opentext™

OpenText™ Information Archive

A Quick Guide

This guide is meant to help new customers, or anyone with basic knowledge of OpenText Information Archive, to become quickly familiar with key functionality. The guide is also useful for professional services, for creating a quick proof of concept or demo for customers.

EARCORE250400-AQS-EN-01

OpenText™ Information Archive
A Quick Guide

EARCORE250400-AQS-EN-01
Rev.: 2025-Sept-08

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

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

Chapter 1

Overview

This guide is meant to help new customers, or anyone with basic knowledge of OpenText Information Archive, to become quickly familiar with key functionality. The guide is also useful for professional services, for creating a quick proof of concept or demo for customers.

This is a “how-to” guide, so rather than containing detailed information about a subject, it explains step-by-step actions for accomplishing required tasks. This guide references standard OpenText Information Archive documentation where you can find more detailed information about subjects.

The guide covers a list of scenarios, from storage object configuration through application creation, ingesting data, applying retention, search, and disposition. The scenarios can be learned and performed one-by-one, sequentially, or if you need to learn and understand a particular scenario, then you can start from anywhere in the guide. However, if you do not perform the scenarios sequentially, you should pay attention to pre-requisites that must be satisfied before performing a scenario.

In general, this guide assumes that you are using the version of OpenText Information Archive with encryption. Most scenarios are valid for the version without encryption, except for the steps related to encryption, which you can skip

The following text styles are used in this document:

- **Bold** – An object for the user to interact with, like web elements.
- *Italic* – A reference to a guide, link, path, or example.
- <Figure brace> – A name of an object specified by the user.

1.1 Understanding OpenText Information Archive basics

The OpenText Information Archive installation package contains two ZIP files. The main file (`infoarchive.zip`) includes the files and directories that are required to install and run all the OpenText Information Archive components and clients. The second file (`infoarchive-support.zip`) is optional and provides sample applications and documentation.

Understanding the directory structure helps you perform many tasks in OpenText Information Archive, like SSL/TLS configuration for components or sample application configuration and installation. Some tasks require you to edit configuration files or run scripts in the various directories. Understanding OpenText Information Archive components, their purpose and features is also important for successfully accomplishing archiving tasks during all the archive data lifecycle period. For more information, see section 1.1 “System components” in *OpenText Information Archive - Installation Guide (EARCORE-IGD)*.

It is also important to understand what an OpenText Information Archive application is. From the distribution point of view, an IA application is a set of resources that contain the configuration for OpenText Information Archive to properly operate archiving data and data lifecycle processes. An application can be imported into OpenText Information Archive.

For more information, see section 2.4 “OpenText Information Archive applications” in *OpenText Information Archive - Installation Guide (EARCORE-IGD)*.

1.2 Quickly installing OpenText Information Archive

You can quickly install a demo configuration of OpenText Information Archive, which is a basic installation on a standalone computer. This simple configuration is meant to help you test its features, set up a proof of concept, give a short demonstration, or set up a basic development environment. For more information, see section 3 “Quickly installing a demo configuration” in *OpenText Information Archive - Installation Guide (EARCORE-IGD)*.

1.3 Creating storage resources: data nodes, databases, and storage systems

This user scenario describes the process of creating data node, database, and storage system configuration resources. For more information, see section 2.1 “Setting up storage for structured data” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)* and .section 2.2 “Setting up storage for unstructured data” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

Data nodes and databases inside OpenText Information Archive are just configuration resources, which allow you to bind an application storage configuration with physical PostgreSQL data nodes and databases.

A storage system is an OpenText Information Archive configuration object that allows binding OpenText Information Archive with physical storage, such as file systems, PowerScale, ECS, S3, and so on.

After performing the steps in this scenario, you will have storage available for your table application.

Prerequisite: OpenText Information Archive is installed and started. For more information, see Quickly Installing OpenText Information Archive.

Steps:

1. Log in to IA Web App as a user with the Administrator role. An Administrator is allowed to manage storage configuration resources. You can use your own Administrator user, or you can use the `adam@iacustomer.com` user from the out-of-the-box example user accounts, such as the following:

- Username: `adam@iacustomer.com`
- Password: `password`

Caution

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

2. After a successful login, you are redirected to the **Administration** page, and you should see the Storage Systems.

   Tip: The IA Web App remembers your context. For example, if you access the **Encryption** tab, then access **Background Requests** and then click **Administration**, the UI automatically sends you to the tab you were last on.

3. Click the **Data Nodes** tab and register a new data node. Note that it is only possible to register a data node, it is not possible to create a data node.

   Registering a data node is only required for table applications. Creating an RDB Data Node or RDB database is not required for SIP Applications

   For more information, see section 2.1.3 “Registering a database in a data node” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

   a. On the **Data Nodes** tab, click the **+**button. The **Register Data Node** form opens.

   b. Fill the form with the following values:

      - Specify `<YOUR_DATA_NODE_NAME>` in the **Data Node Name** field.
      - Type the username in the **Username** field.
      - Type the superuser password in the **Password** text box. That password was set up during OpenText Information Archive installation.

        Alternatively, if you did not encrypt passwords, you can find the password value in the IA Server configuration file: `<IA_ROOT>/config/iaserver/application.yml`.

        Look at the `systemData.psql.databaseCluster.superuser.password` parameter. In this example the value was `test`.
      - Set the Connection URL value (in this example it is `jdbc:postgresql://localhost:5432`). This value indicates the JDBC connection string to the data node.

        Note that you cannot create two data nodes with the same connection URL. If you installed the first-time applications, you would not be able to create this new data node (in which case, skip registering the data node).

- The rest of the fields can remain with the default values and are meant for if you wish to set up SSL with an RDB data node.

c. Click **Register**.

d. Ensure that:

- The data node that you just created (`<YOUR_DATA_NODE_NAME>`) appears in the data nodes table.
- The **In Use** status is empty, which indicates that the data node does not contain any databases yet.

4. Create a new database. You will require a PostgreSQL user with the permission to create databases and create users.

   For more information, see section 2.1.2 “Creating/Editing a database” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

   a. On the **Data Nodes** tab, click the action button beside the applicable data node name and select **Create Database**. The **Create Database** form opens.

   b. Fill the form with the following values:

   - Set the `<YOUR_DATABASE_NAME>` value in the **Database Name** field.
   - Set the username in the **Username** field.
   - Set the password in the **Password** field.
   - Reenter the password in the **Confirm Password** field.

   Note: For the remainder of the other fields, you can leave the default values. They are required if you are using SSL for your data node. For more information, see section 5.3.1 “PostgreSQL data nodes, databases and setup” in *OpenText Information Archive - Installation Guide (EARCORE-IGD)*.

   c. Click **Create**.

   d. Ensure that the new database that you just created, `<YOUR_DATABASE_NAME>` appears in the table.

5. Create a new storage system. It is not possible to create a new space without a storage system created.

   For more details about creating storage systems and their types, see section 2.2.1 “Adding a storage system using IA Web App” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

   a. Navigate to **Administration > Storage**. The `defaultFileSystemRoot` storage may already present in the list if you installed the first-time applications.

   b. On the **Storage** tab, click the + button. The **Create Storage System** form opens.

c. Enter the following values:

   - Set the **Storage Type** value (in this example it is **Local File System**).
   - Set the `<YOUR_STORAGE_SYSTEM_NAME>` value in the **Storage Name** field.
   - Set the `<YOUR_STORAGE_SYSTEM_DESCRIPTION>` value in the **Description** field.
   - Set the `<YOUR_ FOLDER_PATH>` value in the Folder Path field (in this example it is `data/root`).

