	No	No	No
confirmationInvalid-Date	DATETIME	No	No	No	No	No
cid	STRING	No	No	No	No	No
externalId	STRING	No	No	No	No	No
seqno	LONG	No	No	No	No	No
offset	LONG	No	No	No	No	No
length	LONG	No	No	No	No	No
extra	STRING	No	No	No	No	No
fileName	STRING	No	No	No	No	No
fileSize	LONG	No	No	No	No	No
contentType	STRING	No	No	No	No	No
mode	STRING	No	No	No	No	No
contentId	STRING	No	No	No	No	No
contentFormat	STRING	No	No	No	No	No
contentMimeType	STRING	No	No	No	No	No
contentModifier	STRING	No	No	No	No	No
ContentSize	LONG	No	No	No	No	No
contentLast	LONG	No	No	No	No	No
contentStoreId	STRING	No	No	No	No	No
contentStoreName	STRING	No	No	No	No	No

<b>Supplemental Value</b>	<b>Type</b>	<b>Rollback</b>	<b>Rebuild</b>	<b>Confirm.</b>	<b>Confirm. Complete</b>
name	STRING	X	X	X	X

Supplemental Value	Type	Rollback	Rebuild	Confirm.	Confirm. Complete
externalId	STRING	X	X	X	X
dssId	STRING	X	X	X	X
dssHoldingName	STRING	X	X	X	X
dssPdiSchema	STRING	X	X	X	X
dssProductionDate	DATETIME	X	X	X	X
dssBaseRetentionDate	DATETIME	X	X	X	X
dssProducer	STRING	X	X	X	X
dssEntity	STRING	X	X	X	X
dssPriority	INTEGER	X	X	X	X
dssApplication	STRING	X	X	X	X
dssRetentionClass	STRING	X	X	X	X
sipProductionDate	DATETIME	X	X	X	X
sipSeqno	LONG	X	X	X	X
sipIsLast	STRING	X	X	X	X
sipAiuCount	LONG	X	X	X	X
returnCode	STRING	X	X	X	X
returnMsg	STRING	X	X	X	X
pdiFileSize	LONG	No	No	No	No
pdiValuesCharCount	LONG	No	No	No	No
ciCount	LONG	No	No	No	No
ciSize	LONG	No	No	No	No
originalCiSize	LONG	No	No	No	No
ciFormat	STRING	No	No	No	No
stateCode	STRING	No	No	No	No
phaseCode	STRING	No	No	No	No
mode	STRING	No	No	No	No
pdiSchema	STRING	No	No	No	No
partOfAggregate	STRING	No	No	No	No
aggregateCiSeqno	LONG	No	No	No	No
aggregateAiuSeqno	LONG	No	No	No	No
pdiConfigName	STRING	No	No	No	No
priority	INTEGER	No	No	No	No

Supplemental Value	Type	Rollback	Rebuild	Confirm.	Confirm. Complete
receiveStartDate	DATETIME	No	No	No	No
receiverNodeName	STRING	No	No	No	No
sipFileFormat	STRING	No	No	No	No
sipFileSize	STRING	No	No	No	No
sipFileHash	STRING	No	No	No	No
ingestWaitStartDate	DATETIME	No	No	No	No
ingestStartDate	DATETIME	No	No	No	No
ingestNodeName	STRING	No	No	No	No
ingestConfigName	STRING	No	No	No	No
commitWaitStartDate	DATETIME	No	No	No	No
commitSync	STRING	No	No	No	No
commitDate	DATETIME	No	No	No	No
configurationHash	STRING	No	No	No	No
libraryId	STRING	No	No	No	No
storageRetentionDate	DATETIME	No	No	No	No
rejectDate	DATETIME	No	No	No	No
rejectCategory	STRING	No	No	No	No
rejectDescription	STRING	No	No	No	No
rejectUserName	STRING	No	No	No	No
invalidDate	DATETIME	No	No	No	No
invalidCategory	STRING	No	No	No	No
invalidDescription	STRING	No	No	No	No
invalidUserName	STRING	No	No	No	No
confirmationName	STRING	No	No	X	No
confirmationId	STRING	No	No	X	No
confirmationType	STRING	No	No	X	X
confirmationReceive-Date	DATETIME	No	No	No	X
confirmationCommit-Date	DATETIME	No	No	No	X
confirmationStorage-Date	DATETIME	No	No	No	X
confirmationPurge-Date	DATETIME	No	No	No	X

