

Unica Content Integration V12.1.7 Installation and Configuration Guide



Contents

- Chapter 1. Installing Unica Content Integration..... 1**
 - Deploying Unica Content Integration..... 1
 - Performance Tuning..... 7
- Chapter 2. Configuring Unica Content Integration..... 12**
 - Google Chrome and Microsoft Edge configuration..... 12
 - Configuring navigation settings..... 13
 - Configuring proxy settings..... 14
 - Configuring Kafka connectivity..... 15
 - Adding the Content Integration User role to a user or a user group..... 16
 - Logs configuration..... 16
- Chapter 3. Configuring a third-party CMS for integration with Unica Content Integration..... 17**
 - Deploying Unica Content Integration in a distributed environment.....21
 - Verifying Content Integration configuration in Unica Centralized Offer Management.....22
- Chapter 4. Configuring the out-of-the-box integrations offered by Content Integration..... 24**
 - Configuring Slack for Integration with Unica.....41

Chapter 1. Installing Unica Content Integration

The component Unica Content Integration is installed or upgraded as a part of Unica Platform installation or upgrade.

When installing Unica Platform, select **Asset Picker** in the **Components** screen to install Content Integration Framework. For more information, see *Unica Platform Installation Guide*.

Deploying Unica Content Integration

Prerequisites

- Unica Content Integration requires JRE 1.8 or higher to work.



Note: Unica Content integration can be clean installed. For more information, see [Configuring navigation settings on page 14](#)



Note: Unica Content Integration support clustered deployment and no additional configuration is required for clustered deployment.



Note: To launch the installers on Windows 2019 and Windows 2022, server please set parameter - **SET JAVA_TOOL_OPTIONS="-Dos.name=Windows 7"** - on command prompt and then launch the Installers from the same command prompt.

- UnicaPlatformDS JNDI resource for Platform database is required for Unica Content Integration. For more information on setting up UnicaPlatformDS data source, see *Unica Platform Installation Guide*.

Setting Unica Content Integration home JVM argument

1. Add the following JVM argument to the application server, where Unica Content Integration is required to be deployed.

```
-DASSET_PICKER_HOME= path_where_content_integration_is_installed
```

For example: ASSET_PICKER_HOME = /opt/Platform/AssetPicker

Point `ASSET_PICKER_HOME` to the directory, where Unica Content Integration is installed. It is installed within the `AssetPicker` directory under `UNICA_PLATFORM_HOME`.

2. Provide write permissions to the logs directory already created inside `ASSET_PICKER_HOME`.

Setting configuration refresh interval JVM argument

Add the following JVM argument to the application server, where Unica Content Integration is required to be deployed.

```
-Dplatform.config.refresh-schedule = valid_cron_expression
```

Value of this argument must be a valid cron expression in the following format. All fields are mandatory:

```
<seconds> <minutes> <hours> <day of month> <month> <day of week>
```

- <seconds> can have values 0-59 or the special characters , - * /
- <minutes> can have values 0-59 or the special characters , - * /
- <hours> can have values 0-23 or the special characters , - * /
- <day of month> can have values 1-31 or the special characters , - * ? / L W C
- <month> can have values 1-12, JAN-DEC or the special characters , - * /
- <day of week> can have values 0-6, SUN-SAT or the special characters , - * ? / L C #

Special characters in the cron expression

- * represents all values. So, if it is used in the second field, it means every second. If it is used in the day field, it means run every day.
- ? represents no specific value and can be used in either the day of month or day of week field – where using one invalidates the other. If we specify it to trigger on the 15th day of a month, then a ? will be used in the Day of week field.
- - represents an inclusive range of values. For example, 1-3 in the hours field means the hours 1, 2, and 3.
- , represents additional values. For example, putting MON,WED,SUN in the day of week field means on Monday, Wednesday, and Sunday.
- / represents increments. For example 0/15 in the seconds field triggers every 15 seconds starting from 0 (0, 15, 30 and 45).
- L represents the last day of the week or month. Remember that Saturday is the end of the week in this context, so using L in the day of week field will trigger on a Saturday. This can be used in conjunction with a number in the day of month field, such as 6L to represent the last Friday of the month or an expression like L-3 denoting the third from the last day of the month. If we specify a value in the day of week field, we must use ? in the day of month field, and vice versa.
- W represents the nearest weekday of the month. For example, 15W will trigger on the 15th day of the month if it is a weekday. Otherwise, it will run on the closest weekday. This value cannot be used in a list of day values.
- # specifies both the day of the week and the week that the task should trigger. For example, 5#2 means the second Thursday of the month. If the day and week you specified overflows into the next month, then it will not trigger.

Example

0 */30 * * * * - Every 30 minutes (Default if this JVM argument is not supplied)

0 0 15 * * ? - Every 3 PM

0 */15 */2 * * ? - Every 15 mins every 2 hours

0 0 0 * * * - Every midnight

0 0 * * * * - Every hour every day

0 0 */2 * * * - Every 2 hours every day

0 0 0 ? * 1 - Every Sunday midnight

0 0 0 ? * SUN - Every Sunday midnight

0 0 0 ? * SUN,WED,FRI - Every Sunday, Wednesday & Friday midnight

Content Integration Framework periodically looks for configuration changes as per the schedule specified in the `platform.config.refresh-schedule` JVM argument. If you have not set a value for this JVM argument, by default, Content Integration Framework looks for configuration changes every 30 minutes. Thus, changes in all configurations including user data sources, except Kafka settings are detected and applied at runtime without having to restart the application. All the added and removed partitions are identified and configured accordingly. For changes to reflect in the Kafka configuration, restart the application.



Note: If the JVM argument `UNICA_PLATFORM_CACHE_ENABLED` is set to `TRUE`, it might require up to an hour or more for detecting and applying the configuration changes. Expiry of user data source cache can be separately controlled using `cif.userDataSources` cache configuration provided in `ASSET_PICKER_HOME/conf/caching/cif_ehcache.xml`.

Setting JVM argument for Ehcache configuration

Unica Content Integration supports caching for content mapping & auto synchronization features for the sake of improved performance. It can be turned **ON** by setting `UNICA_PLATFORM_CACHE_ENABLED` JVM argument to true. By default, Content Integration uses the Ehcache configuration file placed at `ASSET_PICKER_HOME/conf/caching/cif_ehcache.xml`.

For clustered deployments of Unica Content Integration, wherein same `ASSET_PICKER_HOME` directory is shared by all the nodes, if `cif_ehcache.xml` file must be separated for each node, then copies of `cif_ehcache.xml` can be made inside same directory and `cif_ehcache.config.file` JVM argument can be supplied to specify the filename for each node.