     For a real system, this should never configure the folder path to be inside the installation directory, especially as a relative path. If you have more than one IA Server, this has to be a shared path that all server's have access to.

d. Click **Create**.

e. Ensure that the new storage system, `<YOUR_STORAGE_SYSTEM_NAME>` appears in the table.

1.4 Creating a retention policy

This user scenario describes the process of creating a new retention policy. A retention policy is a compliance configuration object that specifies the rules for how long to retain data. For more information, see section 9.4 “What is a retention policy?” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.

At least one valid retention policy is required to create an application.

After performing the steps from this scenario, you can create applications and use the retention policy over the ingesting data.

Prerequisite: OpenText Information Archive is installed and started

1. Log in to IA Web App as a user with the Retention Manager role. A Retention Manager is allowed to manage compliance configuration resources. You can use your own Retention Manager user, or use the Rita user from the out-of-the-box example user accounts, such as the following:

   - Username: `rita@iacustomer.com`
   - Password: `password`

   After a successful login, you are redirected to the **Dashboard** page. The page will be empty until the Refresh Metrics job is run.

2. Navigate to the **Compliance** tab. The default sub-tab is Retention Policies. The list of policies is either empty or contains previously created retention policies that are available in OpenText Information Archive.

3. Create a new retention policy. For more information, see section 9.1.3 “Creating a retention policy” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

   a. Click the + button. The **Create Retention Policy** form opens.
   b. Enter the following:
      i. Specify `<YOUR_RETENTION_POLICY_NAME>` in the **Policy Name** field.
      ii. Select the **Aging Strategy** (for example, `Duration`).
      iii. Set the number of units for the aging strategy (for example, `89 days`).

4. Click **Create**.

5. Ensure that the new retention policy, `<YOUR_RETENTION_POLICY_NAME>`, is created and appears in the grid.

1.5 Creating a SIP application, space, and store

This scenario describes the process of creating a new SIP application, a space, and a store. At the end of this scenario, you will have everything you need to create a holding and start ingesting data. For more information, see section 4 “Applications and data” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)* and section 2 “Managing an OpenText Information Archive application” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

The following are prerequisites to complete the procedure:

- OpenText Information Archive is installed and started.
- Verify that at least one storage system exists. It can be the default storage system created when the first-time applications are installed or `<YOUR_STORAGE_SYSTEM_NAME>` created in the scenario Creating storage resources: data nodes, databases, and storage. To verify storage, do the following:
  - In IA Web App, navigate to **Administration > Storage**.
  - Verify that there is either a `defaultFileSystemRoot` or `<YOUR_STORAGE_SYSTEM_NAME>` storage in the Storage Systems table.
- The data node and database have been created. For more information, refer to Creating storage resources: data nodes, databases, and storage.
- At least one retention policy exists. For more information, refer to Creating a retention policy.

Steps

1. Log in to IA Web App as a user with the Developer role. A Developer is allowed to manage application configuration resources. You can use your own Developer user, or use the `connie` user from the out-of-the-box example user accounts, such as the following:

   - Username: `connie@iacustomer.com`
   - Password: `password`

   After a successful login, you are redirected to the **Applications** page.

2. Create a new application.

   An application is a logical configuration object in the archive system that presents customer business items for preserving and storing the data.

   For more information, see section 2.3 “Creating an application with the IA Web App” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

   a. Click the + button and select **Create Application** in the menu. The **Create Application** form opens.
   b. Enter the following:

      - Enter an **Application Name** for the application as `<YOUR_APPLICATION_NAME>` (for example, `PhoneCallsTestApp`).
      - Select the appropriate **Application Category** (in this example we are using `Customer Support`). If the list is empty, type the value.

        An **Application Category** is an informative field that can be used for filtering.
      - In the **Default Retention Policy** field, select `<YOUR_RETENTION_POLICY>`, which is a retention policy that either already exists or you created in the scenario Creating storage resources: data nodes, databases and storage.
      - Select the **Application Type** (in this example it is `Active Archiving`).

        An **Application Type** is an option that specifies a type of archive. It can be one of the following:

        – Application Decommissioning or

        – Active Archiving

          This field is not used for any business logic for OpenText Information Archive.
      - Select the **Archive Type** (in this example it is `Based on packages`).

        Possible values for the **Archive Type** field depend on the type of the application.
      - – Assuming that the Archive Type is set to **Based on packages** in the previous step, select the Cache Size option (in this example it is **All data is in cache**).

          You can set a limit for the **Cache Size** for SIP applications.

          For more information about cache size, see section 2.3 “Creating an application with the IA Web App” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)* and section 4.3.2 “Cache management” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*
   c. Click **Create**.

      - A pop-up message informs you that before data ingestion, you need to associate the new application with a dedicated space and store.

- The new application with the name `<YOUR_APPLICATION_NAME>` is visible on the **Applications** listing page. The status of the application is **In Test**.

3. Create a new space in IA Web App.

   A space is a storage configuration object that represents the relation between a storage system and an application. It is used by the space root folder, space root object and space root library. For more information, see section 2.8.1 “Creating a space in IA Web App” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

   a. Select the application `<YOUR_APPLICATION_NAME>` on the **Applications** listing page.
   b. Click on the application’s **Spaces** tab. The list of spaces in the grid is empty.
   c. Click the + button. The **Create Space** form opens. The Application value is pre-populated with the `<YOUR_APPLICATION_NAME>` value.
   d. Complete the following:

      - Set the **Space Name** to `<YOUR_SPACE_NAME>`. By default, the field is pre-populated with `<YOUR_APPLICATION_NAME>-space-0`.
      - In the **Storage Systems** field, select the Storage Type filter (in this example it is `File Storage`).

        The storage system is used for keeping unstructured data. For more information, see section 2.8 “Configuring application storage with spaces” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.
      - Select the appropriate value for the **Storage Name**. You can use the `<YOUR_ STORAGE_SYSTEM_NAME>` value created in the scenario Creating storage resources: data nodes, databases, and storage systems.
   e. Click the **Create** button.
   f. Ensure that the new space with the name `<YOUR_SPACE_NAME>` was created and is visible in the list of the spaces.

4. Create a new regular store.

   A store is a storage configuration object that contains properties for linking a space with the file system folder or bucket (a storage configuration resource used within ECS and S3 storage systems.). Stores hold records in the context of an application. For more information, see section 3.2.1 “Adding a store using the IA Web App” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

   a. Click on the application’s **Stores** tab. The list of stores in the grid is empty.
   b. Click the + button. The **Add Store** form opens.

      - The **Application** value is preselected with the `<YOUR_APPLICATION_NAME>` value.
      - The Space value is preselected with the `<YOUR_SPACE_NAME>` value.

c. Complete the following:

   - Set the **Store Name** as `<YOUR_STORE_NAME>` (for example, `PhoneCallsTestApp-store`).
   - Select the **Type** (in this example it is `Regular`).
   - Select the **Space Root** value. This is the storage system created in the space create step.
   - Click the Create File System Folder button and select the appropriate value. Set the **Name** of the new file system folder (in this example it is `PhoneCallsTestApp-file-system-folder-test`). Click the **Create** button.