Supplemental Value	Type	Rollback	Rebuild	Confirm.	Confirm. Complete
confirmationReject-Date	DATETIME	No	No	No	X
confirmationInvalid-Date	DATETIME	No	No	No	X
cid	STRING	No	No	No	No
externalId	STRING	No	No	No	No
seqno	LONG	No	No	No	No
offset	LONG	No	No	No	No
length	LONG	No	No	No	No
extra	STRING	No	No	No	No
fileName	STRING	No	No	No	No
fileSize	LONG	No	No	No	No
mimeType	STRING	No	No	No	No
mode	STRING	No	No	No	No
contentId	STRING	No	No	No	No
contentFormat	STRING	No	No	No	No
contentMimeType	STRING	No	No	No	No
contentModifier	STRING	No	No	No	No
ContentSize	LONG	No	No	No	No
contentLast	LONG	No	No	No	No
contentstoreId	STRING	No	No	No	No
contentStoreName	STRING	No	No	No	No

## 11.6 Supplemental data per event type and event name

The following event types are documented in this section:

- AIP
- AIC
- Job instance

### 11.6.1 Partial dispose AIP event type

Key	Example	Description
disposedRecordCount	7	Indicates the number of AIUs disposed.
disposedCiSize	4500	Indicates the byte-size of the disposed CI container. Configured by CI Purge Policy defined on the holding in the Retention/Disposition Policy disposedCiSize (will be 0 if set to Keep or Blank).
disposedPdiCharCount	78	Indicates the uncompressed byte-size of the disposed licensed structure data.

### 11.6.2 Dispose

Key	Example	Description
disposedRecordCount	7	Indicates the number of AIUs disposed (this value is the same as sipAiuCount).
disposedCiSize	4500	Indicates the byte-size of the disposed CI container.
disposedPdiCharCount	78	Indicates the uncompressed byte-size of the disposed licensed structure data.

### 11.6.3 AIC

Supplemental Value	Type	Search	Fetch
searchId	STRING	X	X
searchName	STRING	X	X
queryId	STRING	X	X
queryName	STRING	X	X
resultSchema	STRING	X	X
criteria	STRING	X	No
searchCompositionId	STRING	X	No
executionTime	LONG	X	No
recordCount	LONG	X	No
recordCacheCount	LONG	X	No
async	STRING	X	No
executionTime::sdx	LONG	X	No
executionTime::query	LONG	X	No

Supplemental Value	Type	Search	Fetch
executionTime::cache	LONG	X	No
executionTime::page	LONG	X	No
aipCount	LONG	X	No
aiuCount	LONG	X	No
libCount	LONG	X	No
rowFetched	LONG	No	X
row%d	STRING	No	X

## 11.6.4 Job instance

Value	Type	Create	Update	Delete	Run	Restart	Start	End
properties	String	X		X	X	X	X	X
status	String	No	No	No	No	No	No	X
startTime	Date Time	No	No	No	No	No	No	X
endTime	Date Time	No	No	X	No	No	No	X
updated_Fields	String	No	X	No	No	No	No	No
scheduledBy	String	No	No	No	X	No	No	No
scheduledAt	Date Time	No	No	No	X	No	No	No



# Chapter 12

## Glossary and acronyms

The following table contains short definitions for OpenText Information Archive resources, processes, and mechanisms:

Name	Acronyms and Related Terms	Description
Active Archiving	None	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 OpenText Information Archive.
Application	None	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	None	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 information inside OpenText Information Archive. It may contain from zero to multiple structured data elements, which are represented as XML, and/or zero to multiple unstructured data elements. AIP is a shorter name for a package.
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	None	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.  Audits can be associated to the system, a tenant, or a particular application,
Audit Event Type	None	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 Ingestion	None	Ingestion performed in the background. After the ingestion command call, the server creates a background request and returns immediately a response without having to wait for the end of ingestion. See also <a href="#">Direct Ingestion on page 453</a> and <a href="#">Batch Ingestion on page 450</a> .
Background Search	None	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	None	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	None	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 on page 453</a> .