For example,

```
-Dcif.ehcache.config.file=node1_cif_ehcache.xml
```

Setting JVM argument to turn background services off

Unica Content Integration makes use of background services for certain features such as processing of events received via Webhooks as well as from Kafka. Such background processing can occupy significant amount of CPU & memory if large volume of events is received by Unica Content Integration. This can occasionally degrade the experience of users accessing Content Integration features from UI. To address this situation, background services can be totally turned off for the instance(s) serving user requests. Supply following JVM argument to do so –

```
-Dcif.background.services=false
```

Having said that, if background services is also desired, then separate instance without above mentioned JVM argument must be dedicated for it. Sending user requests to event processing instance is not advisable if it is expected to process high volume of events.



Note: Setting this JVM argument is not required if heavy background processing is not expected.

Setting JVM argument for cache (Ehcache) configuration

Unica Content Integration supports caching for content mapping & auto synchronization features for the sake of improved performance. It can be turned ON by setting `UNICA_PLATFORM_CACHE_ENABLED` JVM argument to true. By default, Content Integration uses the Ehcache configuration file placed at `ASSET_PICKER_HOME/conf/caching/cif_ehcache.xml`.

For clustered deployments of Unica Content Integration, wherein same ASSET_PICKER_HOME directory is shared by all the nodes, if cif_ehcache.xml file must be separated for each node, then copies of cif_ehcache.xml can be made inside same directory and cif.ehcache.config.file JVM argument can be supplied to specify the filename for each node.

Example

```
-Dcif.ehcache.config.file=node1_cif_ehcache.xml.
```

Setting JVM argument to turn background services off

Unica Content Integration makes use of background services for certain features such as processing of events received via Webhooks as well as from Kafka. Such background processing can occupy significant amount of CPU & memory if large volume of events is received by Unica Content Integration. This can occasionally degrade the experience of users accessing Content Integration features from UI. To address this situation, background services can be totally turned off for the instance(s) serving user requests. Supply following JVM argument to do so -

```
-Dcif.background.services=false
```

Having said that, if background services is also desired, then separate instance without above mentioned JVM argument must be dedicated for it. Sending user requests to event processing instance is not advisable if it is expected to process high volume of events.

Note that setting this JVM argument is not required if heavy background processing is not expected.

Deployment procedure

You must follow a set of guidelines, when you deploy Unica Content Integration on your web application server. There is a different set of guidelines for deploying Unica Content Integration on WebLogic and on WebSphere. When you ran the suite installer, you completed the following action:

- You created the WAR file of Unica Content Integration (`asset-viewer.war`) inside `UNICA_PLATFORM_HOME/AssetPicker` directory. `UNICA_PLATFORM_HOME` refers to the Unica Platform installation location.



Note: It is assumed that you possess information on how to work with your web application server. For more information, see the web application server documentation.

Guidelines for deploying Unica Content Integration on WebLogic

You must follow a set of guidelines when you deploy Unica Content Integration on the WebLogic application. Use the following guidelines when you deploy Unica Content Integration on any supported version of WebLogic:

- Unica products customize the Java virtual machine (JVM) that is used by WebLogic. If you encounter errors related to JVM, you can create a WebLogic instance that is dedicated to Unica products.
- Open the `startWebLogic.cmd` file and verify that the SDK that is selected for the WebLogic domain that you are using is the Sun SDK for the `JAVA_VENDOR` variable.
- The `JAVA_VENDOR` variable must be set to `Sun` (`JAVA_VENDOR=Sun`). If the `JAVA_VENDOR` variable is set to `JAVA_VENDOR`, it means that JRockit is selected. You must change the selected SDK, as JRockit is not supported. See the *BEA WebLogic documentation* to change the selected SDK.

- Deploy Unica Content Integration as a web application.
- If you are configuring WebLogic to use the IIS plug-in, review the *BEA WebLogic documentation*.
- Complete the following tasks for your installation to support non-ASCII characters, for example for Portuguese or for locales that require multi-byte characters.
 1. Edit the `setDomainEnv` script in the bin directory under your WebLogic domain directory to add `-Dfile.encoding=UTF-8`.
 2. In the WebLogic console, click the Domain link on the home page.
 3. In the **Web Applications** tab, select the **Archived Real Path Enabled** check box.
 4. Restart WebLogic.
 5. Deploy and start the `asset-viewer.war` file.
- If deploying in a production environment, set the JVM memory heap size parameters to 1024 by adding the following line to the `setDomainEnv` script: `Set MEM_ARGS=-Xms1024m -Xmx1024m -XX:MaxPermSize=256m`

Guidelines for deploying Unica Content Integration on WebSphere®

You must follow a set of guidelines when you deploy Unica Content Integration on WebSphere. Ensure that the version of WebSphere meets the requirements that are described in the *Recommended Software Environments and Minimum System Requirements* document, including any required fix packs. Use the following guidelines when deploying Unica Content Integration on WebSphere:

1. Specify the following custom property in the server:
 - Name: `com.ibm.ws.webcontainer.invokefilterscompatibility`
 - Value: `true`
2. Deploy the `asset-viewer.war` file as an enterprise application. When you deploy the `asset-viewer.war` file, ensure that the JDK source level is set to 18 for SDK 1.8:
 - a. In the form, select the WAR file, select **Show me all installation options** and parameters so the **Select Installation Options** wizard runs.
 - b. In step 3 of the **Select Installation Options** wizard, ensure that the JDK Source Level is set to 18 for SDK 1.8.
 - c. In step 8 of the **Select Installation Options** wizard, select UnicaPlatformDS as the matching Target Resource. UnicaPlatformDS JNDI resource must be present for Content Integration. For more information on setting up UnicaPlatformDS data source, see *Unica Platform Installation Guide*.
 - d. In step 10 of the **Select Installation Options** wizard, the context root must be set to `/asset-viewer` (all lower case).
3. For your installation to support non-ASCII characters, for example for Portuguese or for locales that require multi-byte characters, add the following arguments to Generic JVM Arguments at the server level.
 - `-Dfile.encoding=UTF-8`
 - `-Dclient.encoding.override=UTF-8`



Note: Navigation tip: Select **Servers > Application Servers > Java and Process Management > Process Definition > Java Virtual Machine > Generic JVM Arguments**. See the WebSphere documentation for additional details.

4. In the **Applications > Enterprise Applications** section of the server, select the WAR file that you deployed, then select **Class loading and update detection** and specify the following properties.
 - For Class loader order, select **Classes loaded with local class loader first (parent last)**.
 - For WAR class loader policy, select **Single class loader for application**.
5. Start your deployment. If your instance of WebSphere is configured to use a JVM version 1.7 or higher, complete the following steps to work around an issue with the time zone database.
 - a. Stop WebSphere.
 - b. Download the Time Zone Update Utility for Java (JTZU).
 - c. Follow the steps provided by the IBM (JTZU) to update the time zone data in your JVM.
 - d. Restart WebSphere.
6. In Websphere Enterprise Applications, select your **Application > Manage Modules > Your Application > Class Loader Order > Classes** loaded with local class loader first (parent last).
7. The recommended minimum heap size for the basic functioning of the application is 512 MB and the recommended maximum heap size is 1024 MB. Complete the following tasks to specify the heap size.
 - a. In WebSphere Enterprise Applications, select **Servers > WebSphere application servers > server1 > Server Infrastructure > Java and Process Management > Process definition > Java Virtual Machine**.
 - b. Set the initial heap size to 512m.
 - c. Set the maximum heap size to 1024m.