5. Click the **Create** button.
6. Repeat steps 4 and 5, and select store type **Search Result** and store type **Library** to store both search results and structured data.
7. Ensure that you have three stores for the SIP application:
   - Regular for your unstructured data
   - Library for your structured data
   - Search Result for your search results

1.6 Creating a new holding with the holding wizard

This user scenario describes the process of creating a new holding with the holding wizard. For more information, see section 4.2.1 “Creating a holding with the holding wizard” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

The following are prerequisites to complete the procedure:

- OpenText Information Archive is installed and started.
- The application `<YOUR_APPLICATION_NAME>` is created and appears in IA Web App.

  The PhoneCalls example application is referenced as an example in this guide. To use the PhoneCalls example, you need to enable example user accounts.

  For more information about authentication in OpenText Information Archive, see section 4 “Security” in *OpenText Information Archive - Installation Guide (EARCORE-IGD)*.
- A valid PDI XSD schema for your business data is prepared on a local disk.

An example XSD schema for PhoneCalls can be found in the following path: `<IA_ROOT>/examples/applications/PhoneCalls/config/application-config/data-model-config/pdiSchema- urnEasSamplesEnXsdPhonecalls.1.0.xsd`. For more information about PDI schemas and the creation of holdings, refer to the following section and subsections: Section 4.2.1 “Creating a holding with the holding wizard” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

- Appropriate stores are created for `<YOUR_APPLICATION>`. Refer to the Creating a SIP application, space, and store scenario for how to create a new space and a new store.

Log in to IA Web App as a user with the Developer role. A Developer is allowed to manage application configuration resources. You can use your own Developer user, or use the `connie` user from the out-of-the-box example user accounts, such as the following:

- Username: `connie@iacustomer.com`
- Password: `password`

After a successful login you are redirected to the **Applications** page.

Navigate to <YOUR_APPLICATION_NAME>. The list of pre-installed searches will be posted. For more information, see Section 6 “Setting up searches for an application” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

Navigate to the **Holdings** tab.

Important

For more details about the holding configuration object and holding wizard, see Section 4.2.1 “Creating a holding with the holding wizard” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

Complete the following to create a holding with the holding wizard:

1. Click + and select **Create Holding**. The **Start** step is displayed. Complete the following:

   Holding Name
   Enter a name for the holding.

   Application Space
   `<YOUR SPACE_NAME>` is preselected in the **Application Space** select box.

   Schema File
   Select the appropriate **PDI Schema File** for `<YOUR_APPLICATION_NAME>` by uploading it with the **Upload new XSD file…** button (for example, you can use the PhoneCalls schema `urn:eas-samples:en:xsd:phonecalls.1.0`. Refer to the prerequisites for this scenario).

Archival Information Unit (AIU) Node
Select the AIU node and choose the appropriate node (for the PhoneCalls example, it is the `/Calls/Call` node).

An AIU is an information atom, the smallest archival unit of an information package. An AIP contains one or more AIUs. For more information about AIUs and holdings, see Section 4.2.1.1 “Step 1: Start” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

- When done, click **Next**.

2. Complete the following for the **Data and Contents** step:

a. For the **Data and Contents > Search Fields** step:

Select from Schema
Click the **Select from Schema** link and select the nodes corresponding to the fields you will use to create a search (for example, the nodes are the `CallStartDate` and `CallEndDate` nodes).

Searches are the primary method that users use to access data that has been ingested by OpenText Information Archive. For more information, see Section 6 “Setting up searches for an application” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

Partition Key Method
Select the **Partition Key Method** values for the selected fields (in this example it is `Min/Max` value). For more information, see Section 4.3.4.1 “Using partition keys” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

When done, click **Next**.

b. For the **Data and Contents > Encrypted Fields step**, leave the settings as default. For more information about encryption in OpenText Information Archive, refer to *OpenText Information Archive - Encryption Guide (EARCORE-AGE)*.

Click **Next**.

c. For the **Data and Contents > Unstructured Content** step:

Is there unstructured content associated with record?
If the checkbox is not selected, there is no unstructured content associated with records. For this example, select the checkbox.

Skip the step if the checkbox was not selected on the previous step.

Select from Schema
Click the **Select from Schema** link and select the node corresponding to the field in the schema that represents the names of unstructured content. For the PhoneCalls example, the node is `/Calls/Call/Attachments/Attachment/FileName`.

OpenText Information Archive can archive any type of data produced from any source application if the data is packaged into SIPs that meet the file structure and format requirements. Optionally, a SIP can contain any related content files (documents, images, PDFs *etc.*). For more information, see Section 6 “SIP archiving fundamentals” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.

When done, click **Next**.

d. The **Data and Contents > Advanced Settings** step allows you to configure the OpenText Output Transformation plug-in, 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* on support.opentext.com (https://login.otiam.opentext.com/).

   When done, click **Next**.

3. Complete the following for the **Ingestion** step:

   a. For the **Ingestion > Library Mode** step, leave the ingestion mode as **Private**.

      OpenText Information Archive supports three ingestion modes: private, pooled, and aggregated. Using the proper ingestion mode is a key for optimized search performance based on the nature of the ingestion frequency.

      For more information about SIP ingestion modes, refer to Section 4.3.1 “Library mode” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)* and Section 6.7.1 “Ingestion mode” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.

      Click **Next**.

   b. For the **Ingestion > Library Management** step, configure the library management, if needed. By default the Load Balancing Mode is set to Primary level and Primary Location is set to Create new library store. Library management allows to define different scenarios to distribute data between databases. For more information about library management, see Section 4.2.1.3 “Step 3: Ingestion” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

      When done, click **Next**.

   c. For the **Ingestion > Encryption** step, configure the scope and scheme for the key allotment, if needed. By default, the scope is set to application-level and no scheme is used. Key allotment does not enable a rotation of keys or the updating of an existing key. It uses a new key for subsequent data ingestion, without changing any of the keys for data that has already been ingested.

      When done, click **Next**.

   d. The **Ingestion > Metadata** step allows you to declare custom fields at the package level to define common information for all AIUs/ records.

      When done, click **Next**.

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

      When done, click **Next**.

4. For the **Stores** step, you need to select the appropriate Common Store values from the list of available stores (for example, leave the stores with default values). You may use `<YOUR_STORE_NAME>` store, which was created in the scenario Creating a SIP application, space, and store. A store is 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.

   When done, click **Next**.

5. Complete the following for the **Confirmation** step:

   a. For the **Confirmation** > **Confirmations** step, confirmations can be set up for entering various SIP lifecycle states. Depending on the state, additional information about the records may be available. By default, all events are un-selected. A confirmation is a message generated in reaction to an AIP event. Confirmation messages can be configured to be generated during a package's lifecycle transition. For our example, we will not configure any confirmations.

      Click **Next**.

   b. The **Confirmation > Advanced Settings** step allows you to edit SIP and AIU queries. 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. For more information, see Section 4.3.8.1 “Confirmation” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

      When done, click **Next**.

6. Complete the following for the **Retention** step:

   a. Complete the following for the **Retention > Retention Policy** step:

      Retention at ingestion

      Select the appropriate type of retention on ingest and the policy. By default, the `Package` type of retention is selected for ingestion and the application default retention policy (for example, leave the retention policy settings as is). You may use `<YOUR_RETENTION_POLICY_NAME>`, which was created in the scenario Creating a retention policy.

      A retention policy is a compliance configuration object that specifies the rules for how long to retain the data. For more information about retention policies, see Section 9 “Performing compliance tasks” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

When done, click **Next**.

b. The **Retention > Retention Store** step allows you to create a new Retention Data library store. For this scenario, we will use the default values.

   Click **Next**.

c. The **Retention > Disposition Policy** step allows you to configure how disposition impacts structured data and unstructured contents. For more information, see Section 4.2.1.6 “Step 6: Retention” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*. For this scenario, we will use the default values.

   Click **Next**.

d. The **Retention > Retention Display** step allows you to configure the search template to display records in retention sets and purge lists when the granular retention is applied during ingestion. For this scenario, we will use the default values because we are not using granular retention.

   Click **Next**.

7. For the **Finish** step, review the holding settings. The holding wizard validates some of the user values on each step, so all values should be correct. If you need to correct an error, return to the previous wizard step by using the **Back** button.

   - Click the **Finish** button
   - In the pop-up dialog, confirm the creation of the new holding by clicking the Finish button.

     The holding wizard window is closed. The new holding with the name `<YOUR_HOLDING_NAME>` is displayed in the holdings grid.

     In case of any unexpected errors, the holding wizard will display the corresponding pop-up message with the reason for an exception. The window of the wizard will not be closed, and you will have a chance to return to the corresponding step to fix the issue.

Check the new holding properties:

- All properties on the side panel correspond the settings specified on holding wizard steps – name, schema, ingestion mode.
- The holding is not **In Use**.

1.7 Preparing SIPs and ingesting SIPs with IA Shell

This user scenario describes the basic process of ingestion for a SIP based application. For more information, see Section 2.7.5 “ingest” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*.

By following the steps below, you will prepare appropriate data for a SIP package and then ingest the SIP with IA Shell tool into the holding prepared in.

The following are prerequisites:

- OpenText Information Archive is installed and started.
- The application `<YOUR_APPLICATION_NAME>` is created and can be seen in IA Web App.
- An appropriate holding is created for `<YOUR_APPLICATION_NAME>`. Refer to the Creating a new holding with the holding wizard scenario for how to create a new holding.
- The business data to be archived is prepared on a local disk.
  - You may already have the data in XML format prior to ingest. The data in XML format should match the XSD schema, which is used for holding creation in the scenario Creating a new holding with the holding wizard.
  - If, at this point, you only have a system ready to extract the data, follow the ETL process. You can use any convenient ETL tool for XML generation, such as Talend Open Studio, OpenText ContentBridge, or OpenText connectors for SAP, Documentum, or SharePoint, or even a custom script.

Steps:

1. Prepare the information and content for the SIP package.

   In this example we will use a SIP package from the PhoneCalls example application. If you are going to create a SIP package from scratch, you may also refer to the example SIP archive to verify your progress: `<IA_ROOT>/examples/applications/PhoneCalls/data/PhoneCallsSample-2010.zip`.

   a. Unzip the example SIP archive to `<YOUR_PATH>/SampleSIP_Test` folder. The content of the archive will be:

      - `eas_pdi.xml`
      - `eas_sip.xml`
      - `recording1.mp3`

   b. Create a PDI file (`eas_pdi.xml` file).

      - Following the ETL process, generate the XML file, based on your initial business data (see the prerequisites) containing the data to be preserved in OpenText Information Archive. The target XML file, generated based on your data, should match your xsd schema. You can use any convenient ETL tool for XML generation like Talend Open Studio, or OpenText connectors for SAP, Documentum, or SharePoint, or even a custom script.

- In this example verify the content of the `eas_pdi.xml` file: `<YOUR_PATH>/SampleSIP_Test/eas_pdi.xml`.

  The **Preservation Description Information (PDI)** file (`eas_pdi.xml`) contains structured data in AIU format for archiving. It is assumed that on this step you already have a schema file that specifies the structure of this file. There can be only one PDI file in a SIP. The PDI file is part of the OAIS standard. For more information, see Section 6.4 “PDI” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.

c. Create a **SIP descriptor** XML file (`eas_sip.xml`) or manifest (in OAIS).

- Create a metadata file (eas_sip.xml) describing the data that is going to be archived in compliance with the SIP schema. There can be only one SIP descriptor in a SIP package. For more information, see Section 6.3 “SIP descriptor” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)*.
- In this example, open the `eas_sip.xml` file for edit: `<YOUR_PATH>/SampleSIP_Test/eas_sip.xml`
- Pay attention to the **holding name** value – it should be equal to the <YOUR_HOLDING_NAME> (see the scenario Creating a new holding with the holding wizard).
- Pay attention to the number of AIUs (the **aiu_count** value) in the structure – it should be equal to the number of elements in the PDI file.

d. If your holding accepts unstructured content, prepare the corresponding content to be archived.

- Create the content files (i.e., documents, images, PDFs etc.) that you reference to in the PDI file.
- Place the files near the SIP descriptors and structured data.
- In this example, the content file is already there: `<YOUR_PATH>/SampleSIP_Test/recording1.mp3`.

e. Pack the files into a zip archive. The ZIP package should contain:

- `eas_pdi.xml` (the PDI file)
- `eas_sip.xml` (the SIP descriptor)
- The content files (files with unstructured content).
- In this example, select all files in the `<YOUR_PATH>/SampleSIP_Test` folder and send them to a ZIP archive.

f. Ensure that:

- The PDI file and the SIP descriptor should be in the root of the archive.

- Name and move the archive so it will be available by this path: `<YOUR_PATH>/SampleSIP_Test-1.zip`

2. Ingest SIP package to the newly created holding with IA Shell.

   a. Launch the IA Shell tool.

   Refer to *OpenText Information Archive - OpenText Information Archive Shell Guide (EARCORE-ARE)* to learn more about the tool, its commands and how to use it.

   - On Linux: `<IA_ROOT>/bin/iashell`
   - On Windows: `<IA_ROOT>/bin/iashell.bat`

   b. Connect as a user with the Developer role.

```
Type 'help' for a list of commands and options, type 'exit' to leave the shell.

iashell> connect --user connie@iacustomer.com —password password
Connected to "http://localhost:8765/services" as connie@iacustomer.com
```

   c. Ingest the SIP archive prepared in step 2 with the *ingest* command, as below:

```
iashell> ingest --d applications/<YOUR_APPLICATION> --from <YOUR_PATH>/
SampleSIP_Test-1.zip

SampleSIP_Test.zip 100% [==========] 141640/141640 (0:00:01 / 0:00:00)
Received SampleSIP_Test.zip
OK
Ingest took 2.171 seconds
```

3. Check in IA Web App that the SIP package was ingested properly.

   a. Log in to IA Web App as `connie` (Developer).

   b. Navigate to the **Applications >`<YOUR_APPLICATION_NAME>` > Packages** tab.

   c. Verify that:

   - The package with name `SampleSIP_Test-1` is displayed.
   - The package is in the Completed phase.
   - The package has <YOUR_HOLDING_NAME> holding name on the SIP tab of the side panel.
   - Go to the **Applications > `<YOUR_APPLICATION_NAME>` > Holdings** tab and verify that the holding `<YOUR_APPLICATION_NAME>` is now **In Use**.

1.8 Creating and running a search for a SIP application

This user scenario describes the process of the search operation for SIP based applications.

By following the steps below, you will create and use a search for data ingested in a SIP package. For more information, see Section 6 “Setting up searches for an application” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

The following are prerequisites:

- OpenText Information Archive is installed and started.
- The application `<YOUR_APPLICATION_NAME>` is created and can be seen in IA Web App.
- A SIP package has been ingested to `<YOUR_APPLICATION_NAME>`. Refer to the Preparing SIPs and ingesting SIPs with IA Shell scenario for details.

Steps:

1. Log in to IA Web App as a user with the Developer role. A Developer is allowed to manage application configuration resources. You can use your own Developer user, or use the `connie@iacustomer.com` user from the out-of-the-box example user accounts, such as the following:

   - Username: `connie@iacustomer.com`
   - Password: `password`

   After a successful login you will be redirected to the **Applications** page.

2. Navigate to **Applications > <YOUR_APPLICATION_NAME>**.

3. Create a new search. For more information, see Section 6 “Setting up searches for an application” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

   a. Click + (create) button and select **Create Search**. The **Create Search** window is opened.

   b. The current tab is **1. Start**. Complete the following fields:

      - Set the **Name** of the new search as `<YOUR_SEARCH_NAME>`.
      - Leave the Type of the search as the default, *Primary Search*.

        Leave the Search Form as the default, **Composed – use built-in form composer**.
      - The **Archival Collection** field is predefined with the `<YOUR_HOLDING_NAME>-aic` value. The AIC and query have been created in the scenario “Creating a new holding with the holding wizard” based on PDI schema and search parameters.

An archive information collection (AIC) is a search configuration resource that contains a set of criteria to be used during the search. It organizes a set of AIPs, which can support flexible and efficient data access. For more information, see Section 4.4 "AIC configuration details" in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

- The Configuration field is predefined with the `<YOUR_HOLDING_NAME>-query` value.

  A Query Configuration field refers to the query that is going to be used to search data.

c. Click **Next**.

d. The current tab is **Criteria**. Some fields in the grid can be preselected if step 3 has been performed in the scenario Creating a new holding with the holding wizard.

- Select the search fields in the **Show In Form** column. In this example, we will choose only one field, `CallEndDate`.

  It is recommended for a search that you at least have one field a search criteria. You can still decide if you want to allow returning all the records.

- Specify if some of the search fields should be required by selecting the corresponding field in the **Required?** column. In this example, leave all of the columns as is.

  The **Required?** field is only be editable if the Show in Form is selected for a row. It is recommended that, unless you want users to return all of the records, that you make at least one search criteria mandatory.

e. Click **Next**.

f. The current tab is **Results**.

- Select the fields that will be presented in the search result. In this example, select `CallStartDate`, `CallEndDate`, and `CustomerLastName`.

g. Click **Finish**.

- The search `<YOUR_SEARCH_NAME>` is created and the edit search set view is displayed.
- The search status is Draft.

h. Change the status of the search to **Ready**, so that other users can run the search. Normally you would test first before setting the search to be ready.

- Open the **Edit Search** form by clicking on the gear button on the right of the search name.
- Change the **Status** to Ready.
- Click **OK**.

i. Click **Save**.

4. Run the created search.

   a. Navigate to **Applications > <YOUR_APPLICATION_NAME>**.

      You should see the new search in the grid.

   b. Select the `<YOUR_SEARCH_NAME>` search.

      The search form with the corresponding search fields is opened.

   c. Set the search parameters.

      In this example it is a date range: from `2010-01-01` to `2011-01-01`.

      These values are taken from the corresponding PDI file of the ingested SIP (see step 2 in Preparing SIPs and ingesting SIPs with IA Shell.

   d. Click **Search**.

   e. Verify the result:

      The number of rows should be corresponding the number of content elements (AIUs) in the ingested AIP packages. In this example, there should be 10 elements in the search result.

      The result search data is presented in compliance with step 4 of this scenario.

5. Run the background search.

   A background search is a search that runs asynchronously and is associated with 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. For more information, see Section 6.9.2 “Searching for records that are cached out” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

   a. Perform steps 1 - 3 of this scenario.

   b. Open the context menu of the **Search** button and select **Run search in background**.

      The confirmation window opens and the **Request Name** field is predefined with the default value.

   c. Specify the **Request Name** with `<YOUR_BACKGROUND_REQUEST_NAME>` value.

   d. Click **Start Background Request**.

      The confirmation form is closed.

   e. Navigate to **Background Request** tab.

      The `<YOUR_BACKGROUND_REQUEST_NAME>` value is visible in the list of background requests.

      The status of the request is either `In queue` or `Completed`, depending on how fast you navigated to the tab after creating the background search.

   f. Wait until the request `<YOUR_BACKGROUND_REQUEST_NAME>` is finished. The **View** button should appear next to the request. Click the **View** button.

   g. Verify the result:

- The number of rows should correspond to the number of content elements (AIUs) in the ingested SIP. In this example, there should be 10 elements in the search result.
- The search result data is presented in 3 columns in compliance with step 4 of this scenario.

1.9 Applying a retention policy to an AIP package

This user scenario describes the process of applying a retention policy to an AIP package.

By following the steps below, you will optionally create and apply a retention policy to an AIP package. In this scenario it is proposed to create the new policy with appropriate properties, so it will be possible to use this retention policy for immediate disposition.

Note that normally you would not do this procedure and instead configure the holding to apply retention on ingestion for packages. This procedure is more for an example.

The following are prerequisites:

- OpenText Information Archive is installed and started.
- A SIP was ingested in the scenario Preparing SIPs and ingesting SIPs with IA Shell.

Steps:

1. Log in to IA Web App as a user with the Retention Manager role. A Retention Manager is allowed to manage compliance configuration resources. You can use your own Retention Manager user, or use the `rita@iacustomer.com` user from the out-of- the-box example user accounts, such as the following:
   - Username: `rita@iacustomer.com`
   - Password: `password`
2. Following the steps described in the section Creating a retention policy, create a new retention policy with the following properties:
   - **Policy Name**: `<DISPOSE_RETENTION_POLICY_NAME>`
   - **Aging Strategy**: Fixed Date
   - **Aging Strategy Date**: Specify some past date, for example a week before the current date.
3. Apply the new retention policy to an AIP. In this example it is `SampleSIP_Test-1`.
   a. Navigate to the **Applications** page and select the `<YOUR_APPLICATION_NAME>` application.

b. Navigate to **Packages** and find the appropriate package. In this example it is the `SampleSIP_Test-1` package.

c. Verify that the package is in the Completed phase.

d. Open the context menu for the record with the selected package and click **Apply retention**.

   The current tab is **Choose policy**.

e. Select the `<DISPOSE_RETENTION_POLICY_NAME>` policy.

f. Click **Next**.

   The current tab is **View Details**.

g. Verify the Retention Policy Name and the Fixed Date. They should the same as you specified on step 2.

h. Click **Next**.

   The current tab is **Specify Retention Set**.

i. Specify the **Retention Set Name** field with `<YOUR_RETENTION_SET_NAME>` value.

j. Click **Next**.

   The current tab is **Finish**.

k. Verify the **Retention Set** and **Policy Details** values.

l. Click the **Finish** button.

   The **Apply Retention Policy** screen is closed and the **Packages** tab is displayed.

m. Find the package (in this example it is `SampleSIP_Test-1`) and click the link in the **Name** column.

   The package form is opened with the details corresponding to it.

n. Verify that in the **Package Retention** section of the form, there is the `<DISPOSE_RETENTION_POLICY_NAME>` policy and the corresponding **Qualification Date** value equals to the **Aging Strategy Date** specified on step 2.

1.10 Creating and applying a hold on an AIP package

By following the steps below, you will create and apply a hold on an AIP package.

A hold is a compliance configuration object. A hold is applied to an object in order to block the delete operation or disposition either temporarily or indefinitely.