---

Name	Acronyms and Related Terms	Description
Batch Processing	None	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.
Bucket	None	A storage configuration resource used within Dell EMC Elastic Cloud Storage, Microsoft Azure, Archive Center, and Amazon S3 Storage, and PowerScale OneFS S3 storage systems.
Cache-In	None	An action on an AIP object to restore the object in a 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 Lucene.
Cache-Out	None	An action on an AIP object to detach the object from a 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 Lucene.
Chain of Custody	None	A set of tests that check the integrity of the ingested tables in OpenText Information Archive.
Content Identifier	CID	An internal identifier for content information that can be associated to one or more records.
Confirmation	None	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	None	Internally, unstructured content that is related to an OpenText Information Archive configuration object is represented as a Content Item.

Name	Acronyms and Related Terms	Description
Crypto Object	None	The general encryption configuration object that contains settings for encryption/decryption, such as the security provider name, encryption algorithm, padding, key size, etc. All crypto objects can be seen in the IA Web App on the <b>Administration &gt; Encryption</b> page.
Custom Search	None	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	None	Archive information resource that presents a database for table-based applications.
Database Crypto	None	A configuration object specific to table-based applications only. The object contains a reference to crypto object with settings for a security provider name, encryption algorithm, padding, key size etc. The object is used for configuring encryption/decryption of structured and unstructured data for table-based applications.
Declarative Configuration	DC	The way to describe a set of configuration objects in YAML format. Declarative configuration is used to define entire system information, an application, a holding, or a search.
Default Crypto Object	None	OpenText Information Archive uses this object to encrypt and decrypt sensitive data, such as passwords for RDB databases, nodes and credentials for remote storage systems, in the System Data and Search Results databases. It is prohibited to modify or delete the default crypto object because, once the object is in-use and password encrypted with its settings, after the change of object properties, the existing encrypted passwords with the old settings can no longer be decrypted with the new properties' values. OpenText Information Archive blocks change of properties of default crypto object, as it is always "in-use".
Delivery Channel	None	The configuration resource that defines a destination where to send search results.

---

Name	Acronyms and Related Terms	Description
Direct Ingestion	None	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 on page 450</a> .
Disposition	None	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: <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.</li> </ol> Depending on the resource being disposed, additional steps may be required (for example, disposing AIPs require that the Confirmation job is run).
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	None	An xProc pipeline that is specifically intended to export OpenText Information Archive search results.
File System Folder	None	The storage configuration object that represents a file system folder in which unstructured content is stored.
File System Root	None	Storage configuration object that represents the storage of a Local File System or a PowerScale type and indicates a root location on a disk.
Group	None	A group defines a set of users. Groups are used to restrict access to applications and searches.
Hold	None	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	None	A logical configuration destination archive in which to ingest and store data that shares common characteristics used for SIP applications. 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.
Holding Composition	None	A configuration object used by the Holding Wizard to define a new holding.
Holding Crypto	None	A holding configuration object specific to SIP-based applications only. The object contains references to crypto objects with settings for a security provider name, encryption algorithm, padding, key size, etc. The holding object is used for configuring the encryption for structured, unstructured data of the SIP packages, as well as the SIP descriptor.  See also <a href="#">crypto object on page 452</a> , <a href="#">database crypto on page 452</a> , <a href="#">PDI crypto on page 456</a> , and <a href="#">default crypto object on page 452</a> .
Hold Set	None	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 OpenText Information Archive. It is a tool that provides the set of commands for the Administrator to manage the product and its resources.
OpenText Information Archive	IA	OpenText Information Archive 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	None	Ingestion is the process that registers an AIP (for SIP-based applications) or table data (for table-based applications) into an archive so the data can become searchable.

---

Name	Acronyms and Related Terms	Description
Ingestion Mode	mode	Defined on a holding for SIP applications. The way business data is preserved. There are three types of ingestion modes supported by OpenText Information Archive: PRIVATE, POOLED and AGGREGATE.
Ingestion Node	None	An archive process configuration resource that defines the parameters for the ingestion and enumeration processes.
Invalidation	None	“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.
Job	None	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	None	Represents either a scheduled or running instance of a job definition.
Library	None	A storage configuration resource that represents a container stored within a database. Libraries can be taken offline (cached out).
Library Policy	None	An archive process configuration object that contains a set of properties that define when a library can be closed when ingestion in POOLED and AGGREGATE modes.
Lightweight Directory Access Protocol	LDAP	One of the authentication mechanisms supported by OpenText Information Archive.
Nested Search	None	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 OpenText Information Archive throughout the information’s lifecycle.
OpenText Directory Services	OTDS	One of the authentication mechanisms supported by OpenText Information Archive.

Name	Acronyms and Related Terms	Description
Order	None	A configuration object used by SIP applications to set permissions and retention duration when running 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).
Partitioning Keys	PKEY	Keys defined on a holding in a SIP application to improve search performance. A partition key is used in the first tier of the query process to limit the data returned when a search is run. Partition keys are created during ingestion and are stored in the AIP object.
PDI Crypto	None	A configuration object specific to SIP-based applications only. It holds cryptography related information, such as indexes, partitioning, etc., for ingestion processors in its XML content file. It fully replaces indexes and pkeys from main PDI configuration object.
PDI Schema	None	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	None	Indicates which groups have access to either an application, search set, or an AIP.
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	None	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 have a state that controls what actions can be taken. The Clean Up Purge Candidate Lists and Applications job can be used to remove canceled or disposed purge lists.

---

Name	Acronyms and Related Terms	Description
Query	None	A search configuration resource that defines options for building a query to retrieve records.
Query Quota	None	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.
RDB Database	None	A storage configuration resource that represents a database in Postgres. It contains a set of properties to access physical database.
RDB Data Node	None	A storage configuration resource that contains a set of properties to establish a connection with a physical Postgres data node (formerly referred to as a federation). A data node is a container for databases.
Reception	None	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	None	An archive process configuration resource that defines the parameters for the receiver process.
Record	None	A record is either a row in a table or an AIU in an AIP.
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 CI.
Rejection	None	Rejection is one of the AIPs lifecycle states that can happen if the data did not match the schema. “Reject” is an action on the AIP package.  Rejection and Invalidation are similar. The difference is at what state the AIP was in.
Result Configuration Helper	None	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.

Name	Acronyms and Related Terms	Description
Result Master	None	A search configuration resource that represents the result search page columns and tabs (in-line panels and detail panels). Result masters are part of a search set.
Retention	None	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 Class	None	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.
Retention Policy	None	A compliance configuration object that specifies the rules for how long to retain the data.
Retention Set	None	A logical container that references data under retention, including whether items in the set are aging together and the type of items in set (for example, application, package, table or record).
Role	None	A role is mapped to set of actions that can be done in OpenText Information Archive, such as run search, apply retention, or ingest content. Groups are mapped to roles. Roles are fixed by OpenText Information Archive. Some examples include End User, Retention Manager, Administrator, Developer, and E-Discovery Administrator.
Rule	None	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 declarative configuration.
Schema	None	An archive information resource that presents a schema for table-based applications. Schemas are associated with a database and contain one or more tables.

---

Name	Acronyms and Related Terms	Description
Search	None	<p>The primary method used to access data that has been ingested by OpenText Information Archive. The process of searching an archive to retrieve the set of AIUs that satisfy the search criteria.</p> <p>At the same time there is a search configuration object in OpenText Information Archive that is used to represent the search item for either a SIP or table application. A search contains one or more search compositions.</p>
Search Composition	Search Set	A search configuration resource that manages the search components (XForm, result master, search, permissions) and is used to run a search.
Search Quota	AIU quota per parent	Maximum number of AIUs per AIP parent. The value zero indicates an unlimited number.
Space	None	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 RDB Library.
Space Root Folder	None	A storage configuration object that represents the relation between a Space and a Local File System/PowerScale storage systems.
Space Root Object	None	A storage configuration object that represents the relation between a Space and Dell EMC Elastic Cloud Storage/Amazon S3 storage systems.
Space Root RDB Library	None	A storage configuration object that represents the relation between a Space and an RDB database.
SQL Query	None	A search configuration resource that contains the query to be used during the search for table-based applications.
Storage	None	A storage configuration object that contains a list of properties for target storage configuration. A storage system holds data, such as unstructured content for records, library backups, raw XMLs, ingestion logs, etc.

Name	Acronyms and Related Terms	Description
Storage End Point Credentials	None	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: Dell EMC Elastic Cloud Storage, Amazon S3 Storage, and PowerScale OneFS S3.
Store	None	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	None	An archive information resource that presents a table for table-based applications. Tables are contained within schemas.
Tenant	None	A logical configuration object in an archive system that presents a customer business item for preserving and storing the data. Tenants store zero or more applications.
Transformation	None	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 OpenText Information Archive resources. Every resource has its own and unique UUID.
Value List	None	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.
xForm	None	The search resource that represents a user form that contains search criteria on the UI level.
xProc	W3C Recommendation	A W3C Recommendation that defines an XML transformation language to define XML Pipelines. For information on xproc see the W3C website