Note: See the *WebSphere documentation* for more information about sizing.

For DB2, set `progressiveStreaming = 2` in WebSphere console at following path: **JDBC >Data sources > UnicaPlatformDS > Custom properties**.

Guidelines for deploying Unica Content Integration on JBOSS

You must follow a set of guidelines when you deploy Unica Content Integration on JBoss. Ensure that the version of JBoss meets the requirements that are described in the *Recommended Software Environments and Minimum System Requirements* document. Use the following guidelines when you deploy Unica Content Integration on JBOSS:

Use the following guidelines when you deploy Unica Content Integration product on any supported version of JBOSS:

1. Deploy the `asset-viewer.war` file as an enterprise application. See <https://docs.jboss.org/jbossweb/3.0.x/deployer-howto.html> for instructions on Deploying Web Server Application in JBoss.
2. Complete the following task if your installation must support non-ASCII characters, for example for Portuguese or for locales that require multi-byte characters.

- Edit the following `standalone.conf` script in the bin directory under your JBOSS /bin directory to add to

```
JAVA_VENDOR.
  ◦ -Dfile.encoding=UTF-8
  ◦ -Dclient.encoding.override=UTF-8
  ◦ -Djboss.as.management.blocking.timeout=3600
```

3. Restart the JBOSS server.

Guidelines for deploying Unica Content Integration on Apache Tomcat®

You must follow a set of guidelines when you deploy Unica Content Integration on Apache Tomcat. Ensure that the version of Apache Tomcat meets the requirements that are described in the *Recommended Software Environments and Minimum System Requirements* document. Use the following guidelines when you deploy Unica Content Integration on Apache Tomcat.

1. Deploy the HCL `asset-viewer.war` file as an enterprise application on Tomcat Apache server. Complete the following tasks if your installation must support non-ASCII characters, for example for Portuguese or for locales that require multi-byte characters.
 - a. Edit the `setenv.sh` file for the respective product instances script in the bin directory under your tomcat instances directory to add `-Dfile.encoding=UTF-8 Dclient.encoding.override=UTF-8`.
 - b. Restart Tomcat.
2. If deploying in a production environment, you can add JVM heap setting for that Tomcat instance in `app-one/bin/setenv.sh` file respectively for all the instances.

Unica Content Integration | User Role Creation

The procedure included in this section is not required for the following scenarios:

- It is a fresh installation. The installation process creates this role under default partition (partition1).
- It is an upgrade and only one partition exists in the current system.

In case, it is an upgrade and system contains more than one partition, you must execute the following command manually for each partition, except partition1. The command exists under `<PLATFORM_HOME>\tools\bin` directory inside your Platform installation directory. This creates a Content Integration user role under each partition.

On Windows

```
populateDb.bat -n AssetPicker -p <partition_name>
```

On Unix

```
populateDb.sh -n AssetPicker -p <partition_name>
```


Performance Tuning



Content Integration Framework provides a way to tune the performance of content event processing for the events received from Kafka topic(s). To do so, configuration properties can be managed in following file -

```
ASSET_PICKER_HOME/conf/events/tuning.properties
```

This file is laid down by Unica Platform installer with some default settings. Given below is the list of available properties & their significance. Do not set any property to blank in favor of its default value, instead put a comment sign (#) before it.

- **Kafka consumer tuning** – These properties allow tuning message consumption from Kafka topics.

Property	Description
kafka.consumer.threads.min	<p>Minimum Kafka consumer threads that will always be kept active. Default is 1, if not specified.</p> <p> Note: An additional property <code>{service-name}.kafka.max-consumers</code> must be set up in Platform configuration for each individual service (event source) to determine the concurrency limits for each event source. <code>{service-name}.kafka.max-consumers</code> should ideally be set to the total number of partitions of corresponding topics. The number specified for <code>kafka.consumer.threads.min</code> must be able to cater to the needs of all individual event sources thus configured using additional property in Platform <code>({service-name}.kafka.max-consumers)</code>.</p>
kafka.consumer.threads.max	<p>Maximum Kafka consumer thread count. In case application outruns the <code>kafka.consumer.threads.min</code> threads, a few more threads will be created. This number sets upper limit for overall thread count. Default is 32767, if not specified.</p>
kafka.consumer.threads.priority	<p><code>kafka.consumer.threads.priority</code> Kafka consumer thread priority. Application can have many threads running for handling various things, such as requests coming from user interface, messages coming from Kafka topic etc. Thread priority determines the precedence of one thread over the other when both need CPU at the same time for their execution. This property determines the priority of overall event receipts. Default is 5, if not specified.</p> <ul style="list-style-type: none"> ◦ 1 - Least priority ◦ 10 - Highest priority

kafka.consumer.threads.max-idle-seconds	<p>Maximum idle time after which excessive threads over <code>kafka.consumer.threads.min</code> are disposed if additional threads are no longer active. Default is 60 seconds.</p>
kafka.consumer.max-poll-interval-ms	<p>Consumer poll interval in milliseconds. Default is 300000ms if not specified. This value is used for standard Kafka consumer configuration - <code>max.poll.interval.ms</code>.</p> <p>This setting can be overridden for individual event consumer service using <code>{service-name}.kafka.max.poll.interval.ms</code> additional parameter in Platform configuration.</p>
kafka.consumer.heartbeat-interval-ms	<p>Consumer thread heartbeat interval in milliseconds. This value is used for standard Kafka consumer configuration - <code>heartbeat.interval.ms</code>. Default is 3000ms, if not specified.</p> <p>This setting can be overridden for individual event consumer service using <code>{service-name}.kafka.heartbeat.interval.ms</code> additional parameter in Platform configuration.</p> <p> Note: Refer https://kafka.apache.org/documentation/#consumerconfigs for standard Kafka consumer configurations.</p>
kafka.consumer.session-timeout-ms	<p>Consumer session time out interval in milliseconds. This value is used for standard Kafka consumer configuration - <code>session.timeout.ms</code>. Default is 10000ms, if not specified.</p> <p>This setting can be overridden for individual event consumer service using <code>{service-name}.kafka.session.timeout.ms</code> additional parameter in Platform configuration.</p> <p> Note: Refer https://kafka.apache.org/documentation/#consumerconfigs</p>






[nsumerconfigs](#) for standard Kafka consumer configurations.

- **Event processor tuning** – These properties allow tuning processing of events received via Webhook as well as from Kafka topics.

Property	Description
error.kafka.topic	<p>Name of Kafka topic where event processing errors can be sent in addition to logging into application logs. This is an optional property & there is no default topic name considered if it is not specified.</p> <p>This value can be overridden in Platform configuration for the services listening to Kafka topic for incoming events, using <code>{service-name}.error.kafka.topic</code> additional parameter.</p>

- **Kafka producer tuning** – These properties help for tuning message publishing to Kafka topics.

Property	Description
kafka.producer.batch-size	<p>Message producer batch size in bytes. Default is 16384 (16KB), if not specified. This setting can be overridden for individual message producer service using <code>{service-name}.kafka.batch.size</code> additional parameter in Platform configuration. Value of this property is used for standard Kafka producer configuration – batch.size.</p> <p> Note: Refer https://kafka.apache.org/documentation/#producerconfigs for standard Kafka producer configurations.</p>
kafka.producer.linger-ms	<p>Number of milliseconds to wait for gathering <code>kafka.producer.batch-size</code> bytes of data before sending the complete batch. Default is 0, wherein producer does not wait for complete batch of messages and sends the message immediately. This setting can be overridden for individual message producer service using <code>{service-name}.kafka.linger.ms</code> additional parameter in Platform configuration. Value of this property is used for standard Kafka producer configuration – linger.ms.</p>

	 Note: Refer https://kafka.apache.org/documentation/#producerconfigs for standard Kafka producer configurations.
kafka.producer.compression-type	<p>Compression type for all the data produced by the producer to any output topic. The default is none (i.e. no compression), if commented.</p> <p>Valid values are - <code>none, gzip, snappy, lz4, or zstd</code>.</p> <p>This setting can be overridden for individual message producer service using <code>{service-name}.kafka.compression.type</code> additional parameter in Platform configuration.</p> <p>Value of this property is used for standard Kafka producer configuration – compression.type.</p>  Note: Refer https://kafka.apache.org/documentation/#producerconfigs for standard Kafka producer configurations.

Kafka transaction configuration - Transactions help to prevent duplicate messages in target topics. Set `kafka.transactions.enable` property to `true` to enable transactions. Following prerequisites & limitations must be considered before turning the transactions on –

- Consumer of target topic must ensure to read committed messages only by setting isolation level to “read_committed”. For example, when Content Integration Framework is configured to use transactions for publishing messages on topic “output”, then the subsequent Kafka consumer of “output” topic must use “read_committed” isolation level.
- All topics must belong to the same Kafka cluster if transactions are used. Therefore, when `kafka.transactions.enabled` flag is set to true, Content Integration Framework uses the global Kafka configurations made under top level **Content Integration** node in Platform configuration. Any other system level Kafka configuration is ignored.

`kafka.transactions.enabled` is set to false by default.

Chapter 2. Configuring Unica Content Integration

To use Content Integration in Unica Centralized Offer Management, you must configure it in Unica Platform.

About this task

To configure Unica Content Integration, complete the following steps:

1. Access the Unica Platform application with appropriate credentials.
2. Select **Settings > Configuration**.

Result

The **Configuration** page appears.

3. In the Configuration categories panel, select **Content Integration** to expand the selection.
4. Configure the **navigation** settings. For more information, see [Configuring navigation settings on page 13](#).
5. Configure the **proxy** settings. For more information, see [Configuring proxy settings on page 14](#).

Google Chrome and Microsoft Edge configuration

Recently, the browsers Google Chrome and Microsoft Edge updated a security fix and this security fix affects the access of Unica applications. We have received some issues from our customers like:

- issues with UI
- unable to edit flowcharts
- getting logged out from Unica

These issues are observed due to the change of behavior in browsers after applying the security fix. Applying the security fix automatically enables **Origin-keyed Agent Clusters by default**. If the setting **Origin-keyed Agent Clusters by default** is enabled automatically, it prevents changes in document referrer and domain values so that malicious websites cannot execute any type of impersonation. The setting **Origin-keyed Agent Clusters by default** existed earlier as well, but was not enabled by default.

If you update Google Chrome or Microsoft Edge to the latest version, you will observe the earlier mentioned issues. Because of how Unica is designed and because the Unica suite is deployed over multiple JVMs, it is essential that you disable the **Origin-keyed Agent Clusters by default** setting for Unica to function correctly and to provide a good user experience.

As a solution, we recommend that you perform the steps mentioned in the following Knowledge Base article: https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0107185.



Note: The update to Microsoft Edge browser is very recent and the Knowledge Base article is not yet updated for the Microsoft Edge browser.

The CliffsNotes solution is as follows:

1. Open your browser and navigate to one of the following URLs based on your browser:
 - **Google Chrome:** <chrome://flags/#origin-agent-cluster-default>
 - **Microsoft Edge:** <edge://flags/#origin-agent-cluster-default>
2. From the dropdown of the highlighted parameter **Origin-keyed Agent Clusters by default**, select `Disabled`.
3. At the bottom of the page, click the **Apply Changes** button.
4. Log out of Unica applications, log back in, and verify if everything is working as expected.

Configuring navigation settings

About this task

Configure navigation settings so that Unica Products/Applications. knows the URL and the communication channel required to access the artifacts in the Content Integration application.

To configure the navigation settings of Unica Content Integration, complete the following steps:

1. In the Configuration categories panel, select **Content Integration > navigation**.

Result

The **Settings for 'navigation'** appears.

2. Click **Edit settings**.

Result

The **Edit settings** page for (**navigation**) appears.

3. Provide values for the following fields:

- **httpPort** - The port number for communication in an unsecured connection.
- **httpsPort** - The port number for communication in a secured connection.
- **serverURL** - The URL of the **Unica Content Integration** installation. Browser will communicate via this URL, if there is a proxy between the browser and the server then it should be proxy URL.

If users access **Unica Content Integration** with the Chrome browser, use the fully qualified domain name (FQDN) in the URL. If the FQDN is not used, the Chrome browser cannot access the product URLs.

Default value

`http://<server>:<port>/asset-viewer`



Note: <server> must be lowercase.

- **serverURLInternal** - Specifies the internal URL of the **Unica Content Integration** server. This value is used internally by Unica applications for server to server communications. So it should never be the proxy URL, instead it should be the direct URL.

Default value

`http://<server>:<port>/asset-viewer`



Note: <server> must be lowercase.

4. Click **Save changes**.



Note: If **Unica Content Integration** is clean installed on any version apart from base version (12.1.0)-user needs to manually configure both server url and internal server url.

What to do next

See [Configuring proxy settings on page 14](#).

Configuring proxy settings

About this task

When you configure the proxy server, Unica Content Integration routes the outgoing connections to the target content management system through the configured proxy server.

To configure the proxy settings of Unica Content Integration, complete the following steps:

1. In the Configuration categories panel, select **Content Integration > proxy**.

Result

The **Settings for proxy** appears.

2. Click **Edit settings**.

Result

The **Edit settings** page for (**proxy**) appears.

3. Provide values for the following fields:

- **proxyHostName** - The proxy server host name.



Note: Proxy settings are applied only if you provide an appropriate value for **proxyHostName**.

- **proxyPortNumber** - The port number for communication for the proxy server.
- **proxyType** - The type of communication with the proxy server. Currently only HTTP communication is supported.
- **dataSourceNameForProxyCredentials** - Specify the datasource name that contains the proxy server username and password details.
- **unicaUserForProxyCredentials** - Specify the name of the Unica user that has the specified datasource in the **Data source for credentials** property.

4. Click **Save changes**.

5. Based on the configurations made for `dataSourceNameForProxyCredentials` and `unicaUserForProxyCredentials`, assign the datasource to the user containing the HTTP proxy credentials.

Configuring Kafka connectivity

The Kafka connectivity details in this topic applies to all the Kafka based integrations available with Content Integration Framework. Additionally, if required, these details can be overridden by the system level Kafka configuration. This feature offers the flexibility to integrate with multiple Kafka clusters, providing adaptability based on specific requirements or scenarios.

Kafka config properties and their description are as follows:

Kafka Config Property	Description
Use Kafka (checkbox)	Select or deselect to opt in or opt out of Kafka integration.
Bootstrap servers	Comma separated list of Kafka bootstrap servers (brokers).
Security protocol	The security protocol to be used for Kafka connections.
SASL mechanism	The required SASL mechanism.
Unica user for data source	Unica user, holding the required data sources containing SASL and/or SSL secrets.
Data source name for SASL Jaas credentials	Name of the user data source containing SASL credentials. Applicable only for <code>SASL_*</code> security protocols.
SSL - Trust store location	Absolute path to the trust store (<code>.jks</code>). Applicable only for <code>SSL</code> and <code>SASL_SSL</code> security protocols.
SSL - Data source for truststore password	Name of the user data source containing the trust store password (username from the data source is ignored). Applicable only for <code>SSL</code> and <code>SASL_SSL</code> security protocols.
SSL - Keystore location	Absolute path to the key store (<code>.jks</code>). Applicable only for <code>SSL</code> and <code>SASL_SSL</code> security protocols.
SSL - Data source name for keystore password	Name of the user data source containing the key store password (username from the data source is ignored). Applicable only for <code>SSL</code> and <code>SASL_SSL</code> security protocols.
SSL - Data source name for key password	Name of the user data source containing the key password (username from the data source is ignored). Applicable only for <code>SSL</code> and <code>SASL_SSL</code> security protocols.
Kerberos - Configuration file path	Path to Kerberos configuration file. For example, on Unix, it is generally <code>/etc/krb5.conf</code> and on Windows it can be found at <code>C:/Windows/krb5.ini</code> . Applicable only for <code>SASL_*</code> security protocols with KERBEROS SASL mechanism.

Kafka Config Property	Description
Kerberos - Keytab file path	Path to Kerberos keytab file. Applicable only for <code>SASL_*</code> security protocols with KERBEROS SASL mechanism.
Kerberos - Principal	Kerberos principal name having access to kafka service. Applicable only for <code>SASL_*</code> security protocols with KERBEROS SASL mechanism.
Kerberos - Service Name	Kerberos service Name for kafka (as per <code>sasl.kerberos.service.name</code> configuration for Kafka cluster). Applicable only for <code>SASL_*</code> security protocols with KERBEROS SASL mechanism.

The purpose of the **Use Kafka** checkbox is to enhance the handling of events received through Webhooks. Enabling this checkbox enhances the fault tolerance of Webhook event processing by leveraging certain Kafka topics to cache and route events. This approach safeguards against event loss in situations of unexpected system shutdowns. The Kafka topic responsible for routing Webhook events needs to adhere to the provided naming convention. To align with Kafka's topic naming policies, any spaces within the `{system-identifier}` should be substituted with hyphens (-).

```
{system-identifier}.in
```

For examples, see the **Autosync Configurations** topic in the *Unica Centralized Offer Management Administrator's Guide*.

Adding the Content Integration User role to a user or a user group

To access Unica Content Integration, assign the Content Integration User role to users or user groups. For more information on assigning roles to users or user groups, see *Unica Platform Administrator Guide*.

Logs configuration

View the Unica Content Integration logging configuration in the `log4j2.xml` file available in the `AssetPicker/conf/logging` folder inside Platform home. The logs produced by Unica Content Integration are placed within the `AssetPicker/logs` folder inside Platform home.

The `com.unica` logger declared inside `log4j2.xml` file is used for setting up loggers originating from Platform activities, such as configuration reads, user authorization etc.

All the loggers except "com.unica" are used to log the core activities of Unica Content Integration.

The default log level, in both cases, is set to `WARN`. This should suffice for any troubleshooting issues that might arise during installation.

For more information on configuring the `log4j2.xml` file, see the official documentation for *Apache Log4j*.

Chapter 3. Configuring a third-party CMS for integration with Unica Content Integration

You can configure a third-party CMS for integration with Unica Content Integration. When you create partitions in Unica Centralized Offer Management, an entry for Content Integration resides in each partition. For example, if Unica Centralized Offer Management has three partitions, you can configure Content Integration for all the partitions of Unica Centralized Offer Management.

About this task

To create partitions in Unica Centralized Offer Management, see the *Unica Centralized Offer Management Installation Guide*.

To configure a third-party CMS for integration with Unica Content Integration, complete the following steps:

1. Select **Settings > Configuration**.

Result

The **Configuration** page appears.

2. From the **Configuration categories** panel, select **Offer > partitions > partition <n> > Content Integration > Data Sources > (System Configuration Template)**.

- **Offer** - The name provided to Unica Centralized Offer Management application.
- **partition <n>** - Here <n> represents the number of partitions. For example, if Unica Centralized Offer Management has two partitions, then **partition <n>** can be **partition 1** or **partition 2**.

Result

The **Create category from template** page for **(System Configuration Template)** appears.

3. Provide values for the following fields:

- **New category name** - An appropriate name to identify the CMS that you want to configure. For example, `HCL DX Of Adobe Experience Manager`.



Note: From 12.1.1 onwards, the **New category name** value will be used for identifying the CMS in all applicable Unica products, like Centralized Offer Management, Plan, and Deliver. Earlier, the value of the **System Identifier** parameter was used.

- **System Identifier** - A predefined system identifier. Each system is assigned a unique identifier in Unica Content Integration. System identifiers are case sensitive and must match with the one stipulated for the respective system.



Note: The **System Identifier** name must match the `systemId` used in the respective Content Integration plugin. For more information on `systemID`, see *Unica Content Integration Developer's Guide*.

- **User credentials** - Specify the user credential selection strategy for the target system. The available options are:

- **Default user** - If you select **Default user**, the data source for **Default user for credential** account is used while calling APIs of the target CMS.
- **Logged-in user** - If you select **Logged-in user**, the data source of Unica's signed-in user account is used while calling APIs of the target CMS.
- **Hierarchical** - If you select **Hierarchical**, the data source of Unica's signed-in user account is considered first. If the Unica's signed-in user account does not contain a data source, the **Default user for credential** will be considered.
- **Default user for credential** - The default Marketing Platform user having the target CMS credentials. The system uses this configuration when:
 - the User Credential strategy is the Default User.
 - the User Credential strategy is Hierarchical, but the logged-in user does not have the datasource associated. In such a case, the default user will be considered for credential selections.
- **Data source name for credentials** - The data source assigned to the Marketing Platform account. This data source authenticates the target system's REST API, database, etc.
- **Anonymous Content Access** - Select **Yes** if the target system accepts anonymous access to content or select **No** if the target system does not accept anonymous access to content.

**Note:**

- Content Integration framework works with the protected APIs of the target CMS system. However, in this release, the target CMS should be configured for allowing anonymous guest user access to the content URLs.
- **Additional parameters** - From version 12.1.0.4 onwards, additional key-value parameters can be specified as per the requirement by each individual system. It can be left blank if no such parameters are required by the respective system.

Each key-value pair must be specified on separate line. Key and value must be separated by a colon, followed by a space. For example:

```
key1: value1
key2: value2
```



Note: If you have not provided credentials for the target system, or if the provided credentials are incorrect, the target system will decline the connection request. Before saving the changes, ensure that you have provided the credentials and that they are accurate.

4. Click **Save changes**.

Result

A new entry, for example `<CMSName>`, appears under **Data sources** for the configured CMS. Expand the entry to see the following entries:

- `<CMSName> | HTTP Gateway` - contains a configuration.
- `<CMSName> | HTTP Gateway | REST` - contains a configuration.
- `<CMSName> | QOS` - does not contain configurations.
- `<CMSName> | QOS | Retry Policy` - contains configurations.

5. To configure **HTTP Gateway**, complete the following steps:

- a. In the **Configuration properties** pane, expand **<CMSName>** and select **HTTP Gateway**.

Result

The **Settings for 'HTTP Gateway'** page appears.

- b. Select **Edit settings**.

Result

The **(HTTP Gateway) Edit Settings** page appears.

- c. For the **Base URL** field, specify the base location of the target system. Example: `http://`

`<hostname> : <port-number>`.

- d. The **Bypass proxy** field is available from version 12.1.0.4, select **Yes** if you want to bypass proxy while connecting to the respective system. By default the value of this field is set to **No**, which means if the proxy server is configured, all the connections made to respective system will go through the proxy.

- e. The **Content Base URL** field is available from version 12.1.0.4, leave it blank if the content is hosted under the same base URL where the respective system is running.

- f. Click **Save changes**.

6. To configure **HTTP Gateway | REST**, complete the following steps:

- a. In the **Configuration properties** pane, expand **HTTP Gateway** and select **REST**.

Result

The **Settings for 'REST'** page appears.

- b. Select **Edit settings**.

Result

The **(REST) Edit Settings** page appears.

- c. For the **Authentication Type** field, select from one of the following values:

- **Basic**

Select **Basic** if the target system's API needs HTTP Basic Authentication. Credentials need to be set up accordingly under user's data source.

- **Bearer Token**

Select **Bearer Token** if target system's API needs Bearer token authentication. (Bearer token must be configured in the password field of the desired user datasource. Username set up in datasource is ignored for this authentication type.)

- **Unica Token**

The **Unica Token** option can be used if target system belongs to the Unica Product Suite. The **Unica Token** adds the necessary API token request headers during HTTP invocation. (User datasource assignment is not required for this authentication type since tokens are system generated.)

- **None**

Select **None**, if the target system APIs are not protected by authentication, or if the target system uses authentication mechanism other than the supported ones and authentication is completely handled by the plugin implementation.

d. Click **Save changes**.

7. To configure **QOS | Retry Policy**, complete the following steps:

a. In the **Configuration properties** pane, expand **QOS** and select **Retry Policy**.

Result

The **Settings for 'Retry Policy'** page appears.

b. Select **Edit settings**.

Result

The **(Retry Policy) Edit Settings** page appears.

c. Provide values for the following fields:

- **Retry count**

Specify the number of times Content Integration should attempt to access target system.

- **Initial delay**

Specify the number of milliseconds to wait before making next attempt after failed access.

- **Delay multiplier**

Specify if the delay interval between subsequent attempts should be multiplied. Set this to 1 to keep a consistent delay between each attempt. Setting this value to greater than 1 increases the delay interval between each subsequent attempt. Similarly, setting this value to less than 1 decreases delay interval between subsequent attempts.

Example

If you provide the following values:

- **Retry count** - 3

- **Initial delay** - 1000

- **Delay multiplier** - 1.2

The first attempt will be made immediately to access the target system. If the first attempt fails, Content Integration will wait for 1000 ms (1 second) before making the second attempt. And if the second attempt also fails, Content Integration will wait for $1000 * 1.2$ milliseconds (1.2 seconds) before making the third attempt. And if the third attempt also fails, Content Integration will stop making any further attempt and terminate the operation.



Note:

The Content Integration framework understands the settings explained earlier and automatically takes care of authentication, QOS, and other settings when the respective plugin is implemented using



RESTful approach. For more information on RESTful approach, see the *Unica Content Integration Developer's Guide*.

The earlier mentioned properties can still be used for non-RESTful implementations. The Plugin developer can access all the settings programmatically and use them for plugin implementation. For example, you can use Base URL to configure any URL, or location, as per the target system type. This allows you to specify `jdbc:oracle:thin:@localhost:1521:xe` as the Base URL, if the target system is a database and it is used inside the respective Content Integration plugin, when attempting a database connection.

d. Click **Save changes**.

What to do next



Note:

- From version 12.1.0.4 onwards, after every 30 mins system will reload the configuration. Hence, there is no need to restart the Content Integration application every time a change is made in Platform configurations. You can set the time intervals for reloading configurations, for more information see [Setting configuration refresh interval JVM argument on page 1](#)
- Updates to user data source does not require a restart. Changes in user data sources become effective as per configuration refresh schedule. See [Setting configuration refresh interval JVM argument on page 1](#) to learn more.
- Configure the out-of-the-box integrations like Adobe Experience Manager (AEM), HCL WCM, HCL DX, HCL Commerce & Microsoft Azure (for cognitive services). For more information, see [Configuring the out-of-the-box integrations offered by Content Integration on page 24](#).

Webhook Security

Any invocation to Content Integration Framework's Webhooks from external systems are protected by means of API security filters in Unica Platform. In addition to Unica's API security filters, certain environments, such as **IBM Security Verify Access** (formerly known as **IBM Security Access Manager**) provide their own security measures for application access. Hence, to enable Webhook invocation from external systems running outside the realm of IBM Security Verify Access, appropriate permissions must be set up for respective Webhook URL.

Deploying Unica Content Integration in a distributed environment

To deploy Unica Content Integration as a standalone entity without Centralized Offer Management in a distributed environment, where Platform resides on one machine and Unica Content Integration on another machine, ensure the following points:

- While running the installer, you must select the Platform Utilities checkbox along with Asset Picker.
- During registration, select the following checkboxes.

- Manual db population
- Run the manual configuration

Verifying Content Integration configuration in Unica Centralized Offer Management

After configuring Content Integration, verify the configuration on Unica Centralized Offer Management to see if the configuration is successful or not.

About this task



Note: The verification steps are specific to Unica Centralized Offer Management.

To verify the Content Integration configuration, complete the following steps:

1. Select **Settings > Offer settings**.

Result

The **Offer Settings** page appears.

2. Select **Offer template definitions**.

Result

The **Offer template definitions** page appears.

3. In the **Offer template definitions** page, click **+ Add new template**.

Result

The **Metadata** section of the **Add offer template details** page appears.

4. In the **Metadata** section, complete the following steps:

- a. In the Basic options, provide values for the following fields:

- **Template display name** - Mandatory. An appropriate name for the custom template.
- **Select template icon** - Mandatory. Select an appropriate icon from the available list. The selected icon appears beside the template name in the listing page.
- **Security policy** - Mandatory. Select from the list of policies.

- b. Click **Next**.

Result

The **Offer attributes** section appears.

5. In the **Offer attributes** section, complete the following steps:

- a. The **Offer attributes** section is divided into four panel:

- **Available attributes**
- **Parametrized attributes**
- **Static attributes**
- **Hidden attributes**

b. To pick **Custom** attributes as a part of the template, select **Custom**, and drag-and-drop an attribute of type **Picker - URL** from the attributes list to the **Static attributes** panel. You can also search the **Custom** attributes using the search bar.

c. Click **Next**.

Result

The **Default values** section appears.

d. In the **Static attributes** drop-down panel, click **Browse**.

Result

The **Content Integration** dialog appears.

6. If you see the artifacts from the configured URL, it is an indication that the configuration is successful. If you see an error, it is an indication that the configuration was unsuccessful.

Chapter 4. Configuring the out-of-the-box integrations offered by Content Integration

Content Integration offers the following out-of-the-box integrations: Adobe Experience Manager (AEM), HCL WCM, HCL Digital Experience (DX) 95_CF205, HCL Commerce & Microsoft Azure (for cognitive services).

Before you begin

See [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

About this task

To configure Adobe Experience Manager (AEM), HCL WCM, HCL DX, HCL Commerce & Microsoft Azure, complete the following steps:

1. Setting up Adobe Experience Manager

- a. Select **Settings > Configuration**.

Result

The **Configuration** page appears.

- b. From the **Configuration categories** panel, expand **Offer > partitions > partition <n> > Content Integration > Data Sources > Adobe Experience Manager**.

Result

The **Create category from template** for the **Adobe Experience Manager** appears.

- c. Provide values for the following fields and click **Save changes**:

- **New category name** - Specify an appropriate name to identify the new CMS. For example, AEM.
- **System Identifier** - AEM.
- **User credentials**
- **defaultUserCredentials**
- **Data source name for credentials**
- **Anonymous Content Access** - When this field is set to **No**, then under CMS configuration, the content preview will not show the download icon. When set to **Yes**, it will show the download icon.

For more information about the fields, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

- d. In the **Configuration properties** pane, expand **AEM** and select **HTTP Gateway**.

Result

The **Settings for 'HTTP Gateway'** page appears.

- e. Select **Edit settings**.

Result

The **(HTTP Gateway) Edit Settings** page appears.

- f. For the AEM **Base URL** field, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).
- g. Click **Save changes**.
- h. In the **Configuration properties** pane, expand **HTTP Gateway** and select **REST**.

Result

The **Settings for 'REST'** page appears.
- i. Select **Edit settings**.

Result

The **(REST) Edit Settings** page appears.
- j. For the **Authentication Type** field, select the value `Basic`. If your AEM instance is set up to use a different authentication method, use a different value. For more information about the available values, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).
- k. Click **Save changes**.
- l. In the **Configuration properties** pane, expand **QOS** and select **Retry Policy**.

Result

The **Settings for 'Retry Policy'** page appears.
- m. Select **Edit settings**.

Result

The **(Retry Policy) Edit Settings** page appears.
- n. For the **Retry policy** configurations, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).
- o. Click **Save changes**.

2. Publish status filter for Adobe Experience Manager



Note: If you plan to upgrade Unica Content Integration, ensure that you create `custom-plugin-services.yml`, and store the customized configurations. After update two files will be created:

- `plugin-services.yml` (for out of the box plugins)
- `custom-plugin-services.yml` (for custom plugins)

Installer will never overwrite `custom-plugin-services.yml` file since it will contain the service declarations for custom plugins as well as customization for out of the box services.

- a. In the `plugin-services.yml` file, available under `<ASSET_PICKER_HOME>/conf` directory, there exists a configuration for AEM under "systems" section. Under "AEM", the `contentFilters` configuration parameter contains a child parameter named `publishStatus`. This parameter filters search items by their published

status. The parameter accepts only one of the following values: `Published` or `Unpublished`. Provide the required value.

- b. If you do not provide a value or if you provide an incorrect value, AEM returns matching content, irrespective of its published status. This is the default behavior because the `publishStatus` parameter, by default, is commented out.

3. Setting up WCM

- a. Select **Settings > Configuration**.

Result

The **Configuration** page appears.

- b. From the **Configuration categories** panel, expand **Offer > partitions > partition <n> > Content Integration > Data Sources > WCM**.

Result

The **Create category from template** for the **WCM** appears.

- c. Provide values for the following fields and click **Save changes**:

- **New category name** - Specify an appropriate name to identify the new CMS. For example, `wcm`.
- **System Identifier** - `wcm`.
- **User credentials**
- **defaultUserCredentials**
- **Data source name for credentials**
- **Anonymous Content Access** - Set this to **Yes**.

For more information about the fields, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

- d. In the **Configuration properties** pane, expand **WCM** and select **HTTP Gateway**.

Result

The **Settings for 'HTTP Gateway'** page appears.

- e. Select **Edit settings**.

Result

The **(HTTP Gateway) Edit Settings** page appears.

- f. For the WCM **Base URL** field, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

- g. Click **Save changes**.

- h. In the **Configuration properties** pane, expand **HTTP Gateway** and select **REST**.

Result

The **Settings for 'REST'** page appears.

- i. Select **Edit settings**.

Result

The **(REST) Edit Settings** page appears.

- j. For the **Authentication Type** field, select the value `Basic`. If your WCM instance is set up to use a different authentication method, use a different value. For more information about the available values, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).
- k. Click **Save changes**.
- l. In the **Configuration properties** pane, expand **QOS** and select **Retry Policy**.

Result

The **Settings for 'Retry Policy'** page appears.

- m. Select **Edit settings**.

Result

The **(Retry Policy) Edit Settings** page appears.

- n. For the **Retry policy** configurations, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).
- o. Click **Save changes**.

4. Publish status filter for WCM



Note: If you plan to upgrade Unica Content Integration, ensure that you back up the file `plugin-services.yml`, if you have customized the files. The installer overwrites the file and your customizations will be lost.

- a. In the `plugin-services.yml` file, available under `<ASSET_PICKER_HOME>/conf` directory, there exists a configuration for WCM under `systems` section. Under WCM, the `contentFilters` configuration parameter contains a child parameter named `publishStatus`. This parameter filters search items by their published status. The parameter accepts only one of the following values: `DRAFT`, `PUBLISHED`, or `EXPIRED`. Provide the required value.
- b. If you do not provide a value or if you provide an incorrect value, WCM returns matching content, irrespective of its published status. This is the default behavior because the `publishStatus` parameter, by default, is commented out.
- c. If you activate the `publishStatus` parameter or modify its value, an application restart is mandatory.

5. Setting up HCL Digital Experience (DX-CORE)

- a. Select **Settings > Configuration**.

Result

The **Configuration** page appears.

- b. From the **Configuration categories** panel, expand **Offer > partitions > partition <n> > Content Integration > Data Sources > HCL Digital Experience (DX)**.

Result

The **Create category from template** for the **HCL Digital Experience (DX)** appears.

c. Provide values for the following fields and click **Save changes**:

- **New category name** - Specify an appropriate name to identify the new CMS. For example, HCL - DX.
- **System Identifier** - DX-CORE.
- **User credentials** - Default user
- **defaultUserCredentials** - asm_admin
- **Data source name for credentials** - DX_NEW_DS
- **Anonymous Content Access** - Set this to **Yes**.
- **Additional Parameters** - async: true numberOfThreads: 4 dxMediaSystemIdentifier: DX

For more information about the fields, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

d. Click **Save changes**.

e. Ensure that the **Data source name for credentials** field is blank.

f. In the **Configuration properties** pane, expand **HCL - DX** and select **HTTP Gateway**.

Result

The **Settings for 'HTTP Gateway'** page appears.

g. Select **Edit settings**.

Result

The **(HTTP Gateway) Edit Settings** page appears.

h. Provide a value for the **Base URL** field. If required, the **Base URL** can also be used for authenticating the HCL DX - Media Library credentials. See [Step 6](#) for configuring HCL DX - Media Library. Click **Save changes**.

i. In the **Configuration properties** pane, expand **HTTP Gateway** and select **REST**.

Result

The **Settings for 'REST'** page appears.

j. Select **Edit Settings**.

- **Base URL** - `http://<dx_host>:<dx_port>/dx`, where `<dx_host>:<dx_port>` is the host name and port number on which DX is configured.
- **contentBaseUrl** - `http://<dx_host>:<dx_port>`, where `<dx_host>:<dx_port>` is the host name and port number on which DX is configured.
- **bypassProxy** - No

k. For the **Authentication Type** field, select the value `None`. If the HCL - DX instance is set up to use a different authentication method, use a different value. For more information about the available values, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

l. Click **Save changes**.

m. In the **Configuration properties** pane, expand **QOS** and select **Retry Policy**.

Result

The **Settings for 'Retry Policy'** page appears.

- n. Select **Edit settings**.

Result

The **(Retry Policy) Edit Settings** page appears.

- o. For the **Retry policy** configurations, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

- p. Click **Save changes**.

q.

- r. Disable all the security checks for the DX_CCORE webhook url from **Unica Platform > Security > API management > Unica Content Integration > Create a New Security Filter > /webhook/DX-CORE/events/***



Note: Make to give the correct system Identifier. Default value is DX-CORE.

- **API URI** - /webhook/DX-CORE/events/*
- **Block API access** - Disabled
- **Secure API access over HTTPS** - Enabled/Disabled as per your environment requirements.
- **Required authentication for API access** - Disabled
- **Authentication mode** - Not Required
- **Data source credential holder** - Not Required
- **Data source** - Not Required

6. Setting up HCL DX - Media Library

- a. Select **Settings > Configuration**.

Result

The **Configuration** page appears.

- b. From the **Configuration categories** panel, expand **Offer > partitions > partition <n> > Content Integration > Data Sources > (CMS Configuration Template)**.

Result

The **Create category from template** for the **(CMS Configuration Template)** appears.

- c. Provide values for the following fields and click **Save changes**:
- **New category name** - Specify an appropriate name to identify the new CMS. For example, **HCL DX - Media Library**.
 - **System Identifier** - **DX**.
 - **User credentials** - Default user
 - **defaultUserCredentials** - **asm_admin**
 - **Data source name for credentials** - **DX_NEW_DS**

- **Anonymous Content Access** - Set this to **Yes**.
- **additionalParameters** - Provide a ringapi URL for authenticating HCL DX - Media Library. For example, `apiLogonUrl: http://<domain-name>:port-number/dx/api/core/v1/auth/login. .`

For more information about the fields, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

- d. Click **Save changes**.
- e. Ensure that the **Data source name for credentials** field is blank.
- f. In the **Configuration properties** pane, expand **HCL DX - Media Library** and select **HTTP Gateway**.

Result

The **Settings for 'HTTP Gateway'** page appears.

- g. Select **Edit settings**.

Result

The **(HTTP Gateway) Edit Settings** page appears.

- h. Provide an appropriate for the **Base URL** field. Click **Save changes**.

- i. In the **Configuration properties** pane, expand **HTTP Gateway** and select **REST**.

Result

The **Settings for 'REST'** page appears.

- j. Select **Edit Settings**.

- **Base URL** - Please contact DX admin to get these details
- **contentBaseUrl** - Please contact DX admin to get these details.
- **bypassProxy** - No

- k. For the **Authentication Type** field, select the value `None`. If `HCL DX - Media Library` instance is set up to use a different authentication method, use a different value. For more information about the available values, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

- l. Click **Save changes**.

- m. In the **Configuration properties** pane, expand **QOS** and select **Retry Policy**.

Result

The **Settings for 'Retry Policy'** page appears.

- n. Select **Edit settings**.

Result

The **(Retry Policy) Edit Settings** page appears.

- o. For the **Retry policy** configurations, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

- p. Click **Save changes**.

7. Setting up Azure

- a. Select **Settings > Configuration**.

Result

The **Configuration** page appears.

- b. From the **Configuration categories** panel, expand **Offer > partitions > partition <n> > Content Integration > Data Sources > Azure**

Result

The **Create category from template** for the **Azure** appears.

- c. Provide values for the following fields and click **Save Changes**

- **New category name** - Azure
- **System Identifier** - Azure
- **User credentials**
- **defaultUserCredentials**
- **Data source name for credentials**



Note: Data source password must contain the valid Azure subscription key. Username of the data source can be set to **user**.

- **Anonymous Content Access**
- **Additional Parameters**

- d. Click Save changes

- