OpenText™ Information Archive

IA Shell Guide

Learn how to use the CLI to create, read, update, and delete configuration resources. Install applications, holdings and searches. Perform actions on packages and complete a range of administrative tasks, from reviewing product configuration to starting jobs.

EARCORE250400-ARE-EN-01

OpenText™ Information Archive
IA Shell Guide
EARCORE250400-ARE-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

OpenText Information Archive includes a tool called the IA Shell, which is typically used by administrators and developers.

Among other tasks, IA Shell allows a user to:

- Connect to OpenText Information Archive as different users.
- Create, read, update, and delete configuration resources, such as storages, spaces, stores, *etc.*.
- Install applications, holdings and searches into OpenText Information Archive.
- Ingest data into table and SIP applications.
- Perform actions on packages (for example, reject, invalidate, cache in/cache out).
- Complete a wide range of administrative tasks, from reviewing product configuration to starting jobs.

Using IA Shell, it is possible to create automated scripts, for example, for installing applications and ingesting data.

The tool is used to install both first-time applications and sample out-of-box applications.

1.1 Downloading IA Shell from IA Web App

It is possible to download the IA Shell package from the IA Web App. To download IA Shell, click your user name in the top-right corner of the page and select **Download IA Shell**.

The download archive contains IA Shell launch scripts for all supported operating systems, `JAR` files, and configuration files `config/iashell/application.yml` and `config/iashell/application-https.yml` in `YML` format. For more information, see Configuration files.

`config/iashell/application.yml`

When you download IA Shell from the IA Web App, the `userName` value in the `application.yml` file is empty. Be sure to enter `userName` and `userPassword` values prior to configuring IA Shell.

This file contains the following:

	Mandatory	Description	Example
`connection.gatewayUrl`	Yes	Gateway URL if you are not using SSL.	`http://10.145.118.37:8080`
`connection.clientSecret`	Yes	Secret key is not extracted due to security reasons. You must specify required client secret before using the downloaded IA Shell. Ask your Administrator to provide you this information.	`25HA9UVdf6fCWYp50955h4bXrqJDnwmw`
`connection.userName`	No	User name is extracted in fully qualified form. It might include a partition name for SSO or domain name for LDAP authentication.	`connie@iacustomer.com`
`connection.userPassword`	No	User password is not extracted due to security reasons.	`password@123`
`webAuth.loginUrl`	No	In case of SSO, web authentication login and logout URLs.	`null`
`webAuth.logoutUrl`	No		`null`

config/iashell/application-https.yml

This file contains the following:

	Mandatory	Description	Example
`connection.gatewayUrl`	Yes	Gateway URL if you are using SSL	`https://10.145.118.37:8081`
`ssl.trustStore`	No	If you are using OTDS SSO or SSL for the IA Web App and IA Server, ask your Administrator to provide you the trustStore file with the associated password and type.	`../config/iashell/gatewayTrustStore.jks`
`ssl.trustStorePassword`	No		`<PASSWORD>`
`ssl.trustStoreType`	No		`JKS`

Note: No user passwords and secret keys are extracted due to security reasons. You must specify the required passwords and client secrets before using the downloaded IA Shell.

The HTTPS configuration is not generated when the IA Web App runs on Apache Tomcat or it is located behind TLS terminating load balancer.

Complete the following steps to configure IA Shell for use:

If you do not want to be prompted for the password when running IA Shell commands interactively or via script, specify your password; otherwise, you are prompted for a password when IA Shell runs.

If you are not using OTDS/SSO, update the `connection.clientSecret` to the value specified in `iawebapp\config\application-CLIENTS.yml` file.

If you are using OTDS/SSO, the `connection.clientSecret` comes from the answer provided during setup.

If you are using OTDS/SSO or SSL for the IA Web App and IA Server, ask your Administrator to provide you the KeyStore file with the associated password and type. Update the `config/iashell/application-https.yml` file with this information.

1.2 Starting IA Shell

The tool can be found in the OpenText Information Archive installation folder by this relative path:

- For Windows: `bin > iashell.bat`
- For Linux: `bin > iashell`

To start IA Shell, invoke it from the command line, as illustrated below:

```
[/info-archive] % bin/iashell
```

1.3 Running commands

After IA Shell is launched, the user can run commands. While most IA Shell commands interact with the IA Server, some configuration commands can be used in before doing the connect command. Refer to Chapter 2 – IA Shell commands for more information about each command.

The first command to run is the `help` command. It provides a quick view of existing IA Shell commands along with a short explanation for each.

To use the `help` command:

```
iashell> help
```

The `help` command also indicates if a connection is required to perform the commend.

All the commands that interact with the IA Server are unavailable until a connection has been made.

If IA Shell is not connected to OpenText Information Archive, a message is issued when a command is run indicating that a connection has to be established. For example:

```
iashell> ingest
Command 'ingest' exists but is not currently available because connection is not
established
Details of the error have been omitted. You can use the stacktrace command to print the
full stacktrace.
```

1.4 Connecting to OpenText Information Archive

Unless connected, very few actions can be performed in IA Shell.

To connect to OpenText Information Archive, run the `connect` command, as illustrated in the following example:

```
iashell> connect
Please, enter user name:sue@iacustomer.com
Please, enter password for user 'sue@iacustomer.com':: password
Connected to "http://localhost:8080" as sue@iacustomer.com
iashell>
```

Note: It is possible as a convenience feature to store the user name and password in the `application.yml` file to avoid having to type credentials. For more information about these values, refer to the connection information in Configuration files.

To learn more about the `connect` command, and the possible methods of connection to OpenText Information Archive and the available parameters, refer to the command description in Chapter 2 – IA Shell commands.

There are several instances when the connect command might fail, even when used with a valid username and password:

- The URL is not reachable. This might be because the IA Web App or IA Server are not running.
- The connection token that is retrieved from Gateway or provided explicitly with the `connect` command is not correct.
- HTTPS-related settings and/or certificates are not correct, if applicable. If the profile is included, TLS/SSL configuration becomes available for IA Shell to interact with the IA Server/Gateway. The TLS/SSL settings are configured in `config/iashell/application-https.yml` file. For more information, see section 13.4 “Configuring OTDS for SSO mode” in *OpenText Information Archive - Installation Guide (EARCORE-IGD)*.

Important

IA Shell supports using proxies to connect to an external OpenText Information Archive instance, but the use of proxies must be enabled first. To enable HTTP proxy for IA Shell, complete the following:

Linux environment:

- For proxy configuration, set the `IASHELL_OPTS` to the following value: `-Djava.net.useSystemProxies=true`. This is an optional environment variable for IA Shell that, when set, is sent to JVM as a parameter.
- For `no_proxy` configuration, the `IASHELL_OPTS` environment variable is optional, regardless of whether we configure it or not. Note that this is in context of a correctly configured `no_proxy` environment variable. If that environment variable is not set, do not set `IASHELL_OPTS` at all.

  The following is an example of `IASHELL_OPTS` in Linux Red Hat environment:

  ```
  IASHELL_OPTS='"-Djava.net.useSystemProxies=true"'
  ```

Windows environment:

- For proxy configuration, set the following `IASHELL_OPTS` environment variables:
  - `-Djava.net.useSystemProxies=true`
  - `-Dhttp.proxyHost=xx`: `xx` indicates the sample host.
  - `-Dhttp.proxyPort=yy`: `yy` indicates the sample port.
  - `-Dhttps.proxyHost=xx`
  - `-Dhttps.proxyPort=yy`

  This is an optional environment variable for IA Shell that, when set, is sent to JVM as a parameter.
- For `no_proxy` configuration, set the `IASHELL_OPTS` environment variable with the additional value: `-Dhttp.nonProxyHosts=xx` . Note that this is in context of a correctly configured `no_proxy` setting in Windows.

  The following is an example of `IASHELL_OPTS` in a Windows 10 environment:

  ```
  set IASHELL_OPTS="-Djava.net.useSystemProxies=true" "-Dhttp.proxyHost=bp2-
  prox01-l001.otxlab.net" "-Dhttp.proxyPort=3128" "-Dhttps.proxyHost=bp2-prox01-
  l001.otxlab.net" "-Dhttps.proxyPort=3128" "-Dhttp.nonProxyHosts=10.245.*|
  10.194.*|10.9.*|10.145.*|*.otxlab.net|confluence.opentext.com"
  ```

1.5 Configuration files

Three configuration files located in the <INFOARCHIVE_ROOT>/config/iashell folder allow you to override IA Shell's configuration:

- The application.yml file contains properties with default values for connection to OpenText Information Archive and settings for password encryption.

```
connection:
  clientSecret: <client secret>
  gatewayUrl: http://localhost:8080
  restApiUrl: http://localhost:8765/services
  tenant: INFOARCHIVE
  userName:
  userPassword:
```

- The logback-spring.xml file contains settings to be used for enabling different levels of logging. For enabling the DEBUG log level, see Logging.
- IA Shell can work over a TLS/SSL communication protocol. The application-https.yml file is used to configure TLS/SSL connection. For more information, see TLS/SSL configuration.

The following describes the properties in the connection section of the application.yml file:

connection.gatewayUrl

Indicates the default address of the Gateway.

connection.clientSecret

Indicates IA Shell client secret key. The property value must match the value specified in the config\iawebapp\application-CLIENTS.yml file for client with the ID infoarchive.cli. In case of OTDS single sign-on (SSO) mode, the clientSecret has to match the value provided during the setup process. When the clientSecret is specified in the properties file, the value should be encrypted. For more information, see section 7.2.2 “Password encryption utility” in *OpenText Information Archive - Encryption Guide (EARCORE-AGE)*.

connection.userName

Indicates the default user name to connect to the server.

Caution

For security reasons, do not enter your userName and userPassword information unless you are the only one using IA Shell.

connection.userPassword

Indicates the default user password that is used during the connection with the server. When the password is specified in the properties file, and security is of any concern, the password must be encrypted. For more information, see section 7.2.1 “Encrypting database passwords in configuration files for applications” in *OpenText Information Archive - Encryption Guide (EARCORE-AGE)*.

`connection.tenant`
: Indicates the name of the default tenant to be used when connecting to the server.

- The following describes the properties in the `defaultSettings` section of the `application.yml` file:

  `other.uploadGatewaySocketTimeout`
  : Indicates the default time-out when uploading files when IA Shell is configured to connect the IA Server using the gateway. The value is in milliseconds. For more information, see Ingesting large files.

  `pathToDownloadDirectory`
  : Indicates the path to store the downloaded content. By default, when the property is not specified, downloaded content is placed in the directory from where IA Shell has been run.

  `sip.pathToErrorFolder`
  : The properties specify the directory where a SIP's packages are moved when ingestion is unsuccessful. The path can be absolute or can be relevant to the directory where the SIP is placed.

  `sip.pathToSuccessFolder`
  : The properties specify the directory where a SIP's packages are moved after ingestion. The path can be absolute or can be relevant to the directory where the SIP is placed.

  `sip.sipFormat`
  : Indicates the format of the SIP file to be used by default during ingestion. Pre-populated with `sip_zip` value.

  `sip.sipIngestThreads`
  : Indicates the amount of concurrent threads to be used for SIP data ingestion. Pre-populated with value `1`.

  `table.custodySchemaFileName`
  : Indicates the name of the schema file when running the `chain-of-custody` command against tables.

  `table.custodyTableFileName`
  : Indicates the name of the table file when running the `chain-of-custody` command.

  `table.enableChainOfCustodyInBatch`
  : Indicates whether to run the `chain-of-custody` command when running the IA Shell script in batch mode.

  `table.enableIndexBuildInBatch`
  : Indicates whether to run the `index-build` command when running the IA Shell script in batch mode.

  `table.pathToChainOfCustodyFiles`
  : Indicates the path to the files when running the `chain-of-custody` command against tables.

`table.simpleTableFilesFilter`
: Allows you to specify a default filter for selecting the files with structured data to be ingested when running the `ingest` command for table applications. For example, `*.xml`.

`table.tableIngestThreads`
: Indicates the amount of concurrent threads to be used for table data ingestion. Pre-populated with value `4`.

`table.tableMetadataFilesFilter`
: Allows you to specify a default filter for selecting the files with metadata information to be ingested when running the `ingest` command for table applications. For example, `metadata*.xml`.

`table.allowedUnstructuredDataRoot`
: Use the `allowedUnstructuredDataRoot` property in the `iashell/application.yml` file to configure the root directory for unstructured data for the referenced files of table applications. By default, the property value is blank.

```
table:
    allowedUnstructuredDataRoot:
```

The following properties are in the `passwordEncryption` section of the `application.yml` file. It is highly recommended to allow the setup program to set these values on your behalf.

`passwordEncryption.enabled`
: Indicates if the secret key and passwords are encrypted.

`passwordEncryption.encryptionAlgorithm`
: Which algorithm was used to encrypt the passwords.

`passwordEncryption.keyStorePath`
: Path to the keystore.

`passwordEncryption.keyStoreType`
: Type of keystore, the setup program defines which store types are supported.

`noPromptStartup`
: If the passwords are encrypted, the setup program can generate some files that allow IA Shell to startup. If the passwords are encrypted and this is set to false, each time IA Shell is started, the user will be prompted for the keystore password.

Note that not all of the parameters are documented here, as there are also options to use Gemalto. For more information, see section 3 “Connecting system components to CipherTrust/KeySecure appliance” in *OpenText Information Archive - Encryption Guide (EARCORE-AGE)*.

The following properties of the `application.yml` file allows the management of the included profiles to IA Shell:

`spring.profiles.include`
These properties allows you to enable profiles for IA Shell. The following profiles are supported:

`https`
If the profile is included, TLS/SSL configuration becomes available for IA Shell to interact with the IA Server/Gateway. The TLS/SSL settings are configured in `config/iashell/application-https.yml` file.

`infoarchive.cli.webAuth`
If the profile is included, then web-based authentication with OTDS site minder is available. Refer to Web authentication for IA Shell section and the connect command description for more information about this authentication method.

The following properties are used for IA Shell configuration with the web authentication mode when the `infoarchive.cli.webAuth` profile is enabled:

`webAuth.loginUrl`
Specifies the OAuth2 authorization endpoint URL.

`webAuth.logoutUrl`
Specifies OAuth2 logout endpoint URL.

For more information about crypto objects, see section 4.2 “Configuring crypto objects” in *OpenText Information Archive - Encryption Guide (EARCORE-AGE)*.

It is possible to override the values of the files in individual directories. This can be done by creating a `local_iashell.properties` file in a directory to store properties you wish to override in it (for example, specific applications). Refer to Run modes for further information about the directory. When the `iashell` command is then started from such a directory, the properties specified in this file take precedence over the values specified in `application.yml` file.

1.5.1 Optional values in the config-templates.yml file

The following is an optional value in the `config-templates.yml` file:

`connection.restApiUrl`
Indicates the URL to connect to the IA Server directly when IA Shell makes REST calls. Normally, this value is not set. This means IA Shell communicates with the Gateway, which forwards the requests to the IA Server. This value must not be set for cloud deployments.

1.6 Run modes

IA Shell provides the possibility to run commands in two modes:

Interactive mode
This is the mode used when the user runs commands and observes the immediate results in IA Shell;

Batch mode
This is the mode used when the user prepares the script with a list of commands to be run sequentially, one following another. This mode is used for installing declarative configuration (DC) sample applications.

To use IA Shell in batch mode, create a file in which to store commands to be run. Every command should be placed on a new line, as illustrated in the following example.

```
connect
import config
ingest applications/Invoices --from C:/data/sips
```

It is recommended to create either an `install.bat` (Windows) or `install` (Linux). The following is an example for Windows of a script (all of the sample applications and first time applications use this pattern:

```
@echo off
call ..\..\bin\iashell script install.iashell
```

1.7 Logging

IA Shell logs the following information:

- Run commands in both interactive and batch modes. From a security point of view, all sensitive information about passwords and tokens is masked in the output. Commands in human readable form are logged into the `iashell-history.log`. In addition, IA Shell maintains a history of used commands in a separate file `iashell-cmd.util`. This allows user to navigate through the commands history in interactive mode.
- Errors that occurred during a command's run. IA Shell returns a zero code only if it completed all tasks successfully or return a non-zero value if any task failed.
- General logs and debug information.

All logs are in the `<INFOARCHIVE_ROOT>/logs/iashell` folder.

By default, the log level is set to `WARN`, which prevents the output from including unnecessary information and speeds up command runtimes. However, if it is required to debug the IA Shell, it is possible to set `DEBUG` and `TRACE` log levels for IA Shell itself, as well as for its third-party components.

To achieve this, change the `<INFOARCHIVE_ROOT>/config/iashell/logback-spring.xml` file by replacing `level="WARN"` for all the loggers to the required level, either `INFO`, `DEBUG` or `TRACE`. The IA Shell needs to be restarted for the change to take effect.

1.7.1 Disabling IA Shell logging

By default, IA Shell logs at `WARN` level and logs what commands the user did as history. It is possible to change both settings via the IA Shell’s `application.yml` file to change the log level or disable all history logging.

The `iashell-history.log` maintains a list of all commands run by IA Shell. The system, however, allows you to further limit the amount of logging information. After setup generates the `<IA_ROOT>/config/iashell/application.yml` file, update it to:

- Disable the system from logging every command into the `iashell-history.log` file.
- Update the default log level (for example, to `ERROR`) to further limit the amount of logging information that is recorded.

The `<IA_ROOT>/config-templates/iashell/application.yml` file includes an example of how to make these changes:

```
# Uncomment this to change the root logging level or disable logging of the command in
iashell-history.log
#logging:
#  level:
#    root: ERROR
#    com.emc.ia.cli.services.historylog.TraceableShell: OFF
```

Caution

You must not change anything in the `<IA_ROOT>/config-templates/iashell/application.yml` file. Simply use the information to update the `<IA_ROOT>/config/iashell/application.yml` file.

1.8 TLS/SSL configuration

If either the IA Server or Gateway is configured to use TLS/SSL, IA Shell should also be properly configured to interact with it. By default, TLS/SSL is not enabled for IA Shell.

IA Shell supports either one-way or two way TLS/SSL. For more information, see section 13.2.2 “Setting up TLS/SSL in a demo configuration” in *OpenText Information Archive - Installation Guide (EARCORE-IGD).*

1.8.1 TLS/SSL specific properties

To configure IA Shell to use TLS/SSL, it is required to include the `https` profile in the `config/iashell/application.yml` properties file. It can be done with help of the `spring.profiles.include` property. Once the profile is included, the settings for HTTPS configuration are taken from `config/iashell/applications-https.yml` properties file.

The file contains following settings to configure either a one-way TLS/SSL or two way TLS/SSL:

`ssl.trustStore`
Indicate the path to the TrustStore container file with CA certificates.

`ssl.trustStorePassword`
Indicate the password for the TrustStore. When the password is specified in the properties file, and security is of any concern, the value must be encrypted. For more information, see section 7.2.2 “Password encryption utility” in *OpenText Information Archive - Encryption Guide (EARCORE-AGE)*.

`ssl.trustStoreType`
Indicate the type of TrustStore container (JKS or PKCS12).

For a two-way TLS/SSL, configure the above TrustStore settings and configure the following KeyStore settings:

`ssl.keyStore`
Indicate the path to the KeyStore container file that includes the KeyStore certificates and authentication private key.

`ssl.keyStorePassword`
Indicate the password for the KeyStore. When the password is specified in the properties file, and security is of any concern, the value must be encrypted. For more information, see section 2 “Managing cryptographic keystores” in *OpenText Information Archive - Encryption Guide (EARCORE-AGE)*.

`ssl.keyStoreType`
Indicate the type of KeyStore container (JKS or PKCS12).

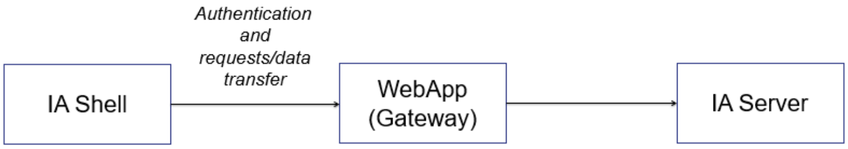
1.8.2 IA Shell TLS/SSL interaction cases

The IA Shell TLS/SSL configuration depends on how the IA Web App (Gateway) and IA Server are configured from a TLS/SSL point of view.

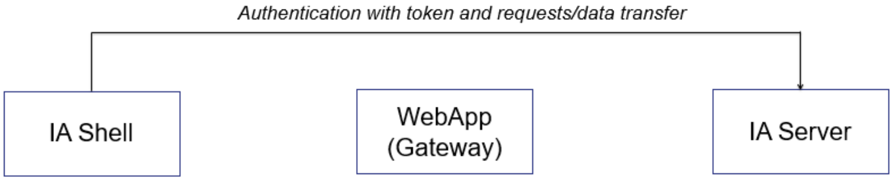
In general, IA Shell can talk to both the IA Web App (Gateway) and IA Server.

There are three main use cases:

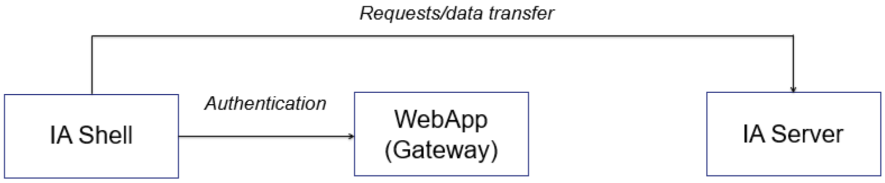
1. IA Shell talks with the IA Web App (Gateway) to retrieve a token and sends the request to the IA Servers over the gateway. There is no direct interaction with the IA Server, as illustrated in the following diagram:



2. IA Shell talks with the IA Server only by usage of predefined token:



3. IA Shell talks with the IA Web App (Gateway) only for authentication and receiving the token, with which IA Shell goes to the IA Server directly:



For more information about interactions and authentication mechanisms, please, refer to connect command.

At the same time, between components (IA Shell/IA Web App (Gateway)/IA Server) can be HTTP or HTTPS protocol used.

1.9 Web authentication for IA Shell

When IA Shell establishes the connection with Gateway or the IA Server, the user name and password from the `config/iashell/application.yml` file are used. However, if the user name and password is not specified in the properties file, the user should enter the names interactively during `connect` command runs.

There is an authentication system that enforces the use of a web browser for authentication instead of interactively entered credentials in command line. In this case, the web authentication mode can be used to authenticate a user when performing the connection.

Refer to the `infoarchive.cli.webAuth` description in the Configuration files section.

1.10 Authentication via external the LDAP/AD/OTDS server

In general, there is no specific configuration for IA Shell to perform authentication via Lightweight Directory Access Protocol (LDAP), Active Directory (AD), or OpenText Directory Services (OTDS). All that is required is to run the `connect` command with the proper user name and password.

The following steps outline the process to configure the IA Shell to work with an external Lightweight Directory Access Protocol (LDAP) server.

However, the following section illustrates how to configure OpenText Information Archive to use external LDAP, and how to connect IA Shell to the server.

1.10.1 Example of OpenText Information Archive configuration with LDAP

The following steps outline the process to configure IA Shell to work with an external Lightweight Directory Access Protocol (LDAP) server.

1. Download the latest OpenText Information Archive distribution.

   The settings described below are related to the `<INFOARCHIVE_ROOT>/config/webapp/application.yml` file.

2. Find the `spring.profiles.active=infoarchive.gateway.profile.AUTHENTICATION_IN_MEMORY` string.

3. Set the string to `spring.profiles.active=infoarchive.gateway.profile.AUTHENTICATION_IN_MEMORY,infoarchive.gateway.profile.AUTHENTICATION_ACTIVE_DIRECTORY` (or `spring.profiles.active=infoarchive.gateway.profile.AUTHENTICATION_IN_MEMORY,infoarchive.gateway.profile.AUTHENTICATION_EXTERNAL_LDAP` if you use external LDAP).

4. Configure the `application-infoarchive.gateway.profile.AUTHENTICATION_ACTIVE_DIRECTORY.properties` file with AD connection settings.

5. Configure the AD in the `application-infoarchive.gateway.profile.AUTHENTICATION_ACTIVE_DIRECTORY.properties` file (or `application-infoarchive.gateway.profile.AUTHENTICATION_EXTERNAL_LDAP.properties` file if using External LDAP).

   Example for Active Directory

```
AUTHENTICATION_ACTIVE_DIRECTORY.managerDn=cn\=Administrator,cn\=users,dc\=iigads,dc
\=com
AUTHENTICATION_ACTIVE_DIRECTORY.managerPassword=Password@123
AUTHENTICATION_ACTIVE_DIRECTORY.userDnPatterns=cn={0},ou\=Users,ou\=infoarchive,dc
\=iigads,-dc\=com
AUTHENTICATION_ACTIVE_DIRECTORY.userSearchFilter=sAMAccountName\={0}
AUTHENTICATION_ACTIVE_DIRECTORY.userSearchBase=ou\=Users,ou\=infoarchive,dc
\=iigads,dc\=com AUTHENTICATION_ACTIVE_DIRECTORY.groupSearchFilter=(member\={0})
AUTHENTICATION_ACTIVE_DIRECTORY.groupSearchBase=ou\=Groups,ou\=infoarchive,dc
```

```
\=iigads,dc\=com
AUTHENTICATION_ACTIVE_DIRECTORY.url=ladp://ldaphostipaddress:389/
```

Example for External LDAP

```
AUTHENTICATION_EXTERNAL_LDAP.userDnPatterns=uid={0},ou\=people
AUTHENTICATION_EXTERNAL_LDAP.userSearchFilter=uid={0}
AUTHENTICATION_EXTERNAL_LDAP.userSearchBase=ou\=people
AUTHENTICATION_EXTERNAL_LDAP.groupSearchFilter=(member\={0})
AUTHENTICATION_EXTERNAL_LDAP.groupSearchBase=ou\=groups
AUTHENTICATION_EXTERNAL_LDAP.url=ldap://localhost:8080/dc\=infoarchive,dc\=emc,dc
\=com
```

6. Install OpenText Information Archive.

7. Start PostgreSQL and server.

8. Start the web application and login as an Administrator (`adam@iacustomer.com`/ password).

9. Navigate to **Administration > Groups**.

10. Configure the group-role mapping for AD groups.

11. Set `spring.profiles.active=infoarchive.gateway.profile.AUTHENTICATION_ACTIVE_DIRECTORY` in the `application.yml` file (or `spring.profiles.active=infoarchive.gateway.profile.AUTHENTICATION_EXTERNAL_LDAP` if you use external LDAP).

12. Restart the IA Web App.

13. Start IA Shell.

14. Connect to server using AD credentials:

```
iashell> connect --rest-api http://localhost:8080/services --gateway
http://localhost:8080 --user adam@iacustomer.com --password ***
```

1.10.2 Configuring IA Shell with OTDS

IA Shell can authenticate a user with OTDS (OpenText Directory Services). For authentication, IA Shell uses Gateway that is bound inside the IA Web App. The way OTDS authentication works is the same as for LDAP and AD. The only items that should be configured for IA Shell and the connect command are the:

- Gateway secret
- User name
- User password

To learn how to obtain the Gateway secret for OTDS, see section 3.11 “Changing passwords and other secrets” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

1.11 Configuring refresh tokens for IA Shell

The method used to configure the time expiration for refresh tokens depends whether you are using OTDS or not:

If you are using OTDS

In OTDS Admin, view the OAuth2 client for the IA Shell and update the settings. For more information, see the *OpenText Directory Services Installation and Administration Guide*.

If you are not using OTDS

Update the settings in the `config/iawebapp/application-CLIENTS.yml` file:

```
  - accessTokenValiditySeconds: 1800
    authorities:
    - ROLE_TRUSTED_CLIENT
    authorizedGrantTypes:
    - password
    - refresh_token
    - implicit
    clientId: infoarchive.iawa
    refreshTokenValiditySeconds: 1860
    scopes:
    - search
    - compliance
    - administration
```

If you update these values, you must restart the Gateway.

Tip: If you are experiencing high latency, change the timeouts for access and refresh tokens, depending on the size of file and the amount of time it will take to upload file. Make the following changes:

1. In the `config/iashell/application.yml` file, update the `defaultSettings.other.idleTimeoutInMinutes` setting:

   ```
   defaultSettings:
     other:
       idleTimeoutInMinutes: 30
   ```

   When the IA Shell is idling for the time specified in `idleTimeoutInMinutes`, the system automatically disconnects. The IA Shell token is periodically refreshed instead of using a long-running token.

2. Add the following to the end of the `connection` section:

   ```
   connection:
     enableAutoConnect: false
   ```

   To enable auto-connect, set this value to `true` (by default, it is `false`). This allows the user to automatically connect when starting the IA Shell using the credentials defined in the `application.yml` file. If credentials are not defined, the system prompts you for the username, password, and secret.

1.12 Usability tips

- Use auto-completion for commands, command parameters and type aliases (use the TAB key). Auto-completion does not work for items that are not on the first page and is supported for the following commands:

  – `cd`

  – `ls`

  – `cat`

  – `ingest`

  To use auto-completion, specify the `--d` option and at least one character of the item you want:

  ```
  iashell> cd --d my-
  my-background-requests    my-orders-items
  ```

- To browse the commands history, use the ↑ and ↓ keys. The history only records commands that were successfully completed.
- You can pre-configure IA Shell by creating the start-up scripts.

Chapter 2

IA Shell commands

All the commands and their respective options are case sensitive.

IA Shell provides an ability to see the help for the existing commands with a list of supported options inside the command line. Use the `help` command for further information.

2.1 Commands context

The IA Shell supports setting a context for commands. Which commands are valid are based on the current state of the object. The context of the IA Shell is changed by the navigation commands.

The following is a list of existing IA Shell commands. Each command indicates whether it uses the IA Shell context.

For example, to invalidate a package, the following commands set the context for the command:

```
iashell> cd /tenants/INFOARCHIVE/applications/PhoneCalls/aips/e42a62e5-7449-4cca-8942
iashell>invalidate
```

Note that many commands also let you set the context by using the `--d` option.

2.2 Background operation commands

2.2.1 activate

Activate a job, which allows it to run. This command is only available if the job is inactive.

Option

- `[--d] rest-folder`: Command context path

Example 2-1:

```
iashell>cd /job-definitions/RefreshMetrics
iashell>activate
Activated job definition 'RefreshMetrics'
OK
```

2.2.2 download-log

Downloads single or composite execution log for job instances and order items.

Options

- `[--d Restfolder]`: Command context path. If no context is specified, the context is resolved from the current path context. Optional.
- `[--single-log or --s or -s Boolean]`: This option indicates to only download the single diagnostic log even if the composite log is available. If not specified, the default is to download the composite log if available. Optional default value: `false`.
- `[--to String]`: Specify a directory path. This option changes the directory where the downloaded file is created. Optional.
- `[--force or --overwrite Boolean]`: Force overwrite of existing file. If not set, and the file already exists, the name of the file will be made unique. Optional default value: `false`.

Example 2-2:

```
iashell> download-log
Start Download-log at [2024-12-10 11:40:01]
Requesting download of composite diagnostic execution log.
```

```
iashell> download-log --single-log
Start Download-log at [2024-12-10 11:40:01]
Requesting download of single diagnostic execution log.
```

If we re-run the following command a second time, the file is updated with a unique filename, such as `test(1).gzip` or `ArchiveAudits_061b64d0-6fca-4e1e-ba01-4999b35836d6(1).log.gz`:

```
iashell> download-log --single-log --to C:\Temp
Start Download-log at [2025-01-09 13:46:14]
Requesting download of single diagnostic execution log.
Downloaded C:\Temp\ArchiveAudits_061b64d0-6fca-4e1e-ba01-4999b35836d6(1).log.gz
OK
End Download-log at [2025-01-09 13:46:14]
Download-log took 0.99 seconds
```

```
iashell> download-log --to C:\Temp
Start Download-log at [2025-01-09 13:48:15]
Requesting download of composite diagnostic execution log.
Downloaded C:\Temp\ArchiveAudits061b64d0-6fca-4e1e-ba01-4999b35836d6_061b64d0-6fca-4e1e-
ba01-4999b35836d6(1).zip
OK
End Download-log at [2025-01-09 13:48:15]
Download-log took 0.31 seconds
```

2.2.3 inactivate

Inactivates a job, which prevents the job from running. Job instances will be skipped until the job is activated.

Option

- `[--d] rest-folder`: Command context path

Example 2-3:

```
iashell>cd /job-definitions/RefreshMetrics
iashell>inactivate
Inactivated job definition 'RefreshMetrics'
OK
```

2.2.4 job *** DEPRECATED

The command is deprecated. Use the other background commands listed in this section instead.

2.2.5 persist-log

Archives composite execution log for job-instances and order-items. Meant to avoid time out when downloading a composite log.

Options

- `[--d RestFolder]`: Command context path. If no context is specified, the context is resolved from the current path context. Optional.
- `[--check-log or --c or -c Boolean]`: This option checks that status of a previous persist command and is meant to determine when the download-log command can use the log prepared for download. Optional default value is `false`.

Example 2-4:

```
iashell> persist-log
Requesting preparation of composite diagnostic executive log for download.
Server accepted the request. Rerun this command with the check-log option to determine
when the log has been prepared for download.
```

```
iashell> persist-log --check-log
The composite log has been prepared for download.
```

```
iashell> persist-log --check-log
The composite log has not been prepared for download yet.
```

2.2.6 resume-schedule

Resumes the schedule of a job. This command is not available for jobs where the scheduling mode is manual or if the schedule has not been suspended.

Option

- `[--d RestFolder]`: Command context path

Example 2-5:

```
iashell>cd /job-definitions/RefreshMetrics
iashell>resume-schedule
Resumed job definition 'RefreshMetrics'
OK
```

2.2.7 run

Runs a job, scheduling it to run immediately. If the job is inactive, the job instance is skipped.

To find out the supported job properties, use the `cat` command to get the list first:

```
iashell> cd /job-definitions/GeneratePurgeCandidateList
iashell> cat
Object:
  version: 1
  name: GeneratePurgeCandidateList
  ...
  jobProperties:
    autoApprove:
      name: autoApprove
      allowedValues: [ ]
      description: If true, generated purge lists will be auto-approved
      label: Auto Approve
      maxValue: null
      minValue: null
      required: false
      type: BOOLEAN
    cancelApproved:
      name: cancelApproved
      allowedValues: [ ]
      description: If true, cancels approved purge list that were not disposed
      label: Cancel Approved
      maxValue: null
      minValue: null
      required: false
      type: BOOLEAN
    restrictPurgeListSize:
      name: restrictPurgeListSize
      allowedValues: [ ]
      description: If true, restricts the contents of the purge list to a single
partition key. To have less purge list generated set to false
      label: Restrict Purge List Size
      maxValue: null
      minValue: null
      required: false
      type: BOOLEAN
  lastModifiedBy: system
```

After this, pick the job property you want to set and run it:

```
iashell> run --properties cancelApproved=true
Scheduled 'GeneratePurgeCandidateList' to run, created 1 job instances
OK
```

Options

- `[--d RestFolder]`: Command context path
- `[--applications]` or `[--app String]`: Allows you to specify the application scope for a job. For the value, enter the name of an application or a list of names separated by commas. If the application cannot be found, the job runs for all valid applications.
- `[--properties]` or `[--p]` or [`--p String]`: Allows you to specify a list of property values for the job. The format is `<name>=<value>` and the list of names should be separated by commas.

Example 2-6:

```
iashell> cd /job-definitions/Post Ingest Processing
iashell> run --applications Audit, PhoneCalls --properties holdingName=Audit,
MaximumNumberOfLibraries=100
Scheduled 'Post Ingest Processing' to run, created 2 job instances
OK
```

2.2.8 run-job *** DEPRECATED

The command is deprecated. Use the `run` command instead.

2.2.9 start-schedule

Starts the schedule for a job definition. The command is not available for jobs where the scheduling mode is manual. If schedule has already started, the schedule will be reset and start on the new schedule.

Option

- `[--d RestFolder]`: Command context path

Example 2-7:

```
iashell>cd /job-definitions/RefreshMetrics
iashell>start-schedule
\Schedule has been started for 'RefreshMetrics', create 1 job instances
OK
```

2.2.10 stop-schedule

Stops and clears the schedule for a job definition. This command is not available for jobs where the scheduling mode is manual.

Option

- `[--d RestFolder]`: Command context path

Example 2-8:

```
iashell>cd /job-definitions/RefreshMetrics
iashell>stop-schedule
Schedule has been stopped for 'RefreshMetrics', deleted 1 scheduled job instances
OK
```

2.2.11 suspend-schedule

Suspends the schedule. This command is not available for jobs where the scheduling mode is manual. Use `resume-schedule` to remove the suspension. Does not prevent manual runs of the job. To prevent the job from running. use the `inactivate` command instead.

Option

- `[--d RestFolder]`: Command context path

Example 2-9:

```
iashell>cd /job-definitions/RefreshMetrics
iashell>suspend-schedule
Suspended job definition 'RefreshMetrics'
OK
```

2.2.12 wait-order-item

Waits for a background request to finish processing. Meant for scripting only and returns (either COMPLETE, EXCEPTION or CANCELED). If a time-out is specified, may return other states, otherwise, IA Shell does not respond until the background request is done.

Options

- `[--d]`: Command context path
- `[--timeout] or [--t integer]`: Maximum time to wait result in seconds. Missing, negative or zero means infinite waiting. Optional, default = <none>

Example 2-10:

```
iashell>cd /tenant/my-background-requests/Baseball_DeleteData_2022-08-31T11:00:56-04:00
iashell>wait-order-item --timeout 30
COMPLETE
```

2.3 Built-in commands

2.3.1 clear

Clears the IA Shell console.

There are no arguments or options for this command.

2.3.2 exit

Exits the IA Shell. Automatically disconnects if the `disconnect` command was not done.

There are no arguments or options for this command.

2.3.3 help

Provides a list of the existing IA Shell commands with a short description. It also provides information about supported options for the selected command if passing the command name as parameter.

Option

- `[--command]`: Command name to provide help for.

Example 2-11:

```
iashell> help import
```

Example

2.3.4 history

Displays or saves the history of previously run commands.

Option

- `[--file]`: A file to save history to.

2.3.5 script

Runs the IA Shell non-interactive shell commands in the specified file unless a command fails, which stops processing.

Example 2-12:

```
iashell> script C:\temp\input.iashell
```

Note: If the command requires input, the shell prompts for any missing information before moving to the next command in the file.

The IA sample applications provide examples of the file format in the `install.iashell` script that show how to configure and ingest data for both SIP and table applications.

2.3.6 stacktrace

Displays the full stacktrace of the last error.

2.4 Cache control commands

2.4.1 cache-in

Creates a background request to cache in a resource. Action is applicable for an AIP resource, PG/Lucene Library or table application.

Options

- `[--d]`: Command context path. If no context is specified, the context is resolved from the current path context.
- `[--backup]`: When this parameter is entered, additional backup of table application data is taken. This parameter is ignored for all targets except a table application.

Example 2-13:

```
iashell> cache-in /tenant/applications/PhoneCalls/aips/
e42a62e5-7449-4cca-8942-15cf17ea3c77
Resource has been requested to be cached-in in background processes. Order Item:
CacheIn_Aipe42a62e5-7449-4cca-8942-15cf17ea3c77_2018-12- 10T11:22:05.578+03:00
OK
```

2.4.2 cache-out

Creates a background request for a resource to be cached-out. Action is applicable for an AIP resource, PG/Lucene Library or table application.

Options

- `[--d]`: Command context path. If no context is specified, the context is resolved from the current path context.
- `[--backup]`: When this parameter is entered, additional backup of table application data is taken. This parameter is ignored for all targets except a table application.

Example 2-14:

```
iashell> cache-out /tenant/applications/PhoneCalls/aips/
e42a62e5-7449-4cca-8942-15cf17ea3c77
Resource has been requested to be cached-out in background processes.
Order Item:
CacheOut_Aipe42a62e5-7449-4cca-8942-15cf17ea3c77_2018-12-10T11:22:40.901+03:00
OK
```

2.5 Commands specific to cryptography

2.5.1 set-as-default

Sets the context representing a crypto object to be the default.

2.5.2 set-mek-alias

This command tells OpenText Information Archive to use a different MEK alias used by the Key Mediator and creates a background request. The command is only available if the Key Mediator is enabled on the IA Server.

Option

- `[--mekAlias string]`: Provide the new MEK alias to use with the Key Mediator (Mandatory).

Example 2-15:

```
iashell> set-mek-alias 20926061-1411-4f5d-9309-547beefbf341
         Request was accepted and will be processed as a background request.
```

The MEK alias is provided by the Key Mediator, and this command tells OpenText Information Archive to use the new key. IA Shell, however, has no way of validating the provided MEK alias and, consequently, the background request will fail.

It is possible with existing IA Shell commands to review your background requests:

```
iashell> cd /tenant
iashell> cd my-background-requests
iashell> ls
+------------------------------------+---------------------------------------+
|id                                  |name                                   |
+------------------------------------+---------------------------------------+
|a2b3df9f-f792-454b-8edf-837a62195c4f|Set Mek Alias_2023-01-16T15:31:42-05:00|
|                                    |                                       |
+------------------------------------+---------------------------------------+
```

The following is an example when the MEK alias is not valid:

```
No access to Key Mediator. Either OpenText Information Archive is not configured for Key
Mediator or you do not have permission for this command.
Details of the error have been omitted. You can use the stacktrace command to print the
full stacktrace.
```

2.6 Command specific to Lucene

The command can be used for SIP structured data and search result.

2.6.1 describe

Describe the associated Lucene field information.

Option

- `[--d rest-folder]`: Command context path. If no context is specified, the context is resolved from the current path context.

  Optional default value:]

Example 2-16:

```
iashell> describe
+---------------------+---------+------------------------------------------
+------------------------------------------+---------+
|name                 |termCount|minTerm                                   |
maxTerm                                   |type     |
+---------------------+---------+------------------------------------------
+------------------------------------------+---------+
|#SentToArchiveDate   |1        |1038700800000                             |
1038700800000                             |STRING   |
|#CallStartDate       |3        |1037462262104                             |
1038681514104                             |STRING   |
|#CallEndDate         |3        |1037463997104                             |
1038683629104                             |STRING   |
|#CallFromPhoneNumber |3        |1320476452                                |
892778754                                 |STRING   |
|#CallToPhoneNumber   |3        |1008136841                                |
33303518                                  |STRING   |
|#CustomerID          |3        |388                                       |
68                                        |STRING   |
|#CustomerLastName    |3        |Singh                                     |
Turner                                    |STRING   |
|#CustomerFirstName_ft|3        |christian                                 |
mia                                       |STRING   |
|#CustomerFirstName   |3        |Christian                                 |
Mia                                       |STRING   |
|#RepresentativeID    |3        |16                                        |
3                                         |STRING   |
|#AttachmentName      |1        |recording                                 |
recording                                 |STRING   |
|#FileName            |3        |recording12.mp3                           |
recording32.mp3                           |STRING   |
|#CreatedBy           |1        |PhoneRecorder                             |
PhoneRecorder                             |STRING   |
|#CreatedOnDate       |3        |1037463997104                             |
1038683629104                             |STRING   |
|ID                   |3        |afac22ad-3d2a-4f87-83da-e5d39d794392:aiu:1|
afac22ad-3d2a-4f87-83da-e5d39d794392:aiu:3|STRING   |
|rank                 |3        |0                                         |
2                                         |STRING   |
|aiu                  |1        |1                                         |
1                                         |STRING   |
|charCount            |3        |169                                       |
177                                       |STRING   |
|object               |0        |null                                      |
null                                      |BYTEARRAY|
|cids                 |3        |afac22ad-3d2a-4f87-83da-e5d39d794392:ci:1 |
afac22ad-3d2a-4f87-83da-e5d39d794392:ci:3 |STRING   |
|                     |         |
|                                          |         |
+---------------------+---------+-------------------------------------------
+------------------------------------------+---------+
```

2.7 Commands specific to SIP applications

In this section, the exceptional command is the `ingest` command, which is used for both SIP- and table-based applications.

2.7.1 apply-library-policy

The command launches the load balancing as a background process after a change in the library policy configuration.

Option

- `[--d]`: Command context path.

2.7.2 backup-resource

Requests that a resource to be backed up. Supported resources are AIP, holding, or database for a table application.

Option

- `[--d RestFolder]`: Command context path. If no context is specified, the context is resolved from the current path context.

Example 2-17:

```
iashell> cd /tenants/INFOARCHIVE/applications/PhoneCalls/aips/75d8b4f4-d0c6-4cac-b2ea-
d7f8f9b34fea
iashell> backup-resource
Request has been sent to backup the resource.
Order Item: PhoneCalls-CC-1000010-1_PhoneCalls_2025-02-11T10:57:13-05:00
OK
```

Example 2-18:

```
iashell> cd /tenants/INFOARCHIVE/applications/Audit/holdings/Audit
iashell> backup-resource
Request has been sent to backup the resource.
Order Item: Audit_2025-02-26T16:35:50-05:00
OK
```

Example 2-19:

```
iashell> cd /tenants/INFOARCHIVE/applications/Tickets/databases/Tickets-sql-db
iashell> backup-resource
Backup has been processed successfully.
Name: Tickets-sql-db
OK
```

2.7.3 close-library

Sends a request to close the library. Actual closing will be performed the next time the Close job is run.

Options

- `[--d]`: Command context path.
- DEPRECATED: `[--id --last]`: An ID value (self link) of a library object or its index in the result of the last `select` command performed.
- DEPRECATED: `[--rel]`: The relation to an PG/Lucene object.

Argument: Command context path

Example 2-20:

```
iashell> close-library --last
```

2.7.4 enumerate

Enumerates a list of AIPs that are ready to be ingested into a certain ingest-node object. Produces the ID indexes (if some results are found) to be stored in the context.

List of AIPs are truncated to `max` of Ingest Node configuration (configuration parameter `enumerationMaxResultCount`, default value is `10`) unless `max` parameter is provided.

Options

- `[--allow-background-request]`: If specified, and the metadata database is not available during the ingestion, the ingestion is falling to an asynchronous ingestion. The parameter considered only when the `--ingest` option is specified.

  Optional default value: `false`
- `[--d]`: Command context path
- `[--list]`: If specified, outputs AIP self link(s) to a list, which includes the AIP name and external ID.

  Default value: `false`
- `[--cutoff]`: Cutoff period in days.
- `[--max]`: Maximum number of AIPs returned.
- `[--minusRunning]`: Indicates that the `max` parameter should be decreased for the AIPs that are currently being ingested.

  Default value: `false`
- `[--ingest]`: Specifies that you need to ingest the results of the enumeration.

Default value: `false`

- `[--parallel-threads]`: Specifies the number of threads to be used during ingestion in parallel mode. This option can only be used within the `--ingest` option.
- `[--async]`: If specified, ingestion is performed asynchronously. The option is taken into account only with the `--ingest` option. Otherwise, it is ignored.

  Optional default value: `false`.
- DEPRECATED: `[--id --last]`: An ID value (self link) of an ingest-node object or its index in the result of the last `select` command performed.

Example 2-21:

```
iashell> enumerate
+------------------------------------+-----------------------+
|externalId                          |name                   |
+------------------------------------+-----------------------+
|55980034-731f-43a2-8b1d-e12ebf06f5da|PhoneCalls-CC-1000010-1|
|8d10fac6-0489-4514-9be0-56f1376e9fca|PhoneCalls-CC-1000009-1|
|1be3435a-970e-4c5c-acb4-a6c48388ad70|PhoneCalls-CC-1000008-1|
|865f3160-6058-422b-80ca-926bb0ac004e|PhoneCalls-CC-1000007-1|
|734ce5c2-1143-46ed-a386-373a3d591f3b|PhoneCalls-CC-1000006-1|
|d1e0a260-f766-42cd-be47-920c262ed6d9|PhoneCalls-CC-1000005-1|
|ce39e753-6176-47a4-a604-049511df3e26|PhoneCalls-CC-1000004-1|
|4d62e111-6957-4735-8cef-f821955a35f2|PhoneCalls-CC-1000003-1|
|0f861fa7-5372-4955-a071-c44291c74e5b|PhoneCalls-CC-1000001-1|
|a5817da1-c3bb-4afe-b9af-c1415b74ec56|PhoneCalls-CC-1000000-1|
|                                    |                       |
+------------------------------------+-----------------------+
```

```
iashell> enumerate --ingest
Ingested http://localhost:8765/systemdata/applications/
c7d3c9a0-92d5-4f1c-9278-9b6c09c8384d/aips/55980034-731f-43a2-8b1d-e12ebf06f5da/ingest
Ingested http://localhost:8765/systemdata/applications/
c7d3c9a0-92d5-4f1c-9278-9b6c09c8384d/aips/8d10fac6-0489-4514-9be0-56f1376e9fca/ingest
Ingested http://localhost:8765/systemdata/applications/
c7d3c9a0-92d5-4f1c-9278-9b6c09c8384d/aips/1be3435a-970e-4c5c-acb4-a6c48388ad70/ingest
Ingested http://localhost:8765/systemdata/applications/
c7d3c9a0-92d5-4f1c-9278-9b6c09c8384d/aips/865f3160-6058-422b-80ca-926bb0ac004e/ingest
Ingested http://localhost:8765/systemdata/applications/
c7d3c9a0-92d5-4f1c-9278-9b6c09c8384d/aips/734ce5c2-1143-46ed-a386-373a3d591f3b/ingest
Ingested http://localhost:8765/systemdata/applications/
c7d3c9a0-92d5-4f1c-9278-9b6c09c8384d/aips/d1e0a260-f766-42cd-be47-920c262ed6d9/ingest
Ingested http://localhost:8765/systemdata/applications/
c7d3c9a0-92d5-4f1c-9278-9b6c09c8384d/aips/ce39e753-6176-47a4-a604-049511df3e26/ingest
Ingested http://localhost:8765/systemdata/applications/
c7d3c9a0-92d5-4f1c-9278-9b6c09c8384d/aips/4d62e111-6957-4735-8cef-f821955a35f2/ingest
Ingested http://localhost:8765/systemdata/applications/
c7d3c9a0-92d5-4f1c-9278-9b6c09c8384d/aips/0f861fa7-5372-4955-a071-c44291c74e5b/ingest
Ingested http://localhost:8765/systemdata/applications/
c7d3c9a0-92d5-4f1c-9278-9b6c09c8384d/aips/a5817da1-c3bb-4afe-b9af-c1415b74ec56/ingest
OK
iashell>
```

```
iashell> enumerate --ingest --async
```

```
iashell> enumerate --max 10 --cutoff 7 --minusRunning
```

```
iashell> enumerate applications/PhoneCalls/ingest--nodes/ingest_node_01
```

```
iashell> enumerate applications/PhoneCalls/ingest--nodes/ingest_node_01 --max 10 --
cutoff 7 --minusRunning
```

```
iashell> enumerate --max 10 --ingest
```

2.7.5 ingest

Allows for the ingestion of data into OpenText Information Archive. It is used for both SIP and table application data ingestion:

- For SIP applications, the command can be run from either the application, AIPs, or a particular AIP context.
- For table applications, the command can be run from either the schema context or table context.

If there is no argument specified, the system tries to find the ingestion link in the current path context and uses it for the ingestion.

For table data files, there is no limitation on the file name. The schema and table names are taken from the file content.

Note: If you are running the `ingest` command for a table application, and there is a reference to unstructured data, only files that are collocated with the application, by default, will be allowed.

Example 2-22: Example of properly collocated data that can be successfully ingested:

```
<?xml version="1.0" encoding="UTF-8"?>
<TICKETS>
  <TICKET_ATTACHMENT>
    <ROW>
      <TICKET_NUMBER>10011553</TICKET_NUMBER>
      <ATTACHMENT ref="../blobs/ticket1.jpg" />
      <MIMETYPE>image/jpeg</MIMETYPE>
    </ROW>
    <ROW>
      <TICKET_NUMBER>10018132</TICKET_NUMBER>
      <ATTACHMENT ref="../blobs/ticket2.jpg" />
      <MIMETYPE>image/jpeg</MIMETYPE>
    </ROW>
    <ROW>
      <TICKET_NUMBER>10018132</TICKET_NUMBER>
      <ATTACHMENT ref="../blobs/ticket3.jpg" />
      <MIMETYPE>image/jpeg</MIMETYPE>
    </ROW>
  </TICKET_ATTACHMENT>
</TICKETS>
```

Example 2-23: Example of data that is not collocated:

```
<?xml version="1.0" encoding="UTF-8"?>
<TICKETS>
  <TICKET_ATTACHMENT>
    <ROW>
      <TICKET_NUMBER>10011553</TICKET_NUMBER>
      <ATTACHMENT ref="c:/<FOLDER_NAME>/blobs/ticket1.jpg" />
```

```
        <MIMETYPE>image/jpeg</MIMETYPE>
    </ROW>
```

If you want the system to trust a different file location (for example `c:/<FOLDER_NAME>/blobs/ticket1.jpg` in Example 2–23), this can be defined by the `defaultSettings.table.allowedUnstructuredDataRoot` property in the IA Shell’s `application.yml` file. This property refers to the root directory for referenced files containing unstructured data for table applications.

So, for Example 2–23, the `application.yml` file would be updated with the following:

```
defaultSettings:
  other:
    uploadGatewaySocketTimeout: 6000000
  pathToDeclarativeConfigurationDefaultProperties: ../config/iashell/default.properties
  pathToDownloadDirectory: _downloads
  sip:
    followAsyncIngestionSleepInSeconds: 10
    pathToErrorFolder: _error
    pathToSuccessFolder: _success
    sipFormat: sip_zip
    sipIngestThreads: 1
  table:
    allowedUnstructuredDataRoot: C:/<FOLDER_NAME>/blobs
```

The user scenarios for the ingest command can be found in the following sections:

- Ingesting a SIP package
- Ingesting table data
- Ingesting SIPs asynchronously

Options

- `[--all-ids]`: Specifies to use all the last IDs from the context (last enumerate or select command). For SIP applications only.
- `[--allow-background-request]`: If specified, and the metadata database is not available during the ingestion, the ingestion becomes an asynchronous ingestion. Has no effect if the `--from` option refers to a file (for example, `--recursive`).

  Optional default value: `false`.
- `[--async]`: If specified, ingestion is performed asynchronously. Use this option along with `--follow` to wait for the background processing to complete.

  Optional default value: `false`.
- `[--d]`: Command context path
- `[--deleteOnError]`: The parameter specifies if to delete SIP package after unsuccessful ingestion. If not used, than SIP package is kept in place.

  Optional default value: `false`.
- `[--deleteOnSuccess]`: The parameter specifies if to delete SIP package after successful ingestion. If not used, than SIP package is kept in place.

Optional default value: `false`.

- `[--detailed]`: If specified, a message is shown for each AIP when processing has finished. If not set, reduce the amount of intermediate reporting.

  Optional default value: `false`.

- `[--filter]`: Filter to be included during file search. All the files that match the filter will be included in the search results. Would have no effect if the `--from` option refers to a file. If no filters are specified, the filter is set to a default value from the configuration. This is like a regular expression filter (for example: `--filter = ".*\.xml"`).

- `[--follow]`: If specified, the command will wait for the order items to complete and report. Only supported for SIP applications and must be used in conjunction with the `--async` option; otherwise, if the`--async` option is specified, as soon as the request to ingest the AIPs has been accepted by the server, the command ends and it is up to the user to check when the background requests have finished.

  Optional default value: `false`.

- `[--from]`: Path to data, which can refer to either SIP(s) or table(s). It can be one or more files delimited by `;` or `:` (the system-dependent path-separator character), or the path to a folder within data files. Simple file filtering expressions, such as `/folder/*.zip` are also allowed. This filtering is based on the “glob” syntax. Both types of paths are supported: absolute and relative ones.

- `[--format]`: The SIP file format. The default format should be configured as `sip_zip` in the `application.yml` file. The alternative format is `eas_sip_zip`, which should be used for ingesting legacy EAS 3.x SIP packages.

- [`--id` or `--last`: An ID value (self link) of an object to be ingested or its index in the result of the last select command performed. For SIP applications only.

- `[--moveOnError]`: The parameter specifies if to move SIP package after unsuccessful ingestion. Target directory name is taken from the `--moveOnErrorTo` parameter or from {{application properties}} file.

- `[--moveOnErrorTo]`: The parameter specifies target directory to move SIP package after unsuccessful ingestion. If the parameter is used, then the `--moveOnError` parameter is automatically considered to be `true`.

- `[--moveOnSuccess]`: The parameter specifies if to move SIP package after successful ingestion. Target directory name is taken from the `--moveOnSuccessTo` parameter or from {{application properties}} file.

  Optional default value: `false`.

- `[--moveOnSuccessTo]`: The parameter specifies target directory to move SIP package after successful ingestion. If the parameter is used, then `--moveOnSuccess` parameter is automatically considered to be `true`.

  Optional default value: `false`.

- `[--parallel-threads]`: Specifies the number of parallel threads for ingestion. The default values for SIP and table applications are different and specified in the `application.yml` file.

- `[--recursive]`: If specified, recursively searches for all the files that match the filter depth-first within the specified path. Would have no effect if the `--from` option refers to a file (for example: `--recursive`).
- DEPRECATED: `[--id --last]`: An ID value (self link) of an object to be ingested or its index in the result of the last select command performed.

Example 2-24:

```
iashell> ingest/tenants/INFOARCHIVE/applications/PhoneCalls/aips/3516cc63-658c-4914-
b90e-9724ee824e99 --async
```

```
iashell> ingest --from /samples/sip01.zip
```

```
iashell> ingest applications/PhoneCalls --from "C:/Demo/samples/sip01.zip"
```

```
iashell> ingest --from "C:/Demo/SIPs" --moveOnSuccess --moveOnSuccessTo "C:/Demo/SIPs/
OK_SIPs" --moveOnErrorTo "C:/Demo/SIPs/SIPs
```

```
iashell> ingest --from "C:/Demo/PhoneCalls/data" --allow-background-request
```

```
ingest --d /tenants/INFOARCHIVE/applications/Baseball/databases/Baseball-sql-db/schemas/
BASEBALL --from "C:/Demo/data/BASEBALL"
```

```
iashell> cd /tenants/INFOARCHIVE/applications/PhoneCalls
iashell> ingest --from data --async --follow --detailed
```

Note: If configured, ingestion includes the extraction and indexing of text from unstructured content contained in a SIP. This text extraction does not happen within the IA Sever's JVM, but in separate processes called `kvoop`. We typically get two per attached piece of unstructured content for text extraction. These processes, albeit briefly, take some memory (typically up to 170 MB per pair), and if they are run in parallel, this can theoretically add up to problematic amounts. If you are experiencing memory issues coming from text extraction on behalf of SIP ingestion, you can reduce the number of concurrent ingest calls, or run ingest commands with a reduced value for the parallel-threads parameter (see above).

2.7.6 invalidate

Invalidates an AIP object.

Options

- `[--d]`: Command context path
- `[--category]`: Specify a category, if required.
- `[--description]`: Specify description, if required.
- DEPRECATED: `[--id --last]`: An ID value (self link) of an AIP object or its index in the result of the last `select` command performed.

Argument: Command context path

Example 2-25:

The following example illustrates the cd navigation approach:

```
iashell> pwd
/tenants/INFOARCHIVE/applications/PhoneCalls/aips/e42a62e5-7449-4cca-8942-15cf17ea3c77
iashell> invalidate
Invalidated PhoneCalls-CC-1000000-1
OK
```

2.7.6.1 Invalidating an AIP

Run the following commands to invalidate an AIP and use the output of the `ls` command to determine the external ID for the AIP:

```
iashell> connect
iashell> cd applications/<APPLICATION_NAME>/aips
iashell> ls
iashell> cd <aip_id>
iashell> invalidate
```

2.7.7 mark-as-error

To facilitate the support, the command allows an easy way to mark an AIP as an error. In some circumstances, an AIP can be blocked during the ingesting phase without an error code. For example, this situation can occur if the server is shutdown during ingestion or if an unexpected error is raised and not caught correctly.

The command should be run from the AIP path context or the AIP path should be passed with the `--d` parameter.

Options

- `[--d]`: Command context path
- `[--error-message string]`: Specify the reason why AIP package is marked as an error.

Example 2-26:

```
iashell> mark-as-error --d 19a3cb5a-3be6-465c-a7b1-ad7045fe2533 --error-message "test
message"
```

```
iashell> mark-as-error --d /tenants/INFOARCHIVE/applications/PhoneCalls/aips/069a45b1-
b257-4bdc-b2a9-e68b6d4516b4
```

2.7.8 post-ingest

Generates a background request that does post-ingestion steps for a library associated with a package. Running the job updates the library and partition keys, and sets the cache lock date.

Option

- `[--d rest-folder]`: Command context path.

Example 2-27:

```
iashell> cd /tenants/INFOARCHIVE/applications/Certificates/libraries/d5055be4-faba-4ad2-
b3ae-a1e1c9ce58ef
iashell> post-ingest
         Post ingest of library d5055be4-faba-4ad2-b3ae-a1e1c9ce58ef has been submitted
as a background task.
         OK
```

2.7.9 rebuild-library

Creates a background request to rebuild a Lucene index library from the PDI XML file of the corresponding AIPs. This command is applicable for a library resource.

Important
Use this command with caution. Prior to running the command, review the *Rebuilding a library* section in the *OpenText Information Archive Administration Guide*.

Option

- `[--overwrite-backup]`: If specified, the library backup generated by the rebuild overwrites the original backup. If not specified, the system keeps the original backup as an archived format while the new backup takes its place for the backup format.

Example 2-28:

```
iashell> cd applications/PhoneCalls
iashell> cd libraries
iashell> cat
+------------------------------------+--------+--------+------+--------+
|name                                |aipCount|aiuCount|closed|detached|
+------------------------------------+--------+--------+------+--------+
|053849e1-925c-4af6-bf06-dda781b1f23c|1       |3       |true  |false   |
|06156b4b-0d7b-42c8-9c0e-5008ac75b197|1       |3       |true  |false   |
|08ddff12-2655-4d34-9c55-18f35cc8b8b1|1       |3       |true  |false   |
|1d3dbd3a-3e9b-4091-b34c-8277dc7f06f2|1       |10      |true  |true    |
|202e56a8-0e05-41fd-9d22-43a1224b7bfe|1       |10      |true  |false   |
|421070a1-1130-4168-95a8-a61be7495b8a|1       |3       |true  |false   |
|6974e58e-4115-41fd-b05d-9c0fa3bfec26|1       |3       |true  |false   |
|82212ee3-3f6c-4147-ad84-8bf0b4837834|1       |3       |true  |false   |
|a3dce469-ceb0-40d3-abcf-428b5fea3178|1       |3       |true  |false   |
|c9ff6ca3-61c7-4264-a437-062013e689ea|1       |4       |true  |false   |
|                                    |        |        |      |        |
|---------------------------------    |        |        |      |        |
|Elements count: 12. Page 1 of 2      |        |        |      |        |
|next ->                              |        |        |      |        |
+------------------------------------+--------+--------+------+--------+
iashell> cd 053849e1-925c-4af6-bf06-dda781b1f23c
iashell> rebuild-library
Rebuild of library been submitted as background process.
Order Item: 053849e1-925c-4af6-bf06-dda781b1f23c_2025-02-13T17:25:08+01:00
OK
```

2.7.10 rebuild-sip

IA Server rebuilds a SIP from the PG/Lucene database data or SIP/PDI content objects. This operation may require AIP cache-in for granular disposition scenarios.

Option

- `[--d]`: Command context path. If no context is specified, the context is resolved from the current path context.

Example 2-29: Recover AIP (applicable for AIP and PG/Lucene Library):

```
iashell> rebuild-sip /tenants/INFOARCHIVE/applications/PhoneCalls/aips/
e42a62e5-7449-4cca-8942-15cf17ea3c77
Package rebuild has been submitted as background process.
Order Item:RebuildSip_Aipe42a62e5-7449-4cca-8942-15cf17ea3c77_2018-12-
10T12:16:50.152+03:00
OK
```

2.7.11 receive

Receives (upload) a SIP package or packages to the server.

After the successful run of the command, the context (path) should be switched to the AIP.

Options

- `[--allow-background-request]`: If specified, and the metadata database is not available during the ingestion, the ingestion is falling to an asynchronous ingestion. The parameter is taken into account only when the `--ingest-direct` option is specified.

  Optional default value: `false`
- `[--d]`: Command context path
- `[--format]`: The SIP file format.

  The default format should be configured as `sip_zip` in the `<INFOARCHIVE_ROOT>/config/iashell/application.yml` file.
- `[--list]`: If specified, outputs the AIP self link(s) of the SIP(s) received to a list (comma-separated).

Default value: `false`

- `[--from]`: Path(s) to one or more SIP files delimited by `;` or `:` (the system-dependent path-separator character) or a directory path where the SIP files are located. Simple file filtering expressions, such as `/folder/*.zip` are also allowed. This filtering is based on the “glob” syntax.
- `[--recursive]`: If specified, recursively searches for all the files that match the filter depth-first within the specified path. Would have no effect if the `[--from]` option refers to a file (for example: `--recursive`).
- `[--ingest-direct]`: If specified, it will be used to direct ingestion.

  Default value: `false`
- `[--parallel-threads]`: Specifies the number of threads to be used during ingestion in parallel mode. This option can only be used within the `ingest-direct option` together.
- `[--moveOnSuccess]`: The parameter specifies whether to move a SIP package after successful ingestion. Target directory name is taken from `[--moveOnSuccessTo]` parameter or from the `application.yml` file.
- `[--moveOnSuccessTo]`: The parameter specifies target directory to move a SIP package after successful ingestion. If the parameter is used, then the `[--moveOnSuccess]` parameter is automatically considered to be 'true'.
- `[--moveOnError]`: The parameter specifies whether to move SIP package after unsuccessful ingestion. The target directory name is taken from the `[--moveOnErrorTo]` parameter or from the `application.yml` file.
- `[-moveOnErrorTo-]`: The parameter specifies target directory to move a SIP package after unsuccessful ingestion. If the parameter is used, then the `[--moveOnError]` parameter is automatically considered to be 'true’.
- `[--deleteOnSuccess]`: The parameter specifies whether to delete a SIP package after successful ingestion. If not used, than the SIP package is kept in place.
- `[--deleteOnError]`: The parameter specifies whether to delete a SIP package after unsuccessful ingestion. If not used, than the SIP package is kept in place.

Example 2-30:

```
iashell> receive --format sip_zip --from "C:/Demo/SampeSipsDir"
```

The following is how to use the `receive` command with the direct ingestion option:

```
iashell> receive --from "C:/Demo/mySip.zip" --direct
```

2.7.12 recover-resource

Recovers a resource from an existing backup. Supported resources are AIP and holding.

Options

- `[--d RestFolder]`: Command context path. If no context is specified, the context is resolved from the current path context.

Example 2-31:

```
iashell> cd /tenants/INFOARCHIVE/applications/PhoneCalls/aips/75d8b4f4-d0c6-4cac-b2ea-
d7f8f9b34fea
iashell> recover-resource
Request has been sent to recover the resource.
Order Item: PhoneCalls-CC-1000010-1_PhoneCalls_2025-02-11T10:57:13-05:00
OK
```

Example 2-32:

```
iashell> cd /tenants/INFOARCHIVE/applications/Audit/holdings/Audit
iashell> recover-resource
Request has been sent to recover the resource.
Order Item: Audit_2025-02-26T16:35:50-05:00
OK
```

2.7.13 reject

Rejects an AIP object.

Options

- `[--d]`: Command context path
- `[--category]`: Specify category, if required.
- `[--description]`: Specify description, if required.
- DEPRECATED: `[--id --last]`: An ID value (self link) of an AIP object or its index in the result of the last `select` command performed.

2.7.14 rollback

Allows you to undo `invalidate` or `reject` actions performed on an AIP resource. The command should be run from the AIP path context or the AIP path should be passed with the `--d` parameter.

Options

- `[--d]`: Command context path. If no context is specified, the context is resolved from the current path context.

Example 2-33:

```
iashell> cd /tenants/INFOARCHIVE/applications/CertificatesAndTrades/aips/847c1b5d-
ce02-4edf-9a56-d3d6e278b70f
iashell> rollback
AIP package 'Certificates-CertApp-2012-11-30_001-1' has been rolled back.
OK
```

Example 2-34:

```
iashell> cd /tenants/INFOARCHIVE/applications/CertificatesAndTrades/aips
iashell> rollback --d 847c1b5d-ce02-4edf-9a56-d3d6e278b70f
AIP package 'Certificates-CertApp-2012-11-30_001-1' has been rolled back.
OK
```

2.8 Commands specific to table applications

2.8.1 backup-database

Backups the database for a table application. The command can be run from the database path context or the path to the required database can be passed with the `-d` option.

Option

- `[--d RestFolder]`: Command context path. If no context is specified, the context is resolved from the current path context.

Example 2-35:

```
iashell> cd applications/Baseball/databases/Baseball-sql-db
iashell> backup-database
Request has been sent to backup the database.
Please track progress using cat /tenant/order-items/58ac23cc-3948-4f3c-85c3-2698c1262a70.
OK
```

2.8.2 chain-of-custody

Runs chain-of-custody tests against tables. If the file parameter points to a file within the schema tests, it is assumed that the table test definition file is located in the same folder and its name is defined by the `custodyTableFileName` configuration setting.

Also, it appends the report generated to the `chain-of-custody.log` file in the current folder.

In a default installation, the `chain-of-custody` command is skipped when it is part of an IA Shell script. To enable it in the context of running an IA Shell script, set the `enableChainOfCustodyInBatch` property to `true` in the `<INFOARCHIVE_ROOT>/config/iashell/application.yml` file.

Options

- `[--d]`: Command context path
- `[--file]`: Path to the file that contains the test definition.

  If omitted, the path is taken from the configuration in the `<INFOARCHIVE_ROOT>/config/iashell/application.yml` file.

  There are properties named `custodySchemaFileName` and `custodyTableFileName`.
- `[--table]`: Optional table name to run the tests on, If omitted, all tables of the schema are checked.
- `[--detailed]`: If the option is set, the command produces detailed output for the report.

  Default value: `false`

Example 2-36:

```
iashell> chain-of-custody schemas/TICKETS --file tools/chain-of-custody-schema.xml
```

```
iashell> chain-of-custody schemas/TICKETS
```

```
iashell> chain-of-custody applications/Tickets/databases/Tickets-sql-db/schemas/TICKETS
--file
```

Example for a single table:

```
iashell> cd applications/Tickets/databases/Tickets-sql-db/schemas/TICKETS
```

```
iashell> chain-of-custody --table TICKET_ATTACHMENT --file tools/chain-of-custody-
schema.xml
```

2.8.2.1 chain-of-custody test file format

The `chain-of-custody` command sends a HTTP POST request to the IA Server to run chain of custody tests. These tests are specified in the `file` parameter. The specified tests are run on a table discovered by the REST template using the provided values contained in the command's parameters.

For more information, refer to `chain-of-custody` command.

The following is an illustration of the input file whose path is specified in the `file` parameter:

```
<chainofcustody>
  <testsuite>
    <test class="com.emc.ia.coc.tests.TableDataTest" />
  </testsuite>
</chainofcustody>
```

The above input will run `TableDataTest` on the table name specified in the `table` parameter.

2.8.2.2 Running chain-of-custody tests for table archives

Chain of custody tests check the integrity of the ingested tables in OpenText Information Archive. They are configurable with respect to which tests users want to run. They can be configured pre- or post-data ingestion via an XML file. Refer to the following files in the `<INFOARCHIVE_ROOT>/config/iashell` directory to review two examples of such files:

- `chain-of-custody-table.xml`
- `chain-of-custody-schema.xml`

Users can POST the XML file containing the tests’ configuration on the REST link contained in the response of the retrieved table (desired table on which these tests need to run against). Refer to the Developer's REST Guide for further information. This will return the result back to the users. The tests rely on the information given in the `metadata.xml` of the tables provided by the users. If any discrepancy is found between the aforementioned metadata and the ingested tables after running the chain of custody tests, OpenText Information Archive reports it to the users.

The following table describes the tests that can be performed for PostgreSQL deployments:

Test Class	Test Case	Description
TableRowCountTest	testRdbTableCount	Tests that the count of tables in the RDB matches the table count specified in the metadata.

Test Class	Test Case	Description
TableRowCountTest	testRdbTableRowCount	Tests that the row count in each RDB table matches the row count listed in the table metadata.
TableRowCountTest	testMetaDataRowCountTest	For a RDB table with no empty rows, this tests whether its corresponding metadata row count is also not empty.
TableRowTest	testRdbTableRow	Tests to ensure at least one row exists for a table that is present in the RDB.
TableDataTest	testRdbTableData	For an RDB table, tests whether the metadata columns count is empty.
MetadataTest	testRdbMetadata	Tests the table present in metadata is actually present in the RDB.
ColumnDataTest	testMetadataContainsColumnType	For a metadata table, tests whether all the columns of the table have an associated type information.
ColumnDataTest	testMatchColumnCountsAndNames	For a table, tests whether the column count and column names match in both RDB and metadata.
ColumnDataTest	testMatchColumnTypes	For a table, tests whether column type in metadata matches to the corresponding column type in RDB.
SpacesTest	testRdbLeadingSpaces	For a table, tests for the presence of leading spaces in each column of data in RDB. The number of leading spaces and rows to test can be configured in the chain-of-custody-table.xml file.
SpacesTest	testRdbTrailingSpaces	For a table, tests for the presence of trailing spaces in each column of data in RDB. The number of trailing spaces and rows to test can be configured in the chain-of-custody-table.xml file.

Test Class	Test Case	Description
SchemaMetadataTest	testRdbDataInMetadata	Tests that each table present in the RDB is listed in the metadata.
SchemaMetadataTest	testMetadataInRdbData	Tests that each table present in the metadata is listed in the RDB.

2.8.3 delete-content

Deletes all content (both structured and unstructured) from a table, but leaves its structure intact. For example, running this command would allow you to re-ingest data without breaking any references to the table (for instance, any searches).

Option

- `[--d rest-folder]`: Context path

Example 2-37:

```
iashell> delete-content applications/Baseball/databases/Baseball-sql-db/schemas/BASEBALL/
tables/ FIELDING
```

2.8.4 index-build

Note: The following index commands end up using the Table Indexing job, which you can monitor from the IA Web App.

- `index-build`
- `index rebuild`
- `index-remove`
- `index-stop`

Initiates building of indexes on the table database.

Options

- `[--d RestFolder]`: Command context path.
- `[--extract-text boolean]`: Optional. Allows you to disable text extraction. Default is true, meaning text extraction function is enabled.
- `[--contained-files boolean]`: If both the `--extract-text` and `--contained-files` are set, any text in contained files will also be extracted and indexed.
- `[--indexes String]`: Optional. Allows you to limit the operation to specified indexes. If more than one index is required, separate each entry with either a comma or a space.

The index name supported is as follows:

- `'tableName'_'columnName'_idx`
- `'tableName'_'columnName'`
- `'columnName'_idx`
- `'columnName'`

Example 2-38: Example of the command with the --contained-files option

```
iashell> cd applications/Tickets/databases/Tickets-sql-db
iashell> index-build --contained-files
Requested building all indexes
OK
```

Example 2-39:

```
iashell> cd applications/Tickets/databases/Tickets-sql-db
iashell> index-build applications/Tickets/databases/Tickets-sql-db
Requested building all indexes for Tickets-sql-db
OK
=================================================================================
jobInstanceLink.href http://localhost:8765/systemdata/job-instances/79438f32-b1b6
                        -44ee-a43d-d32fffceb7ee
startDateTime        null
activities[0]        define
activities[1]        construct
status               SCHEDULED
---------------------------------------------------------------------------------
```

The following is the expected result of using the `index-build --extract-text false` option:

```
iashell> index-build --extract-text false
Requested building all indexes for Tickets-sql-db
OK
=================================================================================
jobInstanceLink.href http://localhost:8765/systemdata/job-instances/2510bcd2-f4fd
-4a21-85e9-e9fafa60dc37
startDateTime        null
activities[0]        define
activities[1]        construct
status               SCHEDULED
---------------------------------------------------------------------------------
```

```
iashell>cd applications/Tickets/databases/Tickets-sql-db
iashell>index-build --extract-text false --indexes
"ticket_cust_id_idx,ticket_ia_rowid_idx"
Requested building selected indexes ticket_cust_id_idx,ticket_ia_rowid_idx for Tickets-
sql-db
OK
=================================================================================
jobInstanceLink.href http://localhost:8765/systemdata/job-instances/2510bcd2-f4fd
-4a21-85e9-e9fafa60dc37
startDateTime        null
activities[0]        define
activities[1]        construct
status               SCHEDULED
---------------------------------------------------------------------------------
```

2.8.5 index-extract-text

Initiates text extraction and triggers a build of full-text indexes only. Required when adding languages for a database using full text indexing on unstructured content (for more information, see section 2.7.3 “Adding a custom language translation to IA Web App” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*). This job only indexes the primary and additional languages specified on an application. This command triggers the Table Indexing job to run. From the Table Indexing job history, view the job logs to see what was done. This command differs from the index-rebuild in that it does not update all indices.

Option

- `[--d RestFolder]`: Command context path
- `[--contained-files boolean]`: If both the `--extract-text` and `--contained-files` are set, any text in contained files will also be extracted and indexed.

Example 2-40: Example of the command with the --contained-files option

```
iashell> index-extract-text --contained-files
Requested text extraction and index for unstructured content
OK
```

2.8.6 index-rebuild

Rebuilds the index for the table application.

Options

- `[--d RestFolder]`: Command context path
- `[--extract-text boolean]`: Optional. Allows you to disable text extraction. Default is true, meaning text extraction function is enabled.
- `[--remove-extracted-text boolean]`: Optional. Allows you to disable removing the extracted text that was previously extracted. Default is true, meaning that the extracted text will be removed.
- `[--indexes String]`: Optional. Allows you to limit the operation to specified indexes. If more than one index is required, separate each entry with either a comma or a space.

  The index name supported is as follows:

  – `‘tableName'_'columnName'_idx`

  – `‘tableName'_'columnName'`

  – `‘columnName'_idx`

  - `‘columnName'`

- `[--contained-files boolean]`: If both the `--extract-text` and `--contained-files` are set, any text in contained files will also be extracted and indexed.

Example 2-41: Example of the command with the --contained-files option

```
iashell> index-rebuild --contained-files
Requested rebuilding all indexes
OK
```

Example 2-42:

```
iashell> index-rebuild applications/Tickets/databases/Tickets-sql-db
Requested rebuilding all indexes for Tickets-sql-db
OK
==================================================================================
jobInstanceLink.href http://localhost:8765/systemdata/job-instances/79438f32-b1b6
                     -44ee-a43d-d32fffceb7ee
startDateTime        null
activities[0]        remove
status               SCHEDULED
```

The following is the expected result of using the `index-rebuild --extract-text false --remove-extracted-text false` option:

```
iashell> index-rebuild --extract-text false --remove-extracted-text false
Requested rebuilding all indexes for Tickets-sql-db
OK
==================================================================================
jobInstanceLink.href http://localhost:8765/systemdata/job-instances/af76ac49-d6fe
-4fb4-8bb3-d485eef8d8f6
startDateTime        null
activities[0]        remove
activities[1]        define
activities[2]        construct
status               SCHEDULED
----------------------------------------------------------------------------------
```

```
iashell>cd applications/Tickets/databases/Tickets-sql-db
iashell>index-rebuild --extract-text false --indexes
"ticket_cust_id_idx,ticket_ia_rowid_idx"
Requested building selected indexes ticket_cust_id_idx,ticket_ia_rowid_idx for Tickets-
sql-db
OK
==================================================================================
jobInstanceLink.href http://localhost:8765/systemdata/job-instances/2510bcd2-f4fd
-4a21-85e9-e9fafa60dc37
startDateTime        null
activities[0]        remove
activities[1]        define
activities[2]        construct
status               SCHEDULED
----------------------------------------------------------------------------------
```

2.8.7 index-remove

Removes the indices for a table application. Normally only used to improve table ingestion performance if additional data needs to be ingested.

Option

- `[--d]`: Command context path
- `[--indexes String]`: Optional. Allows you to limit the operation to specified indexes. If more than one index is required, separate each entry with either a comma or a space.

  The index name supported is as follows:

  – `‘tableName'_'columnName'_idx`

  – `‘tableName'_'columnName'`

  – `‘columnName'_idx`

  – `‘columnName'`

Example 2-43:

```
iashell> index-remove applications/Tickets/databases/Tickets-sql-db
Requested removing all indexes for Tickets-sql-db
OK
=================================================================================
jobInstanceLink.href http://localhost:8765/systemdata/job-instances/79438f32-b1b6
                     -44ee-a43d-d32fffceb7ee
startDateTime        null
activities[0]        remove
status               SCHEDULED
```

```
iashell>cd applications/Tickets/databases/Tickets-sql-db
iashell>index-remove --indexes "ticket_cust_id_idx,ticket_ia_rowid_idx"
Requested removing selected indexes ticket_cust_id_idx,ticket_ia_rowid_idx for Tickets-
sql-db
OK
=================================================================================
jobInstanceLink.href http://localhost:8765/systemdata/job-instances/2510bcd2-f4fd
-4a21-85e9-e9fafa60dc37
startDateTime        null
activities[0]        remove
status               SCHEDULED
---------------------------------------------------------------------------------
```

2.8.8 index-status

Shows the database index status. It can be run from the database context.

Option

- `[--d]`: Command context path

Example 2-44:

```
iashell> index-status applications/Tickets/databases/Tickets-sql-db
===================
indexesInPlace true
-------------------
```

Argument: Path in terms of the `cd` and `pwd` commands. The default value is the current path.

```
iashell> index-status applications/Tickets/databases/Tickets-sql-db
```

2.8.9 index-stop

Stops indexing for a table database.

Option

- `[--d]`: Command context path

Argument: Path in terms of the `cd` and `pwd` commands.

The default value is the current path.

Example 2-45:

```
iashell> index-stop
Stop of index job has been requested for 'Tickets-sql-db'
iashell> index-stop applications/Tickets/databases/Tickets-sql-db
Stop of index job has been requested for 'Tickets-sql-db'
```

```
iashell> index-stop applications/Tickets/databases/Tickets-sql-db
```

2.8.10 ingest-table-metadata

Ingests table metadata. If only the default option is specified, the location of the files should be inferred according to the `examples\applications` folder’s structure. The file filter has default value configured in the `tableMetadataFilesFilter` property.

Options

- `[--d]`: Command context path
- `[--path]`: Can represent a file path or a directory containing files to be ingested. In case the specified path is a file, the `--filter` and `--recursive` flags would have no effect.
- `[--filter]`: Filter to be included during a file search. All the files that match the filter will be included in the search results. Would have no effect if `--path` value is a file. If no filters are specified, the filter is set to a default value from the configuration. This is RegExp like filter (for example: `--filter = ".*\.xml"`).

Example 2-46:

```
iashell> ingest-table-metadata applications/Tickets/databases/Tickets-sql-db --path
applications/Tickets/tables/metadata.xml
```

2.8.11 restore-database

Restores a database from an existing backup for a table application. The command can be run from the database path context or the path to the required database can be passed with the `-d` option.

! **Important**

If the table application is offline, it is not possible to restore the database and you will get the following message:

```
iashell> restore-database
No link with rel http://identifiers.emc.com/restore found!
```

The message also appears if the user does not have the correct permission to perform the command (must be an Administrator).

Option

- `[--d RestFolder]`: The context of a database object.

Example 2-47:

```
iashell> cd applications/Baseball/databases/Baseball-sql-db
iashell> restore-database
Restore has been processed successfully.
OK
```

2.8.12 view-chain-of-custody

Views the results of the last run of the chain of custody tests.

After running, the command outputs the value of the attribute `chainOfCustodyResult.testSuiteResults` for each table in the specified schema.

It also appends the report generated to the `chain-of-custody.log` file in the current folder.

Options

- `[--d]`: Command context path
- `[--file]`: Path to the file that contains the test definition (by default, the file location is `tools/config/iashell/chain-of-custody-schema.xml`).
- `[--detailed]`: If the option is set, the command produces detailed output for the report.

  Default value: `false`

Example 2-48:

```
iashell> view-chain-of-custody applications/Tickets/databases/Tickets-sql-db/schemas/
TICKETS --file
```

2.9 File operation commands

2.9.1 content-upload

Uploads a content for a configuration object.

Options

- `[--d]`: Command context path
- `[--from]`: The file from which to read the content.
- `[--storeName]`: A store name value.
- `[--storeId]`: A store ID value.
- `[--format]`: The content file format.
- DEPRECATED: `[--type --t]`: A configuration type alias.
- DEPRECATED: `[--id --last]`: An ID value (self link) of an object to be uploaded or its index in the result of the last `select` command performed.

Argument: <type alias>

Example 2-49:

```
iashell> content-upload query --id #{myObjectSelf} --storeName myFileStore --format xml
--from "/tmp/content/query.xml"
```

2.9.2 download

Downloads resource contents with the content-download link relation available to file. Downloaded filename will be retrieved from the server records. Download will be the default filename in case the server does not provide a filename.

Command uses a content filename specified by the user or content disposition HTTP header or the download filename.

Options

- `[--d Restfolder]`: Command context path. If no context is specified, the context is resolved from the current path context. Optional.
- `[--f --overwrite]`: Force overwrite of existing file.
- `[--force]` or`[--overwrite Boolean]`: Force overwrite of existing file. If not set, and the file already exists, the name of the file will be made unique. Optional default value: `false`.
- `[--to String]`: Specify filename. This option overrides the filename retrieved from the server. Optional.

Argument: Command context path

Example 2-50:

```
download
```

If we re-run the following command a second time, the file is updated with a unique filename, such as `test(1).gzip` or `ArchiveAudits_061b64d0-6fca-4e1e-ba01-4999b35836d6(1).log.gz`:

```
iashell> download --to C:\Temp\test.gzip
Start Download at [2025-01-09 13:40:45]
C:\Temp\test(1).gzip 100% [====================] 1697/1697 (0:00:00 / 0:00:00)
Downloaded C:\Temp\test(1).gzip
OK
End Download at [2025-01-09 13:40:46]
Download took 0.79 seconds
```

2.9.3 file-upload

Uploads a file by a relation specified. The relation specified will be used for the uploading the file.

Options

- `[--d]`: Command context path
- `[--from]`: Path to a file.
- `[--params]`: Upload parameters. Should be defined as a simplified JSON fragment (for example: `{paramName:"paramValue"}`).
- `[--rel]`: The relation to use for the command.

Argument: Command context path

Example 2-51:

```
iashell> file-upload --from "xml/myfile.xml" --rel contents --params "{storeName:
'file_store_01'}"
```

2.10 General administration commands

2.10.1 check-connection

Allows you to check if the connection with the given access credentials can be established with the storage or not. The check is supported for Amazon S3 and Dell EMC CAS (Content Addressed Storage) Elastic Cloud Storage. The `check-connection` command is available for the `StorageEndPointCredentials` context for Amazon S3 storage. The `content-address-storage` command is available for Dell EMC CAS Elastic Cloud Storage.

Options

- `[--d]`: Command context path. If no context is specified, the context is resolved from the current path context.
- DEPRECATED: `[--id --last]`: An ID value (self link) of a storage-end-point-credential object or its index in the result of the last `select` command performed.
- DEPRECATED: `[--rel]`: The relation to the object used to check the connection.

Default Argument:: Context path

Example 2-52:

```
iashell> pwd
/storage-end-points/ECS_Storage/storage-end-point-credentials/Test_Credentials iashell>
check-connection
Connection to ECS using StorageEndPointCredential [name: Test_Credential] was successfull
```

Navigate to storage-end-point-credential and run the command without any parameters:

2.10.2 connect

Establishes the connection with the IA Server.

There are two places where command options can be passed: directly via command line parameter or via the `application.yml` file. The parameters for the `connect` command that are passed as arguments take priority over the `application.yml` parameters.

The `connect` command can be used without any options provided. In this case, default values for the command options are taken from the `<INFOARCHIVE_ROOT>/config/iashell/application.yml` file.

When authenticating via Gateway, the user name and password values can be omitted in command line arguments and in the `application.yml` file. In this case, the user name and password are prompted to input interactively during the command's run.

Options

- `[/--p /--password]`: Password for user, provided with [`--u --user`]. If omitted, the IA Shell will prompt it. The password that is entered by the user in interactive mode must be the unencrypted value.

  If security is of any concern, then following approaches can be applied:

  1. Do not provide password with the `connect` command. If the password is omitted in both command parameter and the `config/iashell/application.yml` configuration file, IA Shell will prompt the plain text password in security mode by masking the password value.
  2. Use the encrypted value for the password for the `--u --user` property in the `config/iashell/application.yml`or `local_iashell.properties` files. This property can be used for IA Shell script automation. For more information, refer to Configuration files.
- `[--rest-api]`: The base REST API Uniform Resource Identifier (URI) of the IA Server.
- `[--gateway]`: Gateway URI.
- `[--u --user]`: User name
- `[--permissionsExpected]`: Semicolon-separated list of groups to check that the user belongs to.
- `[--tenant]`: Tenant name.

  Default value: `INFOARCHIVE`

- [--token]: Authorization token to use instead of a username and password. If security is of any concern, do not use this optional parameter as it is printed on the screen in plain text. Instead use encrypted value for corresponding property in IA Shell configuration file.

  The token property has highest priority over the --user option and connection.userName option from IA Shell's application.yml file. The following is an example of how to connect using the [--token] option:

  ```
  iashell> connect --token
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
  ```

- [-q --quiet]: Report nothing in quiet mode.

  Supported values: true/false

- [--cookie String]: Cookie name.

Argument: Set the base Uniform Resource Identifier (URI) for REST API of the IA Server. Note that this is only necessary if you want to connect to a different server than what was configured in the IA Shell's application.yml file.

Example 2-53:

```
iashell> connect --rest-api http://localhost:8765/services --gateway http://localhost:
8080 --user adam@iacustomer.com --password <PASSWORD>
```

```
iashell> connect -rest-api http://localhost:8765/services --gateway http://localhost:
8080 --user adam@iacustomer.com
```

```
iashell> connect --u adam@iacustomer.com --p password
```

```
iashell> connect --u adam@iacustomer.com
```

2.10.3 data-delete

Requests a background task to delete an application's data. The command can run from the application path context or the path to the required application can be passed with the -d option. This command is blocked if anything in the application is under hold or retention.

Options

- [--d]: Command context path.
- DEPRECATED: [--id --last]: An ID value (self link) of an application or its index in the result of the last select command performed.

Example 2-54:

```
iashell> data-delete /tenants/INFOARCHIVE/applications/Baseball
Deleted http:localhost:8765/systemdata/applications/81e47658-27bc-4347-b1de-
f2e620cdcb6b/
delete-data
OK
```

2.10.4 disconnect

Clears all of the context data and logs out.

There are no arguments or options for this command.

2.10.5 export

Allows you to export configuration for either: system, tenant, application or holding levels into the declarative configuration format. The output file format may be either `*.yml` or `*.zip`. It depends on the file extension passed with the `--to` option.

- If file extension is`*.zip`, then the exported file is a `ZIP` archive that contains the resources related to user role.
- If file extension is `*.yml`, then the exported file is in `YML` format without filtering resources by user role.
- If file extension is missing or different from `ZIP` or `YML`, then `YML` format is used by default.

For more information about export, user roles and exported resources, see section 8.5 “Export DC configuration” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

By default, if the `--to` option is not provided, the command prints the output directly to the console in `YML` format.

The command can run from a home, tenant, application or holding path context (if navigating to the context path with the `cd` command), or the path context to the required resource can be passed with the `--d` option.

When the export is done, the output does not contain the password values from security reasons.

You may want to update/review system-specific properties before importing the exported configuration to a different environment.

Options

- `[--d]`: Command context path. If no context is specified, the context is resolved from the current path context.
- `[--f --overwrite]`: Force overwrite of existing file.
- `[--to]:` Name of the file where the exported configuration will be saved.

  File name extension specifies the format of the exported configuration.

  – If extension is `ZIP`, DC configuration is exported with a module structure filtered by user role.

  – If extension is `YML`, DC configuration is exported in a single `YML` file without filtering by user role.

  - If the extension is absent or different from YML or ZIP, by default, the YML export format is used. If the file has a different extension, the export is done in YML format.

  Supports both relative and absolute paths for the file.

- [--ed --export-details]: Specifies that export should be done with details.

  The following values are permitted:

  - create: The default value
  - use_existing
  - create_or_update

Example 2-55:

```
iashell> export --to "/samples/phonecalls.yaml"
iashell> export --to "C:/PhoneCalls.zip"
iashell> export --to "C:/PhoneCalls.zip" --export-details
```

2.10.6 export-search

Exports an application search into ZIP with YML. The command can be run from the search path context or search path context can be passed to the command with the --d parameter.

Options

- [--d]: Command context path.
- [--to]: The path to export the search to.
- {--overwrite]: If the flag is used in the event that a search already exists. it will be replaced with new one.

  Optional default value: false

Example 2-56:

```
iashell> connect
iashell> cd applications/Audit/searches/Activity Reports
iashell> export-search --to C:/Search/ActivityReports.zip --overwrite true
```

2.10.7 import

Allows you to import a declarative configuration to the server. It takes a single argument that contains a path to the resources being imported.

If the `export` command was used to export the configuration into a `ZIP` file, the output contains properties that you may want to update/review before importing it. This allows you to change settings, such as passwords and paths, that vary from system to system. The `ZIP` file contains a `configuration.properties` file in the top level of the `ZIP` file. To edit it:

1. Open the `ZIP` file.
2. Edit the `configuration.properties` file.
3. Replace the updated `configuration.properties` file in the `ZIP` file.

Option

- `[--from]`: Path to the `YML`, `ZIP` file or directory

  When importing a configuration, if the file or directory does not exist, the error message reads: “The file or directory [name] does not exist and cannot be imported.” However, if the directory does not contain the `configuration.yml` file, the error message reads: “The directory [name] does not contain the file `configuration.yml` and cannot be imported.”

  If the option refers to a directory, the IA Shell compresses all of the directory's resources into a `ZIP` file and sends it to the server. Specifying a directory for the parameter allows you to install the `YML` configuration and all related configuration resources in one shot.

  If the option is specified by the `ZIP` file, it sends it directly to the server.

  If the option is specified by the file (not `zip-archive`), only a single file is sent to the IA Server.

Example 2-57:

```
iashell> import /samples/phonecalls.yaml
```

```
iashell> import /samples/phonecalls.zip
```

```
iashell> import --from /samples/PhoneCalls_ResourcesDirectory
```

2.10.8 import-search

Imports the search from a ZIP file (search format exported from UI) to the indicated application.

The import-search command can run from particular application context or from an application searched context.

If the search is in a declarative configuration format, it can also be imported with the import command.

IA Shell supports the ability to export a search from one application and import it to another. Search that is presented as DC configuration (*.zip), has application name specified in the YML files. During the import-search command run, IA Shell replaces the name of source application from YML to the name of target application, into which the import should be done (for a cross-application search, only use YML). The target application name is taken from the IA Shell path context, that is either explicitly passed by the [--d] option or is taken from current path in terms of the cd, ls and pwd commands.

By default, if importing a search that already exists in target application, the command reports an error. To update the existing search or overwrite it, use the [--overwrite] flag for the command. In this case, the old search is updated with the new relevant information. If a search is in use (for example, retention is being applied), attempting to update the search with import will report a different error indicating the search name and the details that can be viewed in the logs.

DC search configuration for table-based applications has the names of databases and schemas inside the YML files. There may be a need to substitute an old database and schema names during a search import. In this case, use [--schema] and [--database] parameters for the command.

To have backward compatibility, the import-search command supports searches in old format with the import.descriptor inside.

Options

- [--d]: Command context path. This is the default option.
- [--from]: Path(s) to one or more search files (ZIP) delimited by ; or : (the system-dependent path-separator character).
- [--overwrite]: If the flag is used, if the search already exists, it will be replaced with the new one.

  Default value: false
- [--schema]: The schema name that overwrites the existing schema name value in the import.descriptor file.
- [--database]: The database name that overwrites the existing database name value in the import.descriptor file.

Example 2-58:

```
iashell> import-search --from "C:/PhoneCalls/searches/Date_Operator.zip"

iashell> import-search --from "C:/PhoneCalls/searches/Date_Operator.zip" --overwrite

iashell> import-search --from "C:/PhoneCalls/searches/Date_Operator.zip" --overwrite

iashell> import-search --d applications/PhoneCalls --from "C:/PhoneCalls/searches/
Date_Operator.zip" --overwrite

iashell> import-search --d applications/PhoneCalls/searches --from "C:/PhoneCalls/
searches/Date_Operator.zip"
```

2.10.9 index-progress

Provides an index's progress when indexing is being done for SDPG system data, in particular the async upgrade task to sync SDPG indexes.

Example 2-59:

```
iashell> index-progress
There is no indexing processing happening now.
OK
```

2.10.10 move-content

Moves content from:

- One regular store to another regular store within an application.
- One result store to another result store within an application.
- If content was mistakenly created in a library store, use the command to move the content to a regular store within an application.

The command should run from the store-level or the path to the store should be provided with the `[--d]` option.

Options

- `[--d RestFolder]`: Command context path.
- `[--to String]`: Target store name
- `[--formats String]`: Set of comma-separated formats. If specified, moves only the specified formats; otherwise. all of the content is moved.
- `[--log-level String]`: Sets the log level.
- `[ --keep-binary boolean]`: Unless the `--keep-binary` parameter is specified, the content is marked for removal. The removal of the content is only done when the Clean job runs, assuming that phase is enabled.

- `[--priority]` or `[--p int]`: Option to specify the priority for the background request.
- [`--max-content-count Integer`]: Option to specify the maximum number of content to process. If provided, check the background request to determine if this maximum was reached, which means more content needs to be moved.

Example 2-60:

```
iashell> move-content --to new-store --formats sip_zip,ci_container --log-level DEBUG --
priority 2 --max-content-count 200000

Because you have requested a limit of the number of content to process, please check the
background request to determine if not all the content was moved.

Content move has been submitted as background process.
```

```
iashell> move-content --d default-store --to new-store
Content move has been submitted as background process.
```

```
iashell> move-content --d default-store --to new-store2 --keep-binary
Copy content (keep binary requested) has been submitted as background process.
```

2.10.11 product-info

Prints OpenText Information Archive product information. The command is useful when there is an issue and it is required to provide support information about the environment in which the product is installed.

There are no arguments or options for this command.

2.10.12 recover-system

Initiates the recovery operation for the entire system.

Options

- `[--f]` or `[--force]`: Force recover without manual user confirmation. Default value: `false`
- `--s or --skipHoldingPhase`: Allows you to skip the holding recovery phase in the disaster recovery process. Using this option significantly improves performance during the disaster recovery process. Refer to the Disaster Recovery white paper in the Champion Toolkit for more information. Default value: `false`.

Example 2-61:

```
iashell>recover-system --skipHoldingPhase
Holding phase will be skipped as requested.
WARNING: This command is intended to only be used for DISASTER RECOVERY of
         the entire system and it is expected that all IA Servers have been configured
         to disable background processing.

Do you still want to proceed? (y/N)
```

2.10.13 reset

Resets all system configuration settings or one system configuration setting to the default value.

Example 2-62:

```
iashell> cd /system-configuration-settings
iashell> reset

WARNING: This command resets all system configurations settings to default values and
cannot be undone.
        Do you still want to proceed? (y/N)

iashell> cd /system-configuration-settings/batch.chunk.all
iashell> reset

Current property is reset to the default value.
OK
```

2.10.14 set

Set a system configuration setting.

Tip: System settings require a boolean, positive integer, or string values. Entering the wrong type of value results in an error.

You can also determine the value type by referring to the **Global Settings** tab in the IA Web App.

Options

- `--value String`: This is the new value for the system setting. If this is not specified, the value will be set to a blank string. Some settings may require a Boolean or a number. For string value that includes spaces, surround the value in quotes. Optional

Example 2-63:

```
iashell> set 4
Current property is set to '4'
OK
```

```
iashell> set --value 4
Current property is set to '4'
OK
```

2.10.15 set-type-alias *** DEPRECATED

The command is deprecated. Use the navigation commands `cd`, `ls`, and `pwd` instead.

2.10.16 var-clear

Clears all of the content variables.

There are no arguments or options for this command.

2.10.17 var-get

Shows a variable by name or by value within a SpEL expression.

Options

- `[--name]`: The name of the variable to get.
- `[--value]`: The expression to evaluate.

Example 2-64:

```
iashell> var-get --name application
```

```
iashell> var-get --value #{tenant}
```

2.10.18 var-list

Lists the variables currently set in the context.

There are no arguments or options for this command.

2.10.19 var-set

Sets a user-defined context variable.

After the setting, address this variable by SpEL (for example: `#{myVar}`).

There are variables set by the IA Shell after the `connect` command. Also, each relation in the context is treated as a variable.

Options

- `[--name]`: The name of the variable to set.
- `[--value]`: The value of the variable to set.

Example 2-65:

```
iashell> var-set --name myVar --value someValue
```

2.11 Group and role commands

2.11.1 actions-view

Views actions by specified role name. It is available only after the `roles-view` command has been performed.

Options

- `[--name --role]`: Role name
- `[--output]`: The path to store the output to.

Example 2-66:

```
iashell> actions-view --role ADMINISTRATOR
```

2.11.2 groups-view

Views the groups.

There are no arguments or options for this command.

2.11.3 roles-update

Updates roles by specified group name. It is available only after the `groups-view` command has been performed.

Options

- `[--name --group]`: Group name
- `[--roles]`: Defines a set of comma-separated roles.

Example 2-67:

```
iashell> roles-update --group GROUP_ADMINISTRATOR --roles ADMINISTRATOR
```

2.11.4 roles-view

Views roles by specified group name. It is available only after the `groups-view` command has been performed.

Options

- `[--name --group]`: Group name

Example 2-68:

```
iashell> roles-view --group GROUP_ADMINISTRATOR
```

2.11.5 sync-otds

Updates group/role mapping from OTDS. Can be used to sync changes made using OTDS Admin.

This command is only available if you configure the OTDS (SSO) profile. It is not available for the `AUTHENTICATION_OTDS` profile.

This command takes no parameters and only requires that OTDS is configured and the user is an Administrator (the OpenText Information Archive role).

If syncing was required, the system returns `OK`. You are also notified if the command resulted in no changes.

2.12 Managed items commands

2.12.1 apply-hold

Applies a hold to an object. The data file has XML format and can include a lot of elements. All of the elements inside a file (except the root element) will be transformed to JSON format. The root tag must be named `data` and will be not transformed to JSON.

The application field could be omitted in the data file as the proper value could be retrieved from the context (current application value).

The following is an example of how the file should be tagged:

```
<?xml version="1.0"?>
<data>
   <itemsToProtect>#{myObjectSelfLinkContextVariable}</itemsToProtect>
   <itemsToProtect/>
   <holdSetName>MyHoldSet</holdSetName>
   <type>AIP</type>
   <application>#{application}</application>
</data>
```

Options

- `[--id --last]`: An ID value (self link) of a hold or its index in the result of the last `select` command performed.
- `[--from]`: The file from which to read the data (XML) for the applying.

Example 2-69:

```
iashell> apply-hold --id 1 --from xml/apply-hold-example.xml
```

2.12.2 apply-retention

Applies a retention policy to some objects. The data file has XML format and can include a lot of elements. All of the elements inside a file (except the root element) will be transformed to JSON format. The root tag must be named `data` and will be not transformed to JSON.

The application field could be omitted in the data file, as the proper value can be retrieved from the context (current application value).

Refer to Applying retention – scenarios for different examples demonstrating how to use the command.

Options

- `[--id --last]`: An ID value (self link) of a retention policy or its index in the result of the last `select` command performed.
- `[--from]`: The file from which to read the data (XML) for the applying.

Example 2-70:

```
iashell> apply-retention --id 1 --from xml/apply-retention.xml
```

2.12.3 approve-purge-list

Approves a purge candidates list.

Options

- `[--d]`: Context path

Example 2-71:

```
iashell> approve-purge-list purge-candidate-lists/myList
```

2.12.4 audit-disable

Disables an audit event.

There are three types of aliases related to this command:

- `audit-event-type`: System-wide;
- `tenant-audit-event-type`: Related to a tenant specified in the `use` command; and
- `app-audit-event-type`: Related to an application specified in the `use` command.

Options

- `[--id --last]`: An ID value (self link) of an audit event or its index in the result of the last `select` command performed.

Argument: Command context path

Example 2-72:

```
iashell> audit-disable --id 1
```

2.12.5 audit-enable

Enables an audit event.

There are three type aliases related to this command: `audit-event-type` (system wide), `tenant-audit-event-type` (related to a tenant specified in the `use` command), `app-audit-event-type` (related to an application specified in the `use` command).

Options

- `[--id --last]`: An ID value (self link) of an audit event or its index in the result of the last `select` command performed.

Argument: Command context path

Example 2-73:

```
iashell> audit-enable --id 1
```

2.12.6 cancel-purge-list

Cancels a purge candidates list.

Options

- `[--d]`: Context path

Example 2-74:

```
iashell> cancel-purge-list purge-candidate-lists/myList
```

2.12.7 reject-purge-list

Rejects a purge candidates list.

Option

- `[--d]`: Context path

Example 2-75:

```
iashell> reject-purge-list purge-candidate-lists/myList
```

2.13 Navigation commands

2.13.1 cd

Changes the current working directory.

For resources that have duplicate names or do not support SpEL, use the ID rather than name in the `cd` command (for example, for events). Then you are able to run a `view` or `delete` command.

For resources that do not have a unique name, if you run the `cd` command, the first column indicates the correct context for the object. If you pick the name field, it will not resolve:

```
iashell> cd aips iashell> ls
+------------------------------------+-----------------------+
|externalId    |name    |
+------------------------------------+-----------------------+
|67f03623-89e5-4739-a471-c103c63c7f10|PhoneCalls-CC-1000000-1|
|05de5518-fcde-48ea-b2a6-22cc9e0f0232|PhoneCalls-CC-1000000-1|
|c3d71600-cc21-42bf-93d1-a174b0b553e8|PhoneCalls-CC-1000000-1|
|f474fe8d-85cf-4c0c-b0a0-d0690a860365|PhoneCalls-CC-1000000-1|
|491e805d-be8a-4168-9b98-aff0ab8931f8|PhoneCalls-CC-1000010-1|
|ec7ab231-5980-4b3e-88be-9d6784448921|PhoneCalls-CC-1000009-1|
|e85abc71-2603-42bb-8a52-fb87dab4310d|PhoneCalls-CC-1000008-1|
|dcfa5053-6ccd-49d6-adda-8bed8c846c6f|PhoneCalls-CC-1000007-1|
|0ebb7507-3c47-45be-9f0a-e23c45ad668c|PhoneCalls-CC-1000006-1|
|90a96ea3-40d3-4103-9d90-9f50fd251f67|PhoneCalls-CC-1000005-1|
|    |    |
|--------------------------------    |    |
|next ->
+------------------------------------+------------------------+
iashell> cd PhoneCalls-CC-1000000-1
You have specified unknown directory (context path).
Details of the error have been omitted. You can use the stacktrace command to print the
full stacktrace.
```

In this case, the system is expecting you to use the external ID to reference the package. The following is the correct command to set the context to the first AIP in the list:

```
iashell> cd 67f03623-89e5-4739-a471-c103c63c7f10
iashell> ls
application
backup
contents
detach
groups
invalid
managed-items
rebuild-sip
recovery
restore
ris
searches
transform
```

Known Issue

It is not possible to navigate to the content items that are not on the first page of the returned results. However, the issue is minor, since most of the resources do not have more than 10 content items, which is default server page size. Use the `next` command to navigate to the next items, and then use `cd` command on the item listed.

If the name has spaces in it, do not surround the value with single or double quotes.

Options

- `[--d]`: Command context path.

Example 2-76:

```
iashell> cd applications/PhoneCalls
```

2.13.2 first

Allows you to navigate to the first page of the listed results. The command is available if there is more than one page in the result set of a previously run navigation command, and the current page is not first.

There are no arguments or options for this command.

2.13.3 last

Allows you to navigate to the last page of listed results. The command is available if there is more than one page in the result set of a previously run navigation command, and the current page is not last.

There are no arguments or options for this command.

2.13.4 ls

Lists directories and resources.

The following are special strings related to the paths (directories):

- `.` indicates the current directory.
- `..` indicates the parent directory.
- A path that starts with `/` indicates the absolute path from the root (the base > home REST API URL).
- A path that starts without `/` indicates the relative path from the current directory.

Options

- `[--d]`: Command context path. If no context is specified, the context is resolved from the current path context.
- `[--l --long]`: Enables long output mode if option is specified. With long mode, objects are displayed in a table with the additional properties of the objects, if supported. The long mode can be useful when resources are not presented by names, but by a universally unique identifier (UUID).

  Long mode supports additional properties of following objects:

  - Applications
  - Job definitions
  - Job instances
  - Contents
  - Audits
  - Audit event types
  - Retention applications

- PG/Lucence libraries

Default value: `false`.

Example 2-77:

```
iashell> ls applications/PhoneCalls
```

```
iashell> ls /tenants/INFOARCHIVE
```

2.13.5 next

Allows you to navigate to the next page of the listed results. The command is available if there is more than one page in the result set of a previously run navigation command, and current page is not last.

There are no arguments or options for this command.

2.13.6 page

Allows you to navigate to the page number specified when navigating the listed results. The command is available if there is more than one page in the result set of a previously run navigation command.

Argument: `<page number>` from `1` to `N` (total pages in the result set of the previous command performed).

2.13.7 prev

Allows you to navigate to the page number specified when navigating the listed results. The command is available if there is more than one page in the result set of a previously run navigation command.

There are no arguments or options for this command.

2.13.8 pwd

Displays the current path (context).

There are no arguments or options for this command.

Example 2-78:

```
iashell> pwd
```

2.13.9 relations

Shows all the relations available in the context.

There are no arguments or options for this command.

2.14 Resource operation commands

2.14.1 cat

Prints object properties.

Tip: Prior to using the `cat` command for a particular resource, run the Refresh Metrics job to ensure that the most up-to-date data is available for the resource.

Options

- `[--d]`: Command context path
- `[--format]`: Allows you to specify the output format. The value of the option can be:
  - `table`
  - `xml`
  - `yaml`

  Default value: `yaml`
- `[--properties]`: Define a set of properties to be shown.
- `[--to]`: The path to dump the output to.
- `[--linkFormat]`: Output format for properties that are links. The value of the option can be either `href` or `name`.

  Default value: `href`

Example 2-79: Print certain properties of an object:

```
iashell> cat applications/PhoneCalls --properties name,description
Object:
  name: PhoneCalls
  description: The application has customer support phone calls history
```

2.14.2 delete

Allows you to delete the resource or object by a given context path. If a context path is not provided with the `--d` argument, the current context path is used. If the object cannot be removed, an error message is issued.

Options

- `[--d]`: Command context path. If no context is specified, the context is resolved from the current path context.
- `[--type --t]`: A configuration type alias.
- `[--id --last]`: An ID value (self link) of an object or its index in the result of the last `select` command performed.

Argument: <type alias>

Example 2-80: To remove application use:

```
iashell> delete
```

```
iashell> delete applications/Baseball
```

```
iashell> delete --id #{objToDeleteSelfVariable}
```

2.14.3 delete-application-categories

Allows you to remove application categories. The command should be run either:

- From an `application-categories` path context, or
- The path context can be passed with the `--d` option.

By default, the command removes only unused application categories. However, there is a possibility to remove all existing application categories by providing the corresponding command option.

Options

- `[--d]`: Command context path. If no context is specified, the context is resolved from the current path context.
- `[--remove-all]`: If specified, the command allows you to remove all application categories, including those in use.

  Default value: `false`

Example 2-81:

```
iashell> delete-application-categories
```

```
iashell> delete-application-categories --d /tenants/INFOARCHIVE/application-categories
-- remove-all
```

2.14.4 rm

Allows you to delete the resource or object by a given context path.

Options

- `[--d rest-folder]`: Command context path. If no context is specified, the context is resolved from the current path context.

2.14.5 update

Allows you to change the object specified by an option and properties (names and values).

It updates the object listed by the `--d` property or, if not specified, the command updates the resource by the current context path. The command allows you to update a set of properties provided by the `--properties` option or by a file in YAML format that is provided by the `--from` option. In case both `--properties` and `--from` options are provided, the priority goes to the `--from` option and the `--properties` option is ignored.

The `update` command allows you to update:

- A single attribute
- Several attributes at once
- Attributes, presented as `href`.
- Arrays values

Options

- `[--d]`: Command context path. If no context is specified, the context is resolved from the current path context.
- `[--properties]`: The option allows to specify resource fields to be updated. The format of the data should be defined as key-value pairs. For complex updating cases, JSON format is supported. Keys are separated by a colon. For example:

  ```
  iashell> --properties "'name':'newName', 'status':'READY'"
  ```

- `[--from]`: Path to the YAML file with new resource attributes. The file can be created within the `cat` command, which dumps the resource representation to the file. If the option is specified, then value of `--properties` is not taken into account.
- DEPRECATED: `[--id --last]`: An ID value (self link) of an object to be updated or its index in the result of the last `select` command performed.
- DEPRECATED: `[--type --t]`: A configuration type alias

Argument: <type alias>

Example 2-82:

```
iashell> update --properties "'name': 'MyFavorite','priority': 1"
iashell> update --from NewBaseball.yaml
```

Chapter 3

User scenarios

3.1 Scenario: How to navigate resources

In this scenario, a user connects to the server, navigates over resources, and views tenant properties:

1. Connect to the server by running the `connect` command without any parameters. You are prompted to enter user name and password. By default, IA Shell is connected to Gateway, located at `http://localhost:8080`.

2. Check the current working path. It should be `tenants/INFOARCHIVE`:

   ```
   iashell> pwd
   ```

3. You are able to view a list of all available relations from the current REST path:

   ```
   iashell> ls
   ```

4. Navigate to the home resource:

   ```
   iashell> cd ../..
   ```

5. View the current REST node properties. For the root resource, only the `name OpenText Information Archive Home Resource` is displayed:

   ```
   iashell> cat
   ```

6. Navigate to the list of existing tenants:

   ```
   iashell> cd tenants
   ```

7. See the list of tenants:

   ```
   iashell> ls
   ```

8. Navigate to the `INFOARCHIVE` tenant:

   ```
   iashell> cd INFOARCHIVE
   ```

9. See the properties of the `INFOARCHIVE` tenant:

   ```
   iashell> cat
   ```

3.2 Scenario: How to update an object attribute

This scenario illustrates how to update a resource attribute:

1. Connect to OpenText Information Archive using the `connect` command:

```
iashell> connect
```

2. Navigate to the Baseball sample application and check that you are in the application context:

```
iashell> cd /tenants/INFOARCHIVE/applications/Baseball
iashell> pwd /tenants/INFOARCHIVE/applications/Baseball
```

3. Dump the object properties to a YAML file using the `cat` command:

```
iashell> cat --to NewBaseball.yaml
OK
```

4. Modify the YAML file and leave the properties required to update. For example, the file may contain following information:

```
Object:
name: Baseball archiveType: TABLE
category: Sport business data description: The sample application
```

5. Run the `update` command with the YAML file:

```
iashell> update --from NewBaseball.yaml
The resources with name 'Baseball' has been updated OK
```

   Run the `cat` command to check that the object properties were updated properly:

```
iashell> cat
```

3.3 Scenario: How to use declarative configuration

OpenText Information Archive provides the ability to configure your system using declarative configuration (DC), a YAML-based format. For more information, see section 8 “Declarative configuration” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

The following scenario illustrates how a user would connect to the server and install the sample PhoneCalls application from a declarative configuration (`ZIP` format), view object properties, export configuration from OpenText Information Archive to `YML` format, and remove the application from the server.

1. Connect to the server by providing user credentials and required Gateway URL:

```
iashell> connect --u sue@iacustomer.com --p password --gateway http://localhost:8080
```

2. Load the PhoneCalls configuration to OpenText Information Archive:

```
iashell> import "C:/Demo/PhoneCalls.zip
```

   Note: The configuration can be taken from OpenText Information Archive samples.

3. Check the current working directory to see in what REST path the context is available. It should be `tenants/INFOARCHIVE`:

   ```
   iashell> pwd
   ```

4. Navigate to PhoneCalls by running the following command:

   ```
   iashell> cd applications/PhoneCalls
   ```

5. Check the current working directory. It should be `tenants/INFOARCHIVE/applications/PhoneCalls`:

   ```
   iashell> pwd
   ```

6. View the available relations to ensure that the `holding` relation is available:

   ```
   iashell> ls
   ```

7. Navigate to the holdings by running:

   ```
   iashell> cd holdings
   ```

8. View the existing holding. There should be single one `PhoneCalls`:

   ```
   iashell> ls
   ```

9. Navigate to the holding:

   ```
   iashell> cd PhoneCalls
   ```

10. Check the REST context path. It should be `/tenants/INFOARCHIVE/applications/PhoneCalls/holdings/PhoneCalls`:

    ```
    iashell> pwd
    ```

11. Navigate to the application-level:

    ```
    iashell> cd ../..
    ```

12. Check that working directory is `tenants/INFOARCHIVE/applications/PhoneCalls`:

    ```
    iashell> pwd
    ```

13. Export the application configuration to declarative `YML` format by running the following command. As a result, you will have the configuration in file `C:/Demo/PhoneCalls.yaml`:

    ```
    iashell> export --to "PhoneCalls.yaml"
    ```

14. Delete the application. Ensure that you are in the `tenants/INFOARCHIVE/applications/PhoneCalls` REST path and run:

    ```
    iashell> delete
    ```

15. Check that now context path is switched to `tenants/INFOARCHIVE/applications`:

    ```
    iashell> pwd
    ```

16. Check that no applications are listed:

```
iashell> ls
```

3.4 Scenario: How to ingest a SIP

The following scenario illustrates how to connect to the server and ingest a sample SIP in the following steps:

- Receive
- Enumerate
- Ingest

To ingest SIP data:

1. Connect to the server:

   ```
   iashell> connect --u sue@iacustomer.com --p password --gateway http://localhost:8080
   ```

2. Navigate to PhoneCalls application:

   ```
   iashell> cd applications/PhoneCalls
   ```

3. Receive the SIP:

   ```
   iashell> receive --from "C:/Demo/PhoneCallsSample-2001.zip"
   ```

4. Enumerate the SIP:

   ```
   iashell> enumerate
   ```

5. Ingest the AIP:

   ```
   iashell> ingest
   ```

6. After ingestion, check the working directory. It should be `/tenants/INFOARCHIVE/applications/PhoneCalls/aips/<AIP_ID>`::

   ```
   iashell> pwd
   ```

7. Check the `state` property of the AIP:

   ```
   iashell> cat
   ```

3.4.1 Ingesting SIPs asynchronously

When packages are ingested asynchronously, an ingestion order item is created and run, when possible, by OpenText Information Archive.

To launch an asynchronous ingestion, use IA Shell or the REST API. To learn how to use the REST API to ingest data asynchronously, see *OpenText Information Archive REST API Developer's Guide* on support.opentext.com (https://support.opentext.com/).

For direct background ingestion, use the `receive` command with parameters `ingest-direct` and `async`:

```
receive --from "C:\applications\PhoneCalls\data" --ingest-direct --async
```

To launch background ingestion after a reception, use the `ingest` command with the `async` parameter:

```
receive --from "C:\applications\PhoneCalls\data\SIP.zip
ingest --async
```

To launch background ingestion as a result of the enumeration, use the `enumerate` command with the `ingest` and `async` parameters:

```
enumerate --ingest --async
```

It is also possible to use the `enumerate` command followed by the `ingest` command with the `all-ids` and `async` parameters:

```
enumerate
ingest --all-ids --async
```

3.5 Scenario: How to ingest table data

This scenario illustrates how to run the installation script for declarative configuration for a table application and ingest tables into the configuration. The scenario is completed by running a process to index the data.

1. Connect to the server:

   ```
   iashell> connect --u sue@iacustomer.com --p password --gateway http://localhost:8080
   ```

2. Load the Baseball configuration into OpenText Information Archive.

   ```
   iashell> import "C:/Demo/Baseball/config
   ```

   Note: The configuration can be taken from the OpenText Information Archive samples. At this point, import the configuration from the `config` directory with a set of files.

3. Check the current working directory to see if the REST path context is available. It should be `tenants/INFOARCHIVE`:

   ```
   iashell> pwd
   ```

4. Navigate to `Baseball` by running the following command:

   ```
   iashell> cd applications/Baseball
   ```

5. Check the current working directory. It should be `tenants/INFOARCHIVE/applications/Baseball`:

   ```
   iashell> pwd
   ```

6. View the available relations to ensure sure that the `databases` relation is available:

   ```
   iashell> ls
   ```

7. Ingest the tables with customer data into the application. The following illustrates the command to ingest data with two schemas, `Baseball` and `Tickets`, as the Baseball application contains data from both:

   ```
   iashell> ingest --d databases/Baseball-sql-db/schemas/BASEBALL--from "C:/Demo/
   Baseball/data/BASEBALL"
   ingest --d databases/Baseball-sql-db/schemas/TICKETS --from"C:/Demo/Baseball/data/
   TICKETS"
   ```

8. Start index process to have the possibility to perform a search:

   ```
   iashell> index-build --d databases/Baseball-sql-db
   ```

3.6 Scenario: How to ingest table data that has a foreign key relation to one or more other tables

The following scenarios illustrate how to ensure that ingestion will not fail when ingesting table data with foreign key relations to one or more other tables.

Add a new feature to ingest at the database-level

```
iashell> cd applications/Order_Management/databases/Order_Management-sql-db
iashell> ingest --from  "C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive\examples
\applications\OrderManagement\data\ORDER_MANAGEMENT" --recursive true
Start Ingest at [2024-02-05 09:28:46]
Processing table group 1 of 2: checking first to see if there is any
[2024-02-05 09:28:47] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-ORDERS-00001.xml
ORDER_MANAGEMENT-ORDERS-00001.xml 100% [=] 5978916/5978916 (0:00:01 / 0:00:00)
[2024-02-05 09:28:49] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
ORDERS-00001.xml. It took 2.126 seconds
[2024-02-05 09:28:50] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-REGIONS-00001.xml
ORDER_MANAGEMENT-REGIONS-00001.xml 100% [========] 963/963 (0:00:00 / 0:00:00)
[2024-02-05 09:28:50] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
REGIONS-00001.xml. It took 0.039 seconds
[2024-02-05 09:28:50] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-NATIONS-00001.xml
ORDER_MANAGEMENT-NATIONS-00001.xml 100% [======] 5499/5499 (0:00:00 / 0:00:00)
[2024-02-05 09:28:50] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
NATIONS-00001.xml. It took 0.036 seconds
[2024-02-05 09:28:50] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-PARTS-00001.xml
1380.docx   0% [===============================]    0/2451 (0:00:00 / 0:00:00)
[2024-02-05 09:28:50] Ingested file with attachments: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-PARTS-00001.xml. It took 0.200 seconds
[2024-02-05 09:28:51] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-PARTSUPPLIERREL-00001.xml
1780.docx 100% [===============================] 2459/2459 (0:00:01 / 0:00:00)
[2024-02-05 09:28:52] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
PARTSUPPLIERREL-00001.xml. It took 0.718 seconds
[2024-02-05 09:28:52] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-SUPPLIERS-00002.xml
[2024-02-05 09:28:52] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
```

```
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-SUPPLIERS-00001.xml
ORDER_MANAGEMENT-PARTSUPPLIERREL-00001.xml 100% [=] 2522045/2522045 (0:00:01 /
[2024-02-05 09:28:52] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
SUPPLIERS-00001.xml. It took 0.048 seconds
[2024-02-05 09:28:52] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
SUPPLIERS-00002.xml. It took 0.053 seconds
[2024-02-05 09:28:52] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-CUSTOMERS-00002.xml
[2024-02-05 09:28:52] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-CUSTOMERS-00001.xml
[2024-02-05 09:28:52] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-CUSTOMERS-00003.xml
[2024-02-05 09:28:52] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
CUSTOMERS-00003.xml. It took 0.266 seconds
[2024-02-05 09:28:52] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
CUSTOMERS-00002.xml. It took 0.268 seconds
[2024-02-05 09:28:52] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
CUSTOMERS-00001.xml. It took 0.273 seconds

Processing table group 2 of 2: checking first to see if there is any
[2024-02-05 09:28:53] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\1\ORDER_MANAGEMENT-LINEITEMS-00001.xml
ORDER_MANAGEMENT-LINEITEMS-00001.xml 100% [=] 36628110/36628110 (0:00:02 /
0:00[2024-02-05 09:29:04] Ingested file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\1\ORDER_MANAGEMENT-LINEITEMS-00001.xml. It took 11.598 seconds


Summary: 11/11 file(s) processed successfully.
OK
End Ingest at [2024-02-05 09:29:04]
Ingest took 18 seconds
```

Update an existing feature to ingest at the schema-level

```
iashell> ingest --d applications/Order_Management/databases/Order_Management-sql-db/
schemas/ORDER_MANAGEMENT --from  "C:/InfoArch-Server-Dist-Main/2024-01-22/infoarchive/
examples/applications/OrderManagement/data/ORDER_MANAGEMENT" --recursive true

Start Ingest at [2024-02-05 09:47:24]
Processing table group 1 of 2: checking first to see if there is any
[2024-02-05 09:47:25] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-ORDERS-00001.xml
ORDER_MANAGEMENT-ORDERS-00001.xml 100% [=] 5978916/5978916 (0:00:01 / 0:00:00)
[2024-02-05 09:47:27] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
ORDERS-00001.xml. It took 1.969 seconds
[2024-02-05 09:47:27] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-REGIONS-00001.xml
ORDER_MANAGEMENT-REGIONS-00001.xml 100% [========] 963/963 (0:00:00 / 0:00:00)
[2024-02-05 09:47:27] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
REGIONS-00001.xml. It took 0.034 seconds
[2024-02-05 09:47:27] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-NATIONS-00001.xml
ORDER_MANAGEMENT-NATIONS-00001.xml 100% [======] 5499/5499 (0:00:00 / 0:00:00)
[2024-02-05 09:47:27] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
```

```
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
NATIONS-00001.xml. It took 0.035 seconds
[2024-02-05 09:47:27] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-PARTS-00001.xml
1380.docx 100% [===============================] 2451/2451 (0:00:00 / 0:00:00)
[2024-02-05 09:47:28] Ingested file with attachments: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-PARTS-00001.xml. It took 0.164 seconds
[2024-02-05 09:47:28] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-PARTSUPPLIERREL-00001.xml
1780.docx 100% [===============================] 2459/2459 (0:00:01 / 0:00:00)
[2024-02-05 09:47:29] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
PARTSUPPLIERREL-00001.xml. It took 0.739 seconds [2024-02-05 09:47:29]
Starting ingestion for file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive\examples
\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
SUPPLIERS-00001.xml
[2024-02-05 09:47:29] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-SUPPLIERS-00002.xml
ORDER_MANAGEMENT-PARTSUPPLIERREL-00001.xml 100% [=] 2522045/2522045 (0:00:01 /
[2024-02-05 09:47:29] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
SUPPLIERS-00002.xml. It took 0.098 seconds
[2024-02-05 09:47:29] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
SUPPLIERS-00001.xml. It took 0.103 seconds
[2024-02-05 09:47:29] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-CUSTOMERS-00002.xml
[2024-02-05 09:47:29] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-CUSTOMERS-00003.xml
[2024-02-05 09:47:29] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\0\ORDER_MANAGEMENT-CUSTOMERS-00001.xml
[2024-02-05 09:47:30] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
CUSTOMERS-00003.xml. It took 0.294 seconds
[2024-02-05 09:47:30] Ingested file: C:\InfoArch-Server-Dist-Main\2024-01-22\infoarchive
\examples\applications\OrderManagement\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-
CUSTOMERS-00002.xml. It took 0.298 seconds [2024-02-05 09:47:30] Ingested file: C:
\InfoArch-Server-Dist-Main\2024-01-22\infoarchive\examples\applications\OrderManagement
\data\ORDER_MANAGEMENT\0\ORDER_MANAGEMENT-CUSTOMERS-00001.xml. It took 0.307 seconds

Processing table group 2 of 2: checking first to see if there is any
[2024-02-05 09:47:30] Starting ingestion for file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\1\ORDER_MANAGEMENT-LINEITEMS-00001.xml
ORDER_MANAGEMENT-LINEITEMS-00001.xml 100% [=] 36628110/36628110 (0:00:02 /
0:00[2024-02-05 09:47:46] Ingested file: C:\InfoArch-Server-Dist-Main
\2024-01-22\infoarchive\examples\applications\OrderManagement\data\ORDER_MANAGEMENT
\1\ORDER_MANAGEMENT-LINEITEMS-00001.xml. It took 15.849 seconds


Summary: 11/11 file(s) processed successfully.
OK
End Ingest at [2024-02-05 09:47:46] Ingest took 22 seconds
```

3.7 Scenario: How to run and view chain-of-custody tests

The following scenario illustrates how to use IA Shell to run and view chain-of-custody (CoC) tests.

The scenario also demonstrates the difference between dumping the REST resource with attributes of the CoC object using the `cat` command and the test run information obtained using the `view-chain-of-custody` command. For more information, refer to the command descriptions for chain-of-custody and view-chain-of-custody.

1. Connect to the IA Server by running the `connect` command without any parameters. You are prompted to enter user name and password. By default, IA Shell is connected to Gateway located at `http://localhost:8080`.

2. Navigate to the Baseball example application:

   ```
   iashell> cd applications/Baseball
   ```

   The CoC tests are available only for table-based applications and can be seen from the schema or table level.

3. Navigate to the schema level:

   ```
   iashell> cd databases/Baseball-sql-db/schemas/BASEBALL
   ```

4. Once IA Shell is at the particular schema level, it is possible to run the `chain-of-custody` command.

   ```
   iashell> chain-of-custody
   ```

   The results are the test log of ingestion to the IA Server, as well as the status of the test run. The following is a small section of the output (demonstration purposes):

   ```
   ....
   [2020-10-12 10:54:31] Ingested file: C:\Downloads\XXXXXXX\ <IA_ROOT>\config\iashell
   \chain-
   of-custody-table.xml. It took 0.361 seconds
   chain-of-custody-table.xml 100%
   [=====================================================================
   =============================================]
   ....
   Chain of custody results for schema BASEBALL: (Success: 1 / Failed: 0 / Runtime:
   292 ms)
     SchemaMetadataTest        Success:   1    Failed:   0    Runtime:    292 ms
       testSchemaMetadata        Success                                  292 ms
   Chain of custody results for table AWARDSMANAGERS: (Success: 6 / Failed: 1 /
   Runtime: 330 ms)
     ColumnDataTest            Success:   1    Failed:   0    Runtime:    181 ms
       testColumnData            Success                                  181 ms
     SpacesTest                Success:   2    Failed:   0    Runtime:     39 ms
       testLeadingSpaces         Success                                   23 ms
       testTrailingSpaces        Success                                   16 ms
     TableDataTest             Success:   1    Failed:   0    Runtime:      5 ms
       testTableData             Success                                    5 ms
     TableRowCountTest         Success:   2    Failed:   0    Runtime:    101 ms
       testTableRowCount         Success                                   11 ms
       testTableCount            Success                                   90 ms
         Table count of schema BASEBALL of database Baseball-sql-db is: 24; expected:
   ```

```
23
  MetadataTest              Success:   1    Failed:   0    Runtime:      4 ms
    testMetadata              Success                                     4 ms
....
```

5. The results are printed automatically when the tests are run, but the `view-chain-of-custody` command allows you to preview the results of the last run of chain-of-custody. The `view-chain-of-custody` command allows you to view the results of the last chain-of-custody run. The command can be run from the table’s schema path:

```
iashell> view-chain-of-custody
chain-of-custody-schema.xml 100%
[====================================================================
=============================================] 170/170 (0:00:00 / 0:00:00)
[2020-10-12 10:55:53] Ingested file: C:\Downloads\XXXXX\ <IA_ROOT>\config\iashell
\chain-
of-custody-schema.xml. It took 0.206 seconds
....
Chain of custody results for schema BASEBALL: (Success: 1 / Failed: 0 / Runtime:
187 ms)
  SchemaMetadataTest        Success:   1    Failed:   0    Runtime:    187 ms
    testSchemaMetadata        Success                                   187 ms
Chain of custody results for table AWARDSMANAGERS: (Success: 6 / Failed: 1 /
Runtime: 330 ms)
  ColumnDataTest            Success:   1    Failed:   0    Runtime:    181 ms
    testColumnData            Success                                   181 ms
  SpacesTest                Success:   2    Failed:   0    Runtime:     39 ms
    testLeadingSpaces         Success                                    23 ms
    testTrailingSpaces        Success                                    16 ms
  TableDataTest             Success:   1    Failed:   0    Runtime:      5 ms
    testTableData             Success                                     5 ms
  TableRowCountTest         Success:   2    Failed:   0    Runtime:    101 ms
    testTableRowCount         Success                                    11 ms
    testTableCount            Success                                    90 ms
      Table count of schema BASEBALL of database Baseball-sql-db is: 24; expected:
23
  MetadataTest              Success:   1    Failed:   0    Runtime:      4 ms
    testMetadata              Success                                     4 ms

    ....
```

6. It is also possible to dump the REST resource that contains attributes of the server-side resource that contains the CoC test results:

```
iashell> pwd
/tenants/INFOARCHIVE/applications/Baseball/databases/Baseball-sql-db/schemas/
BASEBALL
iashell> cat
Object:
    testSuiteResults:
    - failureCount: 0
        id: null
        runTime: 292
        testCount: 1
        testResults:
        - name: com.emc.ia.coc.tests.SchemaMetadataTest
            failureCount: 0
            id: null
            runTime: 292
            testCaseCount: 1
            testCaseResults:
            - name: testSchemaMetadata
            endTime: 1602489266870
            failureCount: 0
            id: null
            runTime: 292
```

3.8 Scenario: How to update the storage URL after a change in the IT infrastructure

The following scenario illustrates how to update storage properties in the event that your IT infrastructure has changed and storage becomes unavailable. It is assumed that the storage has `S3_GLOBAL` storage configured.

The OpenText Information Archive S3 client writes a file to S3 storage and waits for a response. If the response is not received from S3 storage within the specified time (the default is 50 seconds), a time-out happens and a `409 Conflict error` is issued. There is no way to setup this time-out from the IA Web App.

Configure the following:

- Minimum socket time out allowed: 1 SECOND
- Minimum part size allowed: 5 MB
- Maximum part size allowed: 100 MB
- Minimum multi-part upload threshold allowed: 25 MB
- Maximum multi-part upload threshold allowed: 100 MB

1. Connect to the server:

   ```
   iashell> connect --u sue@iacustomer.com --p password --gateway http://localhost:8080
   ```

2. The current working path is `tenants/INFOARCHIVE`. Navigate to the Home Resource to retrieve the list of storages:

   ```
   iashell> cd ../..
   ```

3. Navigate to `Storage Systems`:

   ```
   iashell> cd storage-end-points
   ```

4. View the list of existing storages. There should be one `S3_GLOBAL`:

   ```
   ls
   ```

5. View the properties of the storage. Run the `cat` command from the working directory `/storage-end-points`:

   ```
   iashell> cat S3_GLOBAL
   ```

   The following would be the output:

   ```
   Object:
     version: 1
     name: S3_GLOBAL
     createdBy: sue@iacustomer.com
     createdDate: 2018-08-07T15:39:38.019+03:00
     description: This is the global S3 storage
     id: 67790f94-090d-42c0-ae37-6f429890693f
     inUse: true
     lastModifiedBy: sue@iacustomer.com
     lastModifiedDate: 2018-08-07T15:39:38.019+03:00
     multipartUploadThreshold: 26214400
     partSize: 26214400
   ```

```
  proxyUrl: null
  socketTimeout: 50000
  type: S3
  url: http://there.is.no.connection.go
...
```

Note: The property of the URL has the value `http://there.is.no.connection.go`.

6. Update the property value:

```
iashell> update --properties "'url':'http://correct.address.to.go'"
```

7. Check the result of the command to ensure that the `url` property was updated:

```
iashell> cat S3_GLOBAL
```

The following would be the output:

```
Object:
  version: 2
  name: S3_GLobal
  createdBy: sue@iacustomer.com
  createdDate: 2018-08-07T15:39:38.019+03:00
  description: This is the global S3 storage
  id: 67790f94-090d-42c0-ae37-6f429890693f
  inUse: true
  lastModifiedBy: sue@iacustomer.com
  lastModifiedDate: 2018-08-07T15:47:28.655+03:00
  multipartUploadThreshold: 26214400
  partSize: 26214400
  proxyUrl: null
  socketTimeout: 50000
  type: S3
  url: http://correct.address.to.go
```

3.9 Configuring the restrictRolesToApplicationGroups feature

With the `restrictRolesToApplicationGroups` feature enabled, what roles a user gets for an application are based on the role mappings for the groups specified. For more information, see section 3.10.2 “Managing permissions” in *OpenText Information Archive - Administration Guide (EARCORE-AGD)*.

To enable the feature, run the following:

```
iashell> cd /system-configuration-settings/permission.restrictRolesToApplicationGroups
iashell> set true
```

Restart the IA Server after enabling the group permissions restriction feature, as it might take time to update the cache if you have more than one IA Server.

3.10 Scenario: How to apply retention

When using the apply-retention command, an XML file must be prepared. The XML file should contain the following information:

itemsToProtect

The items that are going to be protected under the retention policy. The items are defined with a self-link or row identifier (in case of applying retention to the row or AIU) of the resource.

retainedSetName

The name of the retention set to be created when applying retention to items to be protected.

application

The self-link to the application in which the items to protect are located (refer to *OpenText Information Archive REST API Developer's Guide* on support.opentext.com (https://support.opentext.com/) for more information).

type

The type of the element to which the retention is going to be applied. Can be either aip, aiu, row or application.

baseDate

The date from when the retention is taken to be calculated.

searchComposition

The self-link to the search composition in case the retention is required to be applied on records or AIU elements.

3.10.1 Scenario: How to apply retention to known elements

This scenario demonstrates how to apply retention to an AIP.

In the scenario it is assumed that the:

- Self-link for Item to protect is known,
- Self-link for application is known,
- XML is prepared with hard-coded self-links,
- Retention is applied on the AIP element

1. Prepare the XML document. The XML document contains pre-defined self-links to the AIP resource and application.

```
<?xml version="1.0"?>
<data>
    <itemsToProtect>
        <itemToProtect> http://localhost:8765/systemdata/applications/
dc48623f-05b3-4c4b-a4fc-597d9506886a/0522dc83-651f-4b96-9739-da0a3994e80f</
itemToProtect>
    </itemsToProtect>
    <itemsToProtect/>
    <retainedSetName>set.PhoneCalls</retainedSetName>
    <type>aip</type>
    <application>http://localhost:8765/systemdata/applications/dc48623f-05b3-4c4b-
a4fc-597d9506886a</application>
    <baseDate>2020-01-01T12:00:00.054-07:00</baseDate>
</data>
```

2. Connect to IA Shell and run the `apply-retention` command with passing context path to the required retention policy with the `--d` option, and use `--from` to reference the XML file prepared in Step 1.

   ```
   iashell> connect
   Connected to "http://localhost:8765/services" as sue@iacustomer.com
   iashell> apply-retention --d /tenants/INFOARCHIVE/retention-policies/PhoneCalls-
   policy --from C:\Downloads\retention.xml
   Applied http://localhost:8765/systemdata/retention-policies/
   7a2e1d67-8bc3-453b-8120-4555c2b2f86e/retained-sets
   OK
   iashell>
   ```

The result is that retention has been applied to the indicated AIP. Use the IA Web App to verify the retention has been applied.

3.10.2 Scenario: How to investigate the links with the cd command and preserving them in variable to apply-retention

The previous scenario, Applying retention to pre-known elements, covers the case when the self-links are known. This scenario, however, helps to find out the links and save them in variables that are used in the XML file.

1. Prepare the XML file with variables. There are variables for:

   - The AIP self-link or AIP ID: The variable name is `aipID`.
   - Application self-link: The variable name is `applicationSelfLink`.

   ```
   <?xml version="1.0"?>
   <data>
       <itemsToProtect>
           <itemToProtect>#{aipID}</itemToProtect>
       </itemsToProtect>
       <itemsToProtect/>
       <retainedSetName>set.PhoneCalls</retainedSetName>
       <type>aip</type>
       <application>#{applicationSelfLink}</application>
       <baseDate>2020-01-01T12:00:00.054-07:00</baseDate>
   </data>
   ```

2. Save AIP ID to `aipId`variable with IA Shell.

   Connect IA Shell to the server and navigate to the application's AIPs. Once it is done, then determine the required AIP IDs and save them to the variable with the name `aipId` using the `var-set` command. Once the variable is set, its value can be checked using the `var-get` command.

   If you prefer to use the self-link of an AIP, use the approach from Step 3 to find out the AIP self-link.

```
iashell> connect

Connected to "http://localhost:8765/services" as sue@iacustomer.com
iashell> cd applications/PhoneCalls/aips
iashell> ls
+------------------------------------+-----------------------+
|externalId                          |name                   |
+------------------------------------+-----------------------+
|cb3d1d49-6bca-4d44-89ca-18c835ea455e|PhoneCalls-CC-1000010-1|
|a2e905b4-16c1-495b-819b-17ac18c717a1|PhoneCalls-CC-1000009-1|
|d375c589-68b9-474f-9e3a-7e62490d6e0b|PhoneCalls-CC-1000008-1|
|d4e0186c-fb57-40e8-b454-7965b5e3e5da|PhoneCalls-CC-1000007-1|
|04bb20ff-0d5a-44cd-b0b0-1315cec55283|PhoneCalls-CC-1000006-1|
|eae73c14-e8b6-42e5-bf24-9819fab44d70|PhoneCalls-CC-1000005-1|
|7667ac8f-b0a1-4f32-a6e5-3971e80d7d7d|PhoneCalls-CC-1000004-1|
|85d429e0-64a6-449e-9472-0203985609fb|PhoneCalls-CC-1000003-1|
|ef9f0d4a-e65f-435e-b516-436991813b1c|PhoneCalls-CC-1000001-1|
|04ad2dd6-b850-4c41-9027-79a00c93f428|PhoneCalls-CC-1000000-1|
|                                    |                       |
+------------------------------------+-----------------------+

iashell> var-set --name aipID --value cb3d1d49-6bca-4d44-89ca-18c835ea4455e
iashell> var-get aipId
cb3d1d49-6bca-4d44-89ca-18c835ea4455e
```

3. Save application SelfLink to `applicationSelfLink`. Navigate to the application and find out its self-link using the `view` command and note the relations. Save it to the variable with name `applicationSelfLink` using the `var-set` command. Once the variable is set, then its value can be checked with the `var-get` command.

```
iashell> cd /tenants/INFOARCHIVE/applications/PhoneCalls
iashell> view --relations self
13:45:04.507 WARN  - Deprecated command: view

====================================================================================
====================================================================
groups                             http://localhost:8765/systemdata/applications/
32d9a44d-df4a-4548-855f-868a358c9fc3/groups
self                               http://localhost:8765/systemdata/applications/
32d9a44d-df4a-4548-855f-868a358c9fc3
update                             http://localhost:8765/systemdata/applications/
32d9a44d-df4a-4548-855f-868a358c9fc3
iashell> var-set --name applicationSelfLink --value http://localhost:8765/
systemdata/applications/32d9a44d-df4a-4548-855f-868a358c9fc3
iashell> var-get applicationSelfLink
http://localhost:8765/systemdata/applications/32d9a44d-df4a-4548-855f-868a358c9fc3
iashell>
```

4. Run the `apply-retention` command with the passing context path to the required retention policy with the `--d` option, and use `--from` to reference the XML file prepared in Step 1. The variables in the XML file that were defined are replaced with corresponding values from Step 2 and Step 3.

```
iashell> apply-retention --d /tenants/INFOARCHIVE/retention-policies/PhoneCalls-
policy --from C:\Downloads\retention.xml
Applied http://localhost:8765/systemdata/retention-policies/
7a2e1d67-8bc3-453b-8120-4555c2b2f86e/retained-sets
OK
```

3.10.3 Scenario: How to apply retention to records (AIU and table)

The scenario helps to apply retention to records that are returned by a search. Unlike the previous scenario, for this scenario, it is required to know the record identifier returned by the search and know the self-link to the search composition.

The scenario can be utilized in combination with variables. Refer to Investigating the links with the cd command and preserving them in variable to apply-retention for more information about using variables.

When applying retention, either to an AIU or table row, the `itemToProtect` is the row identifier in the search. For example, if you do a simple record search and review the REST response for the search, you may find the AIU identifier that looks like the following: `c94aba05-2620-4f48-8284-31f3c932cdb6:aiu:2`. For table rows, depending on which release the table was ingested, the format for that identifier will vary. It typically looks like the following: `AAAAAgAAAAAgAAAABw`.

The prerequisites are the following:

- To find out the identifier of the records, run a search in the IA Web App (or with a REST call directly) and, in the detailed panel for selected rows, note the identifier.
- To find out the search composition self-link, use IA Shell and navigate to the search composition using the `cd` command. Use the `view` command with parameter for the relation to find out the value for the `selfLink` relation (refer to Investigating the links with the cd command and preserving them in variable to apply-retention to learn how to find a self-link based on the application example).

1. Prepare the XML document. The XML document contains a pre-defined row identifier and selfLinks to the AIU and application. Refer to Applying retention to pre-known elements for more information. Furthermore, a `searchComposition` node used:

```
<?xml version="1.0"?>
<data>
    <itemsToProtect>
        <itemToProtect>0522dc83-651f-4b96-9739-da0a3994e80f:aiu:1</itemToProtect>
    </itemsToProtect>
    <itemsToProtect/>
    <retainedSetName>set.PhoneCalls</retainedSetName>
    <type>aiu</type>
    <application>http://localhost:8765/systemdata/applications/dc48623f-05b3-4c4b-
a4fc-597d9506886a</application>
    <baseDate>2020-01-01T12:00:00.054-07:00</baseDate>
    <searchComposition>http://localhost:8765/systemdata/search-compositions/
146060d5-0fc3-476b-a781-97052766030d</searchComposition>
</data>
```

2. Connect to IA Shell and run the `apply-retention` command with passing context path to the required retention policy with the `--d` option, and use `--from` to reference the XML file prepared in Step 1.

```
iashell> connect
Connected to "http://localhost:8765/services" as sue@iacustomer.com
iashell> apply-retention --d /tenants/INFOARCHIVE/retention-policies/PhoneCalls-
policy --from C:\Downloads\retention.xml
Applied http://localhost:8765/systemdata/retention-policies/
7a2e1d67-8bc3-453b-8120-4555c2b2f86e/retained-sets
OK
```

The result is that the retention has been applied to the selected AIU of the AIP. Use the IA Web App to verify this.

3.11 Scenario: How to apply a hold to the AIP

The following scenario illustrates how to use the `apply-hold` command on files in the PhoneCalls sample application and verify that the hold set was created.

1. Connect as a Retention Manager.

   Note: Administrators do not see the hold alias.

2. Enter the `use` command:

```
iashell> use --tenant INFOARCHIVE
```

3. Display the list of holds:

```
iashell> ls holds
 PhoneCalls-hold
```

4. Enter the PhoneCalls application context:

```
iashell> cd /tenants/INFOARCHIVE/applications/PhoneCalls
```

5. Display the list of AIPs to determine which AIPs to protect:

```
+------------------------------------+-------------+
|externalId                          |name                   |
+------------------------------------+-----------------------+
|fdc86e23-78a9-4385-a327-e8c65c202540|PhoneCalls-CC-1000000-1|
|ec712e5a-5cdb-4f0a-9b54-9b2525eeca7b|PhoneCalls-CC-1000010-1|
|5a310449-1977-42e9-b062-b3b3eef736cf|PhoneCalls-CC-1000001-1|
|6b482b0e-710a-4a60-a929-9677bca887f0|PhoneCalls-CC-1000009-1|
|dc751217-9207-4866-b42c-2ef6f4ff0e87|PhoneCalls-CC-1000005-1|
|9e7a746f-f9ef-4ffa-ac0d-78df79db450d|PhoneCalls-CC-1000008-1|
|5878c335-d67b-48c4-a2fd-faa00743a665|PhoneCalls-CC-1000003-1|
|db4f2102-82ad-48db-97b5-11f4d7dd2e71|PhoneCalls-CC-1000006-1|
|500732c9-b646-42d0-86c5-799bf5d75d17|PhoneCalls-CC-1000004-1|
|b899470f-107a-4f55-9024-29fabdfc834b|PhoneCalls-CC-1000007-1|
|                                    |                       |
+------------------------------------+-----------------------+
```

6. In a text editor, create an XML file called `applyHold.xml`. If protecting one AIP, put the ID in the `<itemsToProtect>`. Be sure to enter the correct ID, as the system does not return an error if the AIP does not exist. Display the self ID of needed AIP with the `view` command.

```
iashell> cd aips
iashell> view fdc86e23-78a9-4385-a327-e8c65c202540--relations
=================================================================
self        http://localhost:8765/systemdata/applications/fdc86e23-78a9-4385-a327-
e8c65c202540/aips/5db15835-4cea-4710-b2a6-10e09f2e22fd
```

The following is an example of the `applyHold.xml` file:

```
<?xml version="1.0"?>
   <data>
     <itemsToProtect> http://localhost:8765/systemdata/applications/
     fdc86e23-78a9-4385-a327-e8c65c202540/aips/fdc86e23-78a9-4385-
     a327-e8c65c202540 </itemsToProtect>
     <itemsToProtect/>
     <holdSetName>MyHoldSet</holdSetName>
     <type>AIP</type>
     <application> http://localhost:8765/systemdata/applications/fdc86
     e23-78a9-4385-a327-e8c65c202540</application>
   </data>
```

7. Enter the hold context back:

   ```
   iashell> cd /tenants/INFOARCHIVE/holds
   ```

8. Enter the `apply-hold` command:

   ```
   iashell> apply-hold PhoneCalls-hold --from #{PATH_TO_FILE}/applyHold.xml
   ```

   Tips

   - If the following error message is received, hold sets cannot be shared:

     ```
     Command failed org.springframework.web.client.
     HttpClientErrorException: 409 errors:
     ProcessingException Hold set already in use use a new name
     ```

   - If the following error message is received, the file with the hold details is incorrect:

     ```
     400 errors:
     HttpMessage not readable. Problem reading the request body.
      on POST request for "http://localhost:8765/systemdata/holds/
     3937717a-cb84-4c3b-95be-e882686611fa/hold-sets"
     Details of the error have been omitted. You can use the
     stacktrace command to print the full stacktrace.
     ```

9. Enter the `cat` command to confirm the hold was applied:

   ```
   iashell> cd /tenants/INFOARCHIVE/applications/ PhoneCalls/aips
   iashell> cat e42a62e5-7449-4cca-8942-15cf17ea3c77
   =================================================================
   ...
   projectedDispositionDate 2002-02-28T00:00+01:00 underHold true
   underRetention true
   ...
   -----------------------------------------------------------------
   ```

3.12 Using the import command to create a fixed retention policy

When creating a retention policy using the `import` command, a configuration file must be used to specify the configuration.

The `configuration.yml` file should contain the following information:

```
tenant:
  name: INFOARCHIVE

retentionPolicies:
- name: The-First-policy
  agingStrategy:
    agingPeriod:
      units: years
      value: 15
    type: duration
  dispositionStrategy:
    type: destroy all
- name: The-Second-policy
  agingStrategy:
    retainUntil: 2029-08-28T22:00:00Z
    type: fixed_date
  dispositionStrategy:
    type: destroy all
```

Note: When defining a time in the duration attributes of the retention, pay attention to the time format used. After preparing the configuration file that contains the retention information, use the following IA Shell commands to import the configuration:

```
iashell> connect
iashell> import --from C:/Demo/configuration.yml
```

The following output should be displayed:

```
Connected to "http://localhost:8765/services" as sue@iacustomer.com
Start Import at [2019-08-30 12:12:54]
  Skipped tenant: INFOARCHIVE (not changed)

  Created retention policy: The-First-policy
```

3.13 Scenario: How to remove retention from an AIP

1. Connect to the IA Server:

   ```
   iashell> connect
   ```

2. Navigate to the AIPs of the required application:

   ```
   iashell> cd /tenants/INFOARCHIVE/applications/PhoneCalls/aips/
   ```

3. View the list of existing AIPs:

   ```
   iashell> ls
   +------------------------------------+----------------------+
   |externalId                          |name                  |
   +------------------------------------+----------------------+
   |cb3d1d49-6bca-4d44-89ca-18c835ea455e|PhoneCalls-CC-1000010-1|
   |a2e905b4-16c1-495b-819b-17ac18c717a1|PhoneCalls-CC-1000009-1|
   ```

```
|d375c589-68b9-474f-9e3a-7e62490d6e0b|PhoneCalls-CC-1000008-1|
|d4e0186c-fb57-40e8-b454-7965b5e3e5da|PhoneCalls-CC-1000007-1|
|04bb20ff-0d5a-44cd-b0b0-1315cec55283|PhoneCalls-CC-1000006-1|
|eae73c14-e8b6-42e5-bf24-9819fab44d70|PhoneCalls-CC-1000005-1|
|7667ac8f-b0a1-4f32-a6e5-3971e80d7d7d|PhoneCalls-CC-1000004-1|
|85d429e0-64a6-449e-9472-0203985609fb|PhoneCalls-CC-1000003-1|
|ef9f0d4a-e65f-435e-b516-436991813b1c|PhoneCalls-CC-1000001-1|
|04ad2dd6-b850-4c41-9027-79a00c93f428|PhoneCalls-CC-1000000-1|
|                                    |                       |
+------------------------------------+-----------------------+
```

4. Navigate to the particular AIP by its ID:

```
iashell> cd 04bb20ff-0d5a-44cd-b0b0-1315cec55283
```

5. Navigate to managed items:

```
iashell> cd managed-items
```

6. View the list of existing managed items:

```
iashell> ls
+------------------------------------+-----------------------+---------
+-------------------------+----+
|id                                  |name                   |underHold|
projectedDispositionDate |type|
+------------------------------------+-----------------------+---------
+-------------------------+----+
|c51ccffd-1f22-4b19-a858-10f4d1b28e54|PhoneCalls-CC-1000006-1|false    |
2007-02-28T00:00:00+01:00|aip |
|                                    |                       |
|                         |    |
+------------------------------------+-----------------------+---------
+-------------------------+----+
```

7. Navigate to the managed item by its ID attribute:

```
iashell> cd c51ccffd-1f22-4b19-a858-10f4d1b28e54
```

8. Navigate to retention applications with `cd` command:

```
iashell> cd retention-applications
```

9. See the list of existing retention-applications:

```
iashell> ls
+------------------------------------+---------------
+---------------------------------+---------------------------------------
+-------------------+---------------+
|id                                  |ageIndividually|
applicationDate                  |retainedSetName                        |
retentionPolicyName|managedItemType|
+------------------------------------+---------------
+---------------------------------+---------------------------------------
+-------------------+---------------+
|01e7c1b5-14d8-487b-8882-3038ff9d2ff6|true           |
2020-02-19T11:27:52.5370966-05:00|RS-04bb20ff-0d5a-44cd-b0b0-1315cec55283|
PhoneCalls-policy  |aip            |
|                                    |
|                                 |
|                   |               |
+------------------------------------+---------------
+---------------------------------+---------------------------------------
+-------------------+---------------+
```

10. Navigate to existing retention applications by its ID attribute:

```
iashell> cd 01e7c1b5-14d8-487b-8882-3038ff9d2ff6
```

11. Check current path with the `pwd` command:

    ```
    iashell> pwd
    ```

    The output should look like the following:

    ```
    /tenants/INFOARCHIVE/applications/PhoneCalls/aips/04bb20ff-0d5a-44cd-
    b0b0-1315cec55283/managed-items/c51ccffd-1f22-4b19-a858-10f4d1b28e54/retention-
    applications/01e7c1b5-14d8-487b-8882-3038ff9d2ff6
    ```

12. Delete the retention with the `delete` command and ensure that the status of the output is OK.

    ```
    iashell> delete
    OK
    ```

3.14 Scenario: How to remove a hold from an AIP

1. Connect to the IA Server:

    ```
    iashell> connect
    ```

2. Navigate to the AIPs of the required application:

    ```
    iashell> cd /tenants/INFOARCHIVE/applications/Phonecalls/aips/
    ```

3. View the list of existing AIPs:

    ```
    iashell> ls
    +------------------------------------+-----------------------+
    |externalId                          |name                   |
    +------------------------------------+-----------------------+
    |806db3b2-d011-402f-98ee-166ccabb72a8|PhoneCalls-CC-1000000-1|
    |                                    |                       |
    +------------------------------------+-----------------------+
    ```

4. Navigate to the particular AIP by its ID:

    ```
    iashell> cd 806db3b2-d011-402f-98ee-166ccabb72a8
    ```

5. Navigate to the managed items:

    ```
    iashell> cd managed-items
    ```

6. View the list of existing managed items:

    ```
    iashell> ls
    +------------------------------------+---------+---------------------
    ---+----+
    |id                                  |underHold|projectedDispositionDate|type|
    +------------------------------------+---------+---------------------
    ---+----+
    |0d65dd92-0ab8-4883-8ef9-0eee81fb727b|true     |
       |aip |
    |                                    |         |
       |    |
    +------------------------------------+---------+---------------------
    ---+----+
    ```

7. Navigate to the managed item by its ID attribute:

    ```
    iashell> cd 0d65dd92-0ab8-4883-8ef9-0eee81fb727b
    ```

8. Navigate to hold applications with `cd` command:

```
iashell> cd hold-applications
```

9. See the list of existing hold-applications:

```
iashell> ls
+------------------------------------+---------------+---------------
+---------------------------------+---------------+
|id                                  |holdName       |holdSetName
|applicationDate                  |managedItemType|
+------------------------------------+---------------+---------------
+---------------------------------+---------------+
|cdfe18f6-0644-40f1-9a73-ca667a26359a|hold.PhoneCalls|hold.PhoneCalls
|2019-03-26T13:19:50.5727518+03:00|aip            |
|                                    |               |
|                                 |               |
+------------------------------------+---------------+---------------
+---------------------------------+---------------+
```

10. Navigate to existing hold application by its ID attribute:

```
iashell> cd cdfe18f6-0644-40f1-9a73-ca667a26359a
```

11. Check current path with the `pwd` command:

```
iashell> pwd
```

The output should look like the following:

```
/tenants/INFOARCHIVE/applications/Phonecalls/aips/806db3b2-
d011-402f-98ee-166ccabb72a8/managed-items/0d65dd92-0ab8-4883-8ef9-
0eee81fb727b/hold-applications/cdfe18f6-0644-40f1-9a73-ca667a26359a
```

12. Delete the retention with the `delete` command and ensure that the status of the output is OK.

```
iashell> delete
OK
```

3.15 Scenario: How to list AIPs and check their statuses

It is often convenient to use IA Shell to check the status of AIPs.

The following scenario illustrates how a user would check the status of ingested AIPs in an application called Certificates. Because several tenants and applications might already exist, you must first instruct IA Shell which tenant and application you are interested in before you are able to list the AIPs.

1. List the available tenants using the `ls` command:

```
iashell> ls /tenants
INFOARCHIVE
SYSTEM
```

2. Go to the corresponding tenant:

```
iashell> cd INFOARCHIVE
```

3. Use the `ls` command to list the available applications:

```
iashell> ls applications
Certificates
```

The previously installed Certificates application is displayed.

4. Enter to the Certificates application:

```
iashell> cd Certificates
```

5. Enter the `ls` command to display the list of AIPs:

```
iashell> ls aips
+------------------------------------+-------------------------------+
|externalId                          |name
    |
+------------------------------------+-------------------------------+
|fdc86e23-78a9-4385-a327-e8c65c202540|Certificates-CertApp-2012-03-
05_001-1|
|ec712e5a-5cdb-4f0a-9b54-9b2525eeca7b|Certificates-CertApp-2012-11-
30_001-1|
|                                    |
    |
+------------------------------------+-------------------------------+
```

6. Enter the `cat` command to view the details of the first AIP:

```
iashell> cat aips/fdc86e23-78a9-4385-a327-e8c65c202540--properties name,state
Object:
  name: Certificates-CertApp-2012-03-05_001-1
  state: Completed
```

By default, not a lot of details are displayed. To see more information, you can instruct the `cat` command to display all the properties:

```
iashell> cat aips/fdc86e23-78a9-4385-a327-e8c65c202540
```

3.16 Scenario: How to run jobs with IA Shell

The following scenario illustrates how to use the IA Shell to run various OpenText Information Archive jobs. With the scenario, a user connects to the server, navigates and runs a job.

1. Connect to OpenText Information Archive using the `connect` command:

```
iashell> connect
```

2. Navigate to the home resource:

```
iashell> cd ../..
```

3. Display the list of job definitions:

```
iashell> ls job-definitions
CacheOut
Clean
Close
Commit
Confirmation
DisposePurgeCandidateList
Disposition RollForward Recovery
GeneratePurgeCandidateList
Invalidation
Table Indexing
--------------------------------
```

```
Elements count: 22. Page 1 of 3
next ->
```

4. Change the context path to the Confirmation job with `cd` command:

```
iashell> cd job-definitions/Confirmation
```

5. Run the job:

```
iashell> run-job --data “{‘now’:true}”
job-instance ID(s): http://localhost:8765/systemdata/job-instances/b8c08925-
ca91-4f28-8090-340217f5b83c
```

6. Verify that the job instance was created:

```
iashell> ls job-instances
 +------------------------------------+-------------------------------
----------------------------------------------------------------+-------+
|id                                  |jobDefinition
                                                               |status |
+------------------------------------+-------------------------------
---------------------------------------------------------------+-------+
|5c0cd7ab-b51c-4ef2-b279-4a67fcaf47c6|http://localhost:8765/
systemdata/job-definitions/c09d74cb-5f05-4493-8869-fa3db1a2ab86|SUCCESS|
|                                    |
                                                               |       |
+------------------------------------+-------------------------------
---------------------------------------------------------------+-------+
```

3.16.1 How to use the IA Shell to update job properties

You are able to use the IA Shell to change the job properties defined in the IA Web App and run the job. The following outlines the format of the command which updates the desired properties and runs the job: For more information, see the run job.

3.17 Scenario: How to manage groups and roles

The following scenario illustrates how to view and update groups and roles.

1. Run the `groups-view` command:

```
iashell> groups-view
GROUP_RETENTION_MANAGER
GROUP_END_USER
GROUP_AUDITOR
GROUP_DEVELOPER
GROUP_IT_OWNER
GROUP_BUSINESS_OWNER
GROUP_ADMINISTRATOR
```

2. Run the `roles-view` command for the `GROUP_IT_OWNER` group:

```
iashell> roles-view --group GROUP_IT_OWNER
IT_OWNER
```

3. Output the list of the actions for the `IT_OWNER` role to an external file, and then verify the content of the file. The next command will create the file in the working directory of IA Shell (for example, the command line current path when IA Shell was started). You can also output directly to IA Shell without specifying the file:

```
iashell> actions-view --role IT_OWNER --output it_act.txt
OK
```

4. Update `GROUP_IT_OWNER` group:

```
iashell> roles-update GROUP_IT_OWNER --roles "IT_OWNER, END_USER"
END_USER
IT_OWNER
```

3.18 Value lists

3.18.1 Managing value lists

Value lists can be imported with the application by using declarative configuration, but the IA Shell or the REST API must be used to view, update and delete value lists.

The following commands show how you can connect to, and navigate to the value lists of the Baseball application:

Connect:

```
iashell> connect --u sue@iacustomer.com --p password --gateway http://localhost:8080
```

Traverse to value-lists directory:

```
iashell> cd applications/Baseball/value-lists
```

Display value lists:

```
iashell> ls
```

The following list is displayed:

```
US_States
application
birth-years-selection
state-cities
```

To view the content of US_States:

```
iashell> cd US_States
iashell> cat
```

The object information of the US_States value list is displayed.

To delete the value list:

```
iashell> delete
```

To import a new value list into the Baseball application, create a new value list (for example, with the name `myValueList.xml`).

Create a `configuration.yml` file with the following content:

```
version: 1.0.0

tenant:
  name: INFOARCHIVE
  configure: use existing
```

```
application:
  name: Baseball
  configure: use existing

database:
  name: Baseball-sql-db
  configure: use existing

valueLists:
- content:
    format: xml
    resource: myValueList.xml
  name: myValueList
  description: A value list
```

Then navigate back to the value-lists directory:

```
iashell> cd ..
```

Import the `configuration.yml` file:

```
iashell> import -C://my/conf/location/configuration.yml
```

The value list is now imported.

3.18.2 Updating value lists

To update the name of a ValueList, use the IA Shell `update` command.

```
iashell> cd US_States
iashell> update --properties "'name':'New_name'"
The resource with name 'New_name' has been updated
OK
iashell> pwd
/tenants/INFOARCHIVE/applications/Baseball/value-lists/New_name
iashell> cd ..
iashell> ls
  New_name
  application
  birth-years-selection
  state-cities
```

3.18.3 Importing value lists

To create or update the content of a value list, prepare a YAML file with DC configuration for the `ValueList` description. As a reference, use the Baseball sample application in the `examples\applications\Baseball\config\value-lists` folder.

For example, the YAML file can be:

```
version: 1.0.0

tenant:
  name: INFOARCHIVE
  configure: use existing

application:
  name: Baseball
  configure: use existing

database:
  name: Baseball-sql-db
  configure: use existing
```

```
valueList:
  name: MyNewValueList
  content:
    format: xml
    resource: ValueListContent.xml
  description: My new value list
```

Prepare the XML file with `ValueList` content and put it in the same path as the YAML configuration. In our example, it is `C:/Demo/ValueListContent.xml`.

Run IA Shell and run the `import` command:

```
iashell> connect
iashell> import C:/Demo/ValueListConfiguration.yaml

import_configuration1540212510456.zip 100% [============================
================================] 1335/1335 (0:00:00 / 0:00:00)
Skipped tenant: INFOARCHIVE (already exists)
Skipped application: Baseball (already exists)
...
iashell> cd applications/Baseball/value-lists
iashell> ls
   MyNewValueList
   New_name
   application
   birth-years-selection
   state-cities
```

3.19 Search optimizations and XQuery techniques

3.19.1 Scenario: How to change metadata and re-index data

This scenario illustrates how to use IA Shell to re-ingest the Baseball metadata, to re-index the data, and check if the indexing job is done:

1. Load the Baseball metadata into a text editor and make the required changes. Save the metadata file.

   For the Baseball application, the metadata can be found in the `../infoarchive/examples/applications/Baseball/config/data-model-config/database-BaseballSqlDb-00001.xml` file.

2. Use IA Shell to update the metadata in OpenText Information Archive:

   a. Connect as **sue@iacustomer.com**.

      The following output is displayed:

      ```
      Connected to "http://localhost:8080" as sue@iacustomer.com
      ```

   b. Navigate to the Baseball database using the following command:

      ```
      iashell> cd applications/Baseball/databases/Baseball-sql-db
      ```

   c. Upload the modified metadata from `C:/temp/database-BaseballSqlDb-00001.xml` using the following command:

      ```
      iashell> ingest-table-metadata --from C:/temp/database- BaseballSqlDb-00001.xml
      ```

      The following output is displayed:

```
Ingested file: C:/temp/database-BaseballSqlDb-00001.xml
```

The metadata is uploaded.

3. Update the indexes using the following command:

   ```
   iashell> index-rebuild
   ```

   The following output is displayed:

   ```
   ===============
   jobStarted true
   cause      null
   ---------------
   The indexing job has started.
   ```

4. Check if the indexes are loaded using the following command:

   ```
   iashell> index-status
   ```

   The following output is displayed:

   ```
   ==================
   indexInPlace false
   ------------------
   ```

   The output shows that indexing is still in progress. Indexing is finished when the output shows:

   ```
   ==================
   indexInPlace true
   ------------------
   ```

3.20 Retention scenarios

The retention class defines a logical mapping of one or more retention policies. The retention class is defined in the holding the set of possible retention classes. The holding can also define a default retention class.

3.20.1 Checking an AIP object's state

The following illustrates how to use IA Shell to verify AIP information:

To check that all the AIPs are in the Completed state:

```
iashell> cd applications/<APPLICATION_NAME>/aips
iashell> ls --l
```

If required, then navigate over the pages using the `next` and `prev` commands.

To view several AIP properties in a convenient manner:

```
iashell> cd applications/<APPLICATION_NAME>/aips
iashell> ls
iashell> cd <aip_id>
iashell> cat --properties "id, name, state"
```

3.21 Creating rules

The following is an example of how to create the rule using IA Shell. For this example to work, you need to be in the working directory `C:/Demo` and the application needs to already exist.

1. Launch IA Shell

2. Connect to the IA Server:

```
iashell> connect
```

3. Create a YAML configuration file called `C:/Demo/rules_config.yml`.

```
version: 1.0.0

tenant:
  name: INFOARCHIVE
  configure: use existing

application:
  name: DemoApplication
  configure: use existing

rule:
  name: NewRule
  content:
    resource: ruleContent.drl
  type: APPLY_RETENTION
```

4. Place the file with the rules into the same location as the `YML` configuration resource (for example, `C:/Demo/ruleContent.drl`).

5. Run the following IA Shell command to install the configuration:

```
iashell> import -from C:/Demo/rules_config.yml\
```

6. Check with `cd` and `ls` commands that the `NewRule` has been successfully installed:

```
iashell> cd /tenants/INFOARCHIVE/applications/DemoApplication /rules
iashell> ls
   NewRule
```

Chapter 4

Troubleshooting

To resolve any issue with IA Shell, whether the user was in interactive mode or batch mode, review the log files located in the `<INFOARCHIVE_ROOT>/logs/iashell` folder. The error logs should contain the actual error with the stack trace. Use the message to determine the root cause of the error. There is also a log file that contains a history of each command's run.

4.1 Troubleshooting issues related to IA Shell commands

In general, in case of any issue with a command, refer to Chapter 2 – IA Shell commands for a particular command's description. If still unsure, refer to Chapter 3 – user scenarios to learn about a particular usage scenario. If a particular usage scenario has not been included in the guide, notify customer service so that we can update the documentation.

The most frequent issues with IA Shell commands are:

User is not connected to the OpenText Information Archive.

Run the `connect` command.

The following error message is received: "You have specified unknown directory (context path)."

The user incorrectly specified a context path to the resource. Refer to Using the IA Shell to navigate resources to learn more about the `cd` command or `--d` option, which can be used with almost all IA Shell commands.

The logged on user does not have enough permissions for the user to run the commands.

The relevant error message is provided to the user. Connect with a different user.

User sees the message that a specific command has been deprecated.

Refer to Appendix – deprecated commands use to learn about the commands that should be used instead. For every deprecated command, this section includes alternative commands to use in the event that the user ran a deprecated command.

The following error message is received: "Cannot upload the file."

Refer to the IA Shell logs folder for the stack trace of the issue. Most likely, the log will contain the root cause of the issue. If a huge file is uploaded, you will most likely have to increase the multipart parameter in the `<INFOARCHIVE_ROOT>/config/server/application.yml` file.

When experiencing other issues with the commands, use the help `--command xxx` (where xxx is the name of the command for which help is required).

Users may want to update the resource, but are not sure how to do this with IA Shell. Refer to Updating the object attribute to learn the scenario that explains how to update the resource with the `update` command.

Almost all IA Shell commands interact with the IA Server. There can be errors during command runs that do not contain information about the root cause of the issue, making it difficult to determine how to resolve the issue. In this case, refer to the IA Server logs (`logs/iaserver/errors.log` and `logs/iaserver/rest_warnings.log`) to learn more about the issue.

If an error message is not complete, helpful or does not provide a stack trace or other information, there is a possibility to setup a DEBUG log level for IA Shell (refer to Logging for further instructions).

For issues with declarative application installation, refer to Troubleshooting declarative configuration applications.

4.1.1 Troubleshooting issues with the connect command

If the issue occurred during the run of the `connect` command, IA Shell cannot interact with OpenText Information Archive without establishing a connection to the IA Server or the IA Web App Gateway. There are a number of issues that can occur when establishing a connection:

- An incorrect user name, password or gateway secret may have been provided in the `<INFOARCHIVE_ROOT>/config/iashell/application.yml` file. Ensure that the gateway secret matches the value in `webapp application-CLIENTS.yml` file.

  If contacting customer service, provide the entire `<INFOARCHIVE_ROOT>/config/` directory.

- If contacting customer service, provide the authentication mechanism used by OpenText Information Archive (LDAP/AD/OTDS/IN-MEMORY profile). Provide the entire `<INFOARCHIVE_ROOT>/config/` directory.

- It is also possible that HTTPS is wrongly configured. Review the `<INFOARCHIVE_ROOT>/config/iashell/application-https.yml` file to check every parameter that is specified for TLS/SSL and the corresponding passwords .

If you cannot access IA Shell using the `connect` command after two attempts, and the following error message is issued:

```
Command failed java.lang.IllegalArgumentException: The max number of connection attempts
has been reached.
Connection with given parameters is NOT established.
```

Ensure that the following values in the `applications.yml` file are correct:

- `connection.userName`
- `connection.userPassword`

- `connection.gatewayUrl`
- `connection.clientSecret`

4.1.2 Troubleshooting issues related to the import and update commands

When working with the `import` or `update` commands that take YAML files, there can be a syntax error in the input files. IA Shell and the IA server provide information about the place in the file that contains the syntax error. Refer to either the command run output or to the `logs/iaserver` or `logs/iashell` log files.

The rules regarding the importation of application configuration are more strict. Applications that could have been installed in previous versions may now fail validation. Unknown `enum` values were previously ignored and now result in a failure. IA Shell logs indicate what the value are set to and the possible values. For example, if we try to import a holding, and we accidentally set the `libraryMode` to `POOL` instead of `POOLED`, the following error may be issued:

```
400 errors:
Error: /holdings/Certificates
Message: Enum value INVALID-MODE is not supported at Holding.libraryMode
```

Previously, the system would have ignored the specified `libraryMode` and set the value to the default of `PRIVATE`.

4.2 Troubleshooting issues related to ingestion

If you experience an error during SIP reception or SIP ingestion, refer to the AIP ingest logs, which can be downloaded from the IA Web App's **Packages** tab. You can also view the IA Shell logs locally or look at the logs on the IA Server. If there was a connection error to the IA Server, only the local IA Shell logs will be created.

4.2.1 Ingesting legacy SIP formats

Can I continue to ingest the OpenText Information Archive 3.x SIP `ZIP` format with OpenText Information Archive 4.x?

It is possible to continue to ingest the legacy SIP format (OpenText Information Archive 3.x version) with a 4.x version by specifying the format `eas_sip_zip` instead of `sip_zip` during reception.

4.2.2 Ingesting large files

Whether you are ingesting SIP or table data using IA Shell, an error may be issued because of a time-out and/or large file size:

```
09:32:50.180 ERROR - Cannot upload the file: '...'. Please, check 'logs/iashell/
iashell.log' for additional information. => Error could be related to the configuration
properties [maxFileSize & maxRequestSize] if the file size greater than the property
value.
```

To ingest large data sizes, edit the IA Server's `config/iaserver/application.yml` file by increasing the values of the `infoarchive.rest.maxUploadSize` and `infoarchive.rest.multipartFileSizeThreshold` parameters. By default, the values are set to 2 GB. The IA Server must be restarted for any change to take effect.

To ingest large data sizes, edit the IA Server's `config/iaserver/application.yml` file by increasing the values of the `spring.servlet.maxFileSize` and `spring.servlet.maxRequestSize` parameters. By default, the values are set to 2 GB. The IA Server must be restarted for any change to take effect.

IA Shell can be configured to send requests directly to the IA Server or to use the Gateway.

- If IA Shell is configured to communicate directly to the IA Server, there is no time-out limit.
- If IA Shell is configured to communicate with the Gateway, you have a default time-out limit of 10 minutes. To increase the duration, edit IA Shell's `config/ishell/application.yml` file by increasing the value of the `defaultSettings.other.uploadGatewaySocketTimeout` parameter. By default, the value is set to 600,000 MS.

Important

When using the synchronous (direct) ingestion, the connection from a client, for example IA Shell to IA Web App, may stay idle for a long time. If there is a proxy between the two, it may close the connection due to timeout. In such cases, it is recommended that you use asynchronous ingestion. If it is not possible to use asynchronous ingestion, the proxy timeouts should be configured to be large enough to keep the connections open for the entire duration of the ingestion process.

If a timeout occurs during a synchronous (direct) ingestion procedure, before restarting the ingestion process, check that the ingestion is not really in progress. This might happen if the timeout occurred after the reception phase is complete.

4.2.3 Investigating the root cause of ingestion failure

Ingestion might fail because of a number of reasons but, from troubleshooting perspective, there are two possible cases.

In all the cases, the IA Shell logs should contain information explaining the root cause. Depending on the case, however, there may be slightly different actions that must be undertaken to troubleshoot a particular issue.

If you are ingesting only a few packages, it is easy to identify the problematic SIPs and fix the content or an application’s configuration. If there are a lot of SIPs in a batch (more than 100), and only some of them failed, it may be difficult to pinpoint the problematic SIPs.

In this case, use the following options for the `ingest` command: `--moveOnSuccess` and `--moveOnError`.

With help of the options, the initial SIP packages after ingestion are separated into folders, depending on the ingestion status. This process helps to identify any problematic SIPs, allowing you to fix the root cause and repeat the ingestion.

Two Cases

1. Holding does not exist or you do not have access to the application.

   Most likely, the reception failed. The following is an example of the error message:

   ```
   10:00:27.557 ERROR - Command failed: ingest applications/GNA --from data
   java.lang.IllegalStateException: java.lang.Exception: Cannot upload the file: 'D:
   \infoarchive\GNA\data\aip-CASEDETAILS-1_39.zip'. Please, contact 'logs/iashell/
   iashell.log' for additional information.
   400 errors:
   IllegalArgumentException Reception failed with error: Could not find the object
   Holding with the name WFIS for the application 6169e0e4-fd29-4d47-8d22-ae2f87c54139
   ```

   In this case, the options `--moveOnSuccess` and `--moveOnError` may help to separate the successfully ingested SIP from the problematic ones. Also, the log contains the name of the SIP that failed to ingest.

2. Holding does exist, but there is a problem with the package.

   ```
   14:56:06.503 ERROR - Cannot upload the file: 'data\Trades-2.zip'. Please, contact
   'logs/iashell/iashell.log' for additional information.
   org.springframework.web.client.HttpClientErrorException: 400 errors:
   Error: aip
   Message: Aip reception fails with code ERROR due to: 'java.lang.
   RuntimeException: The count of elements is not the same in the AIP:
   '2' than in the PDI: '3''. Consult aip id '2f361496-86d4-4e89-bd53-f808e42f778a'
   logs.
    on POST request for "http://localhost:8765/systemdata/applications/f366bd19-
   b1b6-45b0-a31a-939ba6e877b0/aips?ingestDirect=true"
   ```

   In addition to the benefits of `--moveOnSuccess` and `--moveOnError` options, there is an ingestion log that contains more details about all ingestion steps, including the step when ingestion failed. The ingestion log is attached to the AIP resource, and can be reached with either help of the IA Web App or IA Shell.

   In this case, the name of the initial SIP package is not helpful enough because there can be multiple AIPs with the same name. The SIP name is just a name of the data container and nothing more. The unique identifier of package that should be used for troubleshooting is `AIP ID`.

The logs contain an AIP ID, and this ID can be used in the IA Web App to filter the AIPs on the **Packages** page to download the AIP ingestion logs.

Ingestion logs can also be downloaded using the IA Shell `download` command.

4.2.4 Unsupported compression method during ingestion

Customers using a built-in Windows tool to create `ZIP` files may encounter an issue with ingestion.

The error message in the IA Shell reception log will be as follows:

```
14:35:08.618 ERROR - Cannot upload the file: 'data\64bitPhoneCalls-2001.zip'.
Please, contact 'logs/iashell/iashell.log' for additional information.
org.springframework.web.client.HttpServerErrorException: 500 errors:
Error: RuntimeException
Message: Reception failed with unexpected error: java.util.zip.ZipException:
invalid CEN header (bad compression method: 9)
on POST request for "http://localhost:8765/systemdata/applications/6bcc5639-
fee3-4f05-887c-8ac831c4234f/aips?ingestDirect=true"
```

The issue occurs because the built-in tool uses the DEFLATE64 compression method, even for files less than 4 GB.

Java supports compressing files larger than 4 GB by means of ZIP64, which is why OpenText Information Archive supports the ZIP64 format. A ZIP64 archive can be prepared with different compression methods. One of them is DEFLATE64. DEFLATE64 (compression method 9), however, is not supported by Oracle in all versions of JDK. DEFLATE64 is a proprietary variant of DEFLATE specified by PKWARE with its primary purpose being to provide a slightly better compression ratio and performance compared to DEFLATE (compression method 8),

The recommendation from OpenText Information Archive is to not use DEFLATE64 compression method (or Enhanced DEFLATE compression method).

To support compression of large `ZIP` files, customers can use third party tools, such as 7Zip or WinZIP. The 7Zip tool, for example, does not have a limitation on the file size when running:

```
"C:\Program Files\7-Zip\7z"a -mm=Deflate c:\sips\7sip.zip "C:\Reference\*"
```

Customers also have the option use Windows PowerShell, which allows the creation of `ZIP` files less than 2 GB.

Use the following Windows PowerShell command (all commands are case-sensitive):

```
Compress-Archive -Path C:\Reference\* -DestinationPath C:\sips\someSIP.zip
```

To specify the compression level:

```
Compress-Archive -Path C:\Reference\* -CompressionLevel Fastest -DestinationPath C:\sips
\someSIP.zip
```

Another option is to use the Jar Utility, which is part of the Oracle JDK):

```
jar -cf c:\sips\jarsip.zip c:\temp\1*
```

Note: OpenText Information Archive supports ZIP64 as long as the `ZIP` file does not use the DEFLATE64 compression method. The DEFLATE64 compression method is not licensed for use in Java.

4.3 Troubleshooting declarative configuration application installation

This section provides information in case a user experiences issues during declarative configuration (DC) application installation. Review the following checklist prior to contacting customer support about the issue.

- Which step did the user experience the issue? For example, be prepared to tell the customer support representative whether the issue occurred during which of the following steps:
  - Installing application: The entire `install.iashell` script runs, which includes connection, importing application configuration and ingesting the data steps.
  - Importing configuration: Only the `import` command was run to import the application configuration. When importing a configuration, if the file or directory does not exist, the error message reads: “The file or directory [name] does not exist and cannot be imported.” However, if the directory does not contain the `configuration.yml`, the error message reads: “The directory [name] does not contain the file `configuration.yml` and cannot be imported.”
  - Ingesting data: Only the `ingest` command was run to upload the data to the imported application configuration.
- If the application was not installed successfully during the `install.bat`/`install` script run, review the log of the application installation. Try to determine the root cause of failure based on the logs and include the following information when contacting customer support:
  - the installation logs (installation script output).
  - the `install.iashell` script with the steps to be performed during application installation.
  - IA Shell logs will also be required for first-line investigation. The logs can be found in the `<INFOARCHIVE_ROOT>/logs/iashell` directory.

- An issue may have occurred during the `connect` command run. The DC application cannot be installed without establishing the connection to OpenText Information Archive. The `connect` command should be the first command in the `install.iashell` script. A number of issues can occur while establishing a connection with the IA Server. More likely, the user provided an incorrect user name, password or gateway secret in the `<INFOARCHIVE_ROOT>/config/iashell/application.yml` file. Double-check the IA Shell configuration to ensure that the gateway secret matches the value in the `webapp application-CLIENTS.yml` file. If contacting customer support, provide the entire `<INFOARCHIVE_ROOT>/config/` directory.

- If the `connect` command ran successfully, but then the issue occurred during the `import` command, investigate if the command is properly used in the `install.iashell` script. Refer to the IA Shell guide for more information about command options and properties. The main issues can be:

  - The `import` command uses incorrect parameter and path to the configuration directory. In the out-of-box examples, the installation script contains the line `import config`, which makes the `import` command look for the `config` folder from the location where the installation script is launched.

  - If the `import` command finds the folder, but complains about a different issue (for example, a missing file, incorrect syntax, etc.), then review the error message. Try to fix the root cause in the configuration. If it is not possible, then provide the entire folder with the application configuration so customer support can reproduce the issue in-house. To simplify reproducing the case, provide the actual error message that was observed.

- If importing the application has completed successfully, but an issue occurs during the `ingest` command, investigate if the command is properly used in the `install.iashell` script. Refer to Chapter 2 – IA Shell commands for more information about command options and properties. The main issues with the `ingest` command can be:

  - If the `ingest` command uses an incorrect parameter and path to the configuration directory. In out-of-box examples, the installation script contains the line:

    ```
    ingest applications/Invoices --from data
    ```

    , which makes the command look for SIPs in a data folder from the place where installation script is launched.

  - User may also experience issues with SIPs. If contacting customer service, provide the ingestion logs.

- After ingestion, there can be set of commands used, based on the application type. The commands can be either related to job run, like `index-building` or running `chain-of-custody`. Refer to Chapter 2 – IA Shell commands for more information about command options and properties. If contacting customer service, provide the logs for the command run.

- If debugging the `install.iashell` script manually, it is possible to run each command in interactive mode with IA Shell. Open `<INFOARCHIVE_ROOT>/bin/iashell.bat` and run commands from the `install.iashell` file line by line looking for the errors and issues. Fix the command. Once done, move to next command to have entire script working.
- There can be cases when the application was installed for the first time, but a user attempted to update it with the same installation script (`install.iashell`) with a different configuration. The issues encountered can be:
  - The installation is not passing through. If this is the case, review the logs and try to fix the issue on-site. If it is not possible, provide installation logs and the current application configuration deployed in OpenText Information Archive (within the system configuration resources). To retrieve the current configuration:
    1. Run the `export` command.
    2. Run the `cd applications/xxx` command, where xxx is the name of application.
    3. Do the export.

    Refer to the export command for further information. Also refer to Working with a declarative configuration.
  - Installation is passing through, but the required configuration is not reached. Consequently, ingestion is not working as expected. In this case, investigate the current application configuration deployed in OpenText Information Archive using the `export` command. If contacting customer service, include this data.
- Errors can occur during the installation of a DC application, but the error messages do not contain information about the root cause of the issue. This is especially true if the problem is reported as 'Internal error'. In this case, review the IAServer logs (`logs/iaserver/errors.log` and `logs/iaserver/rest_warnings.log`) to extract sensitive and critical information. If contacting customer service, provide this data if it is not possible to resolve issue on-site.

Chapter 5

Appendix – deprecated commands

5.1 General administration commands

Deprecated Command	New Command
`recover`	Use the new `recover-system` command.
`set-type-alias`	Use the navigation commands `cd`, `ls`, and `pwd` instead.

5.2 Navigation commands

Deprecated Command	New Command
`list-aliases`	Use the new navigation approach with `cd` and `ls` commands.
`select`	Use the `cd`, `ls`, or `pwd` commands instead.
`use`	Use the `cd`, `ls`, or `pwd` commands instead.

5.3 Resource operation commands

Deprecated Command	New Command
`configure`	It is recommended that you use the `import` and `update` commands in YAML format instead.
`set-alias-properties`	Use the navigation approach with `cd` and `ls` commands.
`view`	It is recommended that you use the `cat` command instead.

5.4 Managed items commands

Deprecated Command	New Command
`remove-hold`	For further information, refer to Removing a hold from an AIP.
`remove-retention`	For further information, refer to Removing retention from an AIP.