For more information about holds, see Section 9.5 “What is a hold?” in *OpenText Information Archive - Fundamentals Guide (EARCORE-ACS)* and Section 9.13 “Holds” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

The following are prerequisites:

- OpenText Information Archive is installed and started.

- A SIP was ingested in the scenario Preparing SIPs and ingesting SIPs with IA Shell.

Steps:

1. Log in to IA Web App as a user with the Retention Manager role. A Retention Manager is allowed to manage compliance configuration resources. You can use your own Retention Manager user, or use the `rita` user from the out-of- the-box example user accounts, such as the following:

   - Username: `rita@iacustomer.com`
   - Password: `password`

   After a successful login, you will be redirected to the **Dashboard** page.

2. Navigate to the **Compliance > Holds** tab.

3. Create a new hold.

   a. Click + (create) button.

   b. Fill in the create form with appropriate values:

      - **Hold Name**: `<YOUR_HOLD_NAME>`.
      - **Type**: Specify the type of the hold. In this example it is Legal.

        For more information, see the following section and its subsections: Section 9.13 “Holds” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

      - Other fields are not required, but you can specify your own Approved Date and Review Date. Otherwise, the system will set the current dates by default.

   c. Click the **Create** button.

      - The hold with `<YOUR_HOLD_NAME>` is created and visible in the Holds table.
      - The In Use value for the Hold is No.

4. Navigate to **Applications** and select the `<YOUR_APPLICATION_NAME>` application.

5. Navigate to **Packages** and find the appropriate package. In this example it is the `SampleSIP_Test-1` package.

6. Apply the new hold to the AIP package.

   a. Open the context menu for the record with the selected package and click the **Apply hold** menu item.

      The current tab is **Choose Hold**.

   b. On the tab, find and select the `<YOUR_HOLD_NAME>` hold.

   c. Click **Next**.

The current tab is **Specify Hold Set**.

d. Specify the **Hold Set Name** with `<YOUR_HOLD_SET_NAME>` value.

e. Click **Next**.

The current tab is **Finish**.

f. Verify the **Hold Set** and **Hold Details** values.

g. Click **Finish**.

The **Apply Hold** form is closed, and the **Packages** tab is displayed

h. Find the package (in this example it is the `SampleSIP_Test-1`) and click the link in the **Name** column.

The package form is opened with the details corresponding to it.

i. Verify that in the **Package Hold** section of the package form, there is the `<YOUR_HOLD_NAME>` hold with the corresponding type and name specified on step 3.

1.11 Disposing of an AIP package

A disposition is the controlled process of removing data from the archive after the required aging period has elapsed (defined by the retention polices applied). Only items that are under retention go through disposition process.

The disposition process has the following steps:

- Put information into purge lists.
- Get approval to dispose of a list.
- Run the disposition job.
- Run the clean job to clean up the space.

Depending on the resource being disposed of additional steps may be required (for example disposing of AIPs requires the confirmation job to be run).

Disposition can be prevented by applying holds to the items or its parent (for example, applying a hold to an AIP is effective for all of its records).

For more information about disposition, see the following section and its subsections: Section 9.10 “Disposition flow” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

The following are prerequisites:

- OpenText Information Archive is installed and started.
- A SIP was ingested in the scenario Preparing SIPs and ingesting SIPs with IA Shell.
- An appropriate `<YOUR_RETENTION_POLICY_NAME>` retention policy is applied on the target AIP package in the scenario Applying a retention policy to an AIP package.

- The Dispose Purge Candidate List job should be enabled for the `<YOUR_APPLICATION_NAME>` application.
  - Log in to IA Web App as an Administrator (see the scenario Creating storage resources: data nodes, databases and storage, step 1).
  - Navigate to Administration > Jobs.
  - Find and edit the Dispose Purge Candidate List job. Select the `<YOUR_APPLICATION_NAME>` application in the list of **Applied To** applications.
- The system application should be installed. Navigate to the `<IA_ROOT>/first-time-setup/applications/System` and run the install(.bat) file. For more information, see Section 3.3 "Installing applications for a demo configuration" in *OpenText Information Archive - Installation Guide (EARCORE-IGD)* and the following section and its subsections: Section 2.4 "OpenText Information Archive applications" in *OpenText Information Archive - Installation Guide (EARCORE-IGD)*.

By following the steps below, you will prepare an AIP package for disposition and then dispose of it.

Steps:

1. Log in to IA Web App as a user with the Retention Manager role. A Retention Manager is allowed to manage compliance configuration resources. You can use your own Retention Manager user, or use the `rita` user from the out-of- the-box example user accounts, such as the following:

   - Username: `rita@iacustomer.com`
   - Password: `password`

   After a successful login you will be redirected to the **Dashboard** page.

2. Select the `<YOUR_APPLICATION_NAME>` application and navigate to the **Packages** tab.

3. Verify that the package can be disposed of.

   a. Find the package (in this example it is `SampleSIP_Test-1`) and click the link in the **Name** column.

   b. Remove all records from the Package Hold table. After confirmation, the changes will be saved automatically.

   c. Remove any retention policies that have qualification dates greater than the current date.

      Otherwise, disposition will not be possible for the AIP package in Completed phase.

4. Log in to IA Web App as a user with the Developer role. A Developer is allowed to manage application configuration resources. You can use your own Developer user or use the `connie@iacustomer.com` user from the out-of-the-box example user accounts, such as the following:

- Username: `connie@iacustomer.com`
- Password: `password`

After a successful login you will be redirected to the **Applications** page.

5. Select the `<YOUR_APPLICATION_NAME>` application and navigate to the **Packages** tab.

6. Generate the purge list.

   a. Navigate to the **Administration > Jobs** page.

   b. Run the Generate Purge Candidate List job. Wait until the job is finished and has a status of SUCCESS. For more information, see *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

7. Approve the generated purge list.

   a. Log in to IA Web App as a Retention Manager role (see step 1).

   b. Navigate to **`<YOUR_APPLICATION_NAME>` > Purge Lists** page.

   c. In the Purge Lists table, find the new purge list with following properties:

      - **Associated Policy**: `<YOUR_RETENTION_POLICY_NAME>`.
      - `Status`: Under Review.
      - `Created Date`: Current date and time.

   d. Open the details of the selected purge list by click on the button on the left of the record (a triangle button).

   e. Verify that in the details the list of AIP packages is correct.

   f. Click **Approve**.

   g. Enter the approve purge list **reason**.

   h. Click **Approve**.

      The status of the purge list is changed to `Approved`.

8. Log in to IA Web App with the Administrator role (see step 4).

9. Run the Disposition job for the purge list.

   a. Navigate to the **Administration > Jobs** page.

   b. Run the Dispose Purge Candidate List job, making sure to set the application scope to your application. Wait until the job is finished and has a status of SUCCESS.

   c. Navigate to **`<YOUR_APPLICATION_NAME>` > Packages** page.

   d. Verify that the AIP package (in this example it is `SampleSIP_Test-1`) is disposed of and is not visible in the list of packages.

10. Run the Clean job to clean-up the space.

    a. Navigate to the **Administration > Jobs** page.

b. Run the Clean job. Wait until the job is finished and has a status of SUCCESS.

1.12 Creating a table database and table metadata

This user scenario describes the process of creating a table database and table metadata for the table-based application.

By following the steps below, you will learn how to prepare appropriate configuration files for a table database and table metadata and import them into the table-based application. For more information about data organization for table archiving, see Section 3 “Configuring a table application” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

The following are prerequisites:

- OpenText Information Archive is installed and started.
- The appropriate database `<YOUR_TABLE_DATABASE_NAME>` and data node `<YOUR_DATA_NODE_NAME>` are created. See the scenario Creating storage resources: data nodes, databases, and storage systems.

Steps:

1. Prepare the table database and metadata configuration files.

   You can refer to configuration from the table-based example application *Patent* to verify your progress: `<IA_ROOT>/examples/applications/Patent/config/application-config/data- model-config`.

   a. Create a folder for your configuration files. It can be placed anywhere on your drive. For convenience, we create it in the OpenText Information Archive example applications folder here: `<IA_ROOT>/examples/applications/<YOUR_TABLE_APPLICATION_NAME>/mytable_config`.

   b. Inside the `mytable_config` folder, prepare two configuration files.

      You should have the following file structure:

      - `configuration.yml`
      - `database-<YOUR_TABLE_APPLICATION_NAME>SqlDb.xml`

   c. Prepare the appropriate table metadata describing the corresponding table data in the `database-<YOUR_TABLE_APPLICATION_NAME>SqlDb.xml` file.

      For more information, see Section 3.2.3.1 “Storing extracted text metadata” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

      These are some of the important values that require attention:

      - `metadata.schemaMetadataList.schemaMetadata.name`: The schema name value that will be a reference value for other parts of the table configuration.

- `metadata.schemaMetadataList.schemaMetadata.tableCount`: The number of tables to ingest. For each table, the corresponding `tableMetadata` section is required
- `metadata.schemaMetadataList.schemaMetadata.tableMetadataList`: The list of metadata for each table.
- `tableMetadata`: Defines the table name, the number of records (`recordCount`) in it and the list of columns (`columnList`).
- Each `column` defines the corresponding attributes for a column:
  - `name`: A `<YOUR_TABLE_COLUMN_NAME>` string value for the name of the field
  - `type`: The type of the data for the field (INTEGER, DATE, VARCHAR *etc.*)

    For the list of acceptable SQL types and query translation, see the following section and its subsections: Section 3.2 “Configuring databases, schemas and tables” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)* and Section 11.4 “Mapping table SQL data types to PostgreSQL data types” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.
- `typeLength`: The maximum string length of the value.
- `indexing`: The multi-path index value. The possible values are: FULL_TEXT, VALUE, FULL_TEXT_AND_VALUE.

  For more information, see Section 3.4 “Table archive indexing” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.
- `encrypt`: The Boolean value enabling encryption for the field.

  For more information about column encryption, see Section 5.1 “Encryption process” in *OpenText Information Archive - Encryption Guide (EARCORE-AGE)*.

d. Fill in the `configuration.yml` file:

- `namespaces`: Optionally, specify the prefix and URI values for namespace.
- `tenant.name`: Specify the name of the tenant. A tenant is a logical configuration object in archive system that presents a customer business item for preserving and storing the data. Tenants store zero or more applications. In this example, the tenant value is `INFOARCHIVE`.
- `tenant.configure`: Specify whether the tenant should be created, updated or the existing tenant should be used. In this example, set it as `use existing`.
- `application.name`: Specify the name of your table application, `<YOUR_TABLE_APPLICATION_NAME>`.
- `application.configure`: In this example, set it as `use existing`.

   - `stores.name`: Specify the name of your table store, `<YOUR_TABLE_STORE_NAME>`.
   - `stores.configure`: In this example, set it as `use existing`.
   - `database.name`: Specify the table database name `<YOUR_TABLE_APPLICATION_NAME>-sql-db`.
   - `database.ciStore`: The <YOUR_TABLE_STORE_NAME> store name for unstructured content.
   - `database.defaultSchema`: The <YOUR_TABLE_SCHEMA_NAME> value specified in metadata file.
   - `database.indexingOnIngest`: The Boolean value that enables rebuilding the index on each ingest. By default, it is false as it may affect ingestion performance.
   - `database.retentionBackupStore`: The `<YOUR_TABLE_STORE_NAME>` store name for retention data backups.
   - `database.retentionDataStore`: The `<YOUR_TABLE_STORE_NAME>` store name for granular retention data.
   - `database.metadata – resource`: The table metadata file relative path. In this example: `database-<YOUR_TABLE_APPLICATION_NAME>SqlDb.xml`
   - database.backupStore: The `<YOUR_TABLE_STORE_NAME>` reference for the `store.name` value. Cannot be configured if the `store.name` is not specified.

e. Verify that the `configuration.yml` file has the corresponding structure, but with proper resources names.

2. Import the configuration with IA Shell.

a. Launch IA Shell:

   - On Linux: `<IA_ROOT>/bin/iashell`
   - On Windows: `<IA_ROOT>/bin/iashell.bat`

b. Connect as a user with the Developer role.

```
iashell> connect --user connie@iacustomer.com --password password
Connected to http://localhost:8765/services as connie@iacustomer.com.
```

c. Import the table configuration prepared on step 1 with the `import` command, as shown below:

```
iashell> import
<INSTALL_DIR>/examples/applications/<YOUR_TABLE_APPLICATION_NA ME>/
mytable_config

import_configuration1539092898710.zip 100% [==============]
2420/2420 (0:00:01 / 0:00:00)
Skipped tenant: INFOARCHIVE (already exists)
Skipped application: <YOUR_TABLE_APPLICATION_NAME> (already exists)
Skipped store: <YOUR_TABLE_STORE_NAME> (already exists)
Created database: <YOUR_TABLE_APPLICATION_NAME> -sql-db
Created schema: <YOUR_TABLE_SCHEMA_NAME> Created table: <YOUR_TABLE_NAME>
OK
```

3. Verify the table metadata has been ingested with IA Web App.

    a. Log in to IA Web App as `connie@iacustomer.com` (Developer).

    b. Navigate to **Applications > `<YOUR_TABLE_APPLICATION_NAME>` > Databases** tab.

    c. Verify that:

        - The table database with the name `<YOUR_TABLE_APPLICATION_NAME>-sql-db` is displayed.
        - The side panel tab contains the details corresponding to the configuration file completed on step 4.

    d. Navigate to **Applications > `<YOUR_TABLE_APPLICATION_NAME>` > Tables** tab.

    e. Verify that:

        - The table `<YOUR_TABLE_NAME>` is displayed in the **Tables** grid.
        - The Reference Records for the table equals the `recordCount` value specified on step 3.
        - The side panel tab contains the details corresponding to the table metadata configuration file completed in step 3.

1.13 Ingesting tables

This user scenario describes the process of preparing and ingesting table data to the table-based application.

Following the steps below you will learn how to prepare appropriate table data with structured content, ingest the table data to the table-based application, and then verify it and build an index. For more information about data organization for table-based applications, see Section 3 “Configuring a table application” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

The following are prerequisites:

- OpenText Information Archive is installed and started.
- The application `<YOUR_TABLE_APPLICATION_NAME>` is created and can be seen in IA Web App.
- The appropriate table metadata for `<YOUR_TABLE_NAME>` table and the table database `<YOUR_TABLE_APPLICATION_NAME>-sql-db` have been imported in the scenario Creating a table database and table metadata.
- The business data to be archived is prepared on a local disk.
    - You may already have the data in XML format prior to ingest.
    - If, at this point, you only have a system ready to extract the data, follow the ETL process. You can use any convenient ETL tool for XML generation like Talend Open Studio or a custom script.

Steps:

1. Prepare the table data as mentioned in the prerequisites.

   You can refer to table data from the table-based example application Patent to verify your progress: `<IA_ROOT>/examples/applications/Patent/config/data`.

   - Create a folder and place the table data in the following path: `<IA_ROOT>/examples/applications/<YOUR_TABLE_APPLICATION_NAME>/my TableData/<YOUR_TABLE_APPLICATION_NAME>_<YOUR_TABLE_NAME>_RECORDS.xml`.

     Pay attention to the root element name of the table data xml file – it should be equal to the `<YOUR_TABLE_SCHEMA_NAME>` value.

2. Ingest the table data with IA Shell.

   - On Linux: `<IA_ROOT>/bin/iashell`
   - On Windows: `<IA_ROOT>/bin/iashell.bat`

   Refer to the *OpenText Information Archive - OpenText Information Archive Shell Guide (EARCORE-ARE)* to learn more about the tool, its commands and how to use it.

   a. Connect as a user with the Developer role.

   ```
   iashell> connect --user connie@iacustomer.com –password password
   Connected to "http://localhost:8765/services" com as connie@iacustomer.com.
   ```

   b. Ingest the table data prepared on step 1 with the `ingest` command, as shown below:

   ```
   iashell> ingest --d applications/<YOUR_TABLE_APPLICATION_NAME>/databases/
   <YOUR_T ABLE_APPLICATION_NAME>-sql- db/schemas/<YOUR_TABLE_SCHEMA_NAME> --
   from <IA_ROOT>/examples/applications/<YOUR_TABLE_APPLICATION_NAME>/myTableData

   Starting ingestion for file:
   <IA_ROOT>\examples\applications\<YOUR_TABLE_APPLICATION_NAME>\myTableData\
   <YOUR_TABLE_APPLICATION_NAME>_<YOUR_TABLE_NAME>_RECO
   RDS.xml
   Ingested file:
   <IA_ROOT>\examples\applications\<YOUR_TABLE_APPLICATION_NAME>\myTableData\
   <YOUR_TABLE_APPLICATION_NAME>_<YOUR_TABLE_NAME>_RECORDS.xml
   OK
   ```

3. Check the data with the `chain-of-custody` command.

   For more information, see Section 2.8.2 “chain-of-custody” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*.

4. Build the index for the ingested table data with the `index-build` command.

   This IA Shell command automatically runs the Table Indexing job (this job cannot be run manually). For more information, see Section 3.7.23 “Table Indexing job” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

```
iashell> index-build
Started building index for 'Baseball-sql-db'
=================================================================================
```

```
jobInstanceLink.href http://localhost:8765/systemdata/job-instances/7abad4ce-1b9f
-4ac5-bd2d-a528a7f98917
startDateTime null
activities[0] define
activities[1] construct
activities[2] move
status SCHEDULED
-----------------------------------------------------------------------------
```

For more information about the `index-build` command, see Section 2.8.4 “index-build” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*.

5. Once the Table Indexing job has finished, it is possible to check the indexes with the IA Shell command `index-status`. For more information, see Section 2.8.8 “index-status” in *OpenText Information Archive - IA Shell Guide (EARCORE-ARE)*.

1.14 Creating and running a search for a table application

This user scenario describes the process of the search operation for table-based applications.

By following the steps below, you will create and use a search over data ingested by a table-based application. For more information, see Section 6 “Setting up searches for an application” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

The following are prerequisites:

- OpenText Information Archive is installed and started.
- The appropriate table data for the application `<YOUR_TABLE_APPLICATION_NAME>` is ingested and verified (see the scenario Ingesting tables).

Steps:

1. Log in to IA Web App as a user with the Developer role. A Developer is allowed to manage application configuration resources. You can use your own Developer use, or use the `connie` user from the out-of-the-box example user accounts like below:

   - Username: `connie@iacustomer.com`
   - Password: `password`

   After a successful login you will be redirected to the **Applications** page.

2. Navigate to **Applications >** **`<YOUR_TABLE_APPLICATION_NAME>`**. The list of searches is empty.

3. Create a new search.

   For more information about search creation, see Section 6.5 “Composing a search form using IA Web App” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

a. Click the + (create) button and then select **Create Search**. The Create Search form opens.

b. Complete the fields in the **Start** tab:

- Set the name of the new search as `<YOUR_TABLE_SEARCH_NAME>`.
- The **Archival Collection** field is predefined with the table database name `<YOUR_TABLE_APPLICATION_NAME>-sql-db` value.
- The schema field is predefined with the `<YOUR_TABLE_SCHEMA_NAME>` value.
- Optionally, if your query will only query a single table, select the table that the search form will access when run. If selected, the designer is presented with a list of tables. Browse through the list or conduct a search to locate a specific table and click the **Select** button.

c. Click **Create**.

- The search `<YOUR_TABLE_SEARCH_NAME>` is created and the **Edit Search** set view is displayed.
- The search status is Draft.

4. Configure the new search.

   A search configuration is a very flexible process. The complexity may vary significantly depending on the search purpose.

   In this example, we assume that the search should query over a string value of a table column `<YOUR_TABLE_COLUMN_NAME>`.

a. Navigate to the **QUERY EDITOR** tab and define the appropriate query for the search.

- For convenience, there is a table schema displayed on the right panel of the query editor.
- Note that the query should return variables with appropriate names, so the Search form fields could be bound to the data returned by the query.

  For more information, see Section 6.2.3 “Creating a single application search” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

b. Add form fields for the search form.

- Navigate to the **Search Form** tab.
- Click the **+ Add Form Field** button and select the type of the element to display. In this example, it is `Input`.

c. Edit the new form field with the appropriate configuration values:

- **Control Type**: Set the type of data to search. In this example, it is `Text`.
- **Binding Type**: Specify whether a Single or Composite Search is required. In this example, it is Single.

- **Data Binding**: Set the name of the variable in the query.
- **Data Resolution**: Not a required field, in this example leave as is. 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 more information. see Section 6.5.14 “Configuring data resolution” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.
- Other values like **Tooltip Text** and **Required** are intuitively easy to understand and will not be covered in this scenario. Check the rest of the fields and specify values, if needed.

d. Click **OK**.

e. Configure the **Result List** view. For more information, see Section 6.6 “Composing the search result” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

f. Navigate to the **Result List** tab and click the **+ Add Column** button. The new column, **Column 1**, is created.

- Edit the new column:
  - **Column Label**: Set the displayed label for the column.
  - **Column Type**: Set the column type.
  - **Binding**: Specify the corresponding table column that is going to be used to display data.
  - **Sensitive Information**: Specify if the data should be encrypted for the background cache. If the binding is set to a field that is known to be encrypted, the field cannot be changed.
  - **Data Type**: Set the appropriate data type corresponding to the table column type in Binding.
- Click **OK**.
- Repeat adding as many columns as needed.

g. Change the status of the search to Ready:

- Open the edit search form by click on the gear button on the right of the search name.
- Change the **Status** to Ready.
- Click **OK**.

h. Click **Save**.

5. Run the created search.

   a. Navigate to **Applications >** **`<YOUR_TABLE_APPLICATION_NAME>`**.

   You should see the new search in the grid.

   b. Select the `<YOUR_TABLE_SEARCH_NAME>` search.

The search form with the corresponding search fields is opened.

c. Set the search parameters in compliance with the search form fields added in step 2.

d. Click **Search**.

6. Verify the result.

   - The number of rows should be corresponding the query and the search parameters specified in step 3.
   - The result search data is presented in compliance with the **Result List** tab configured in step 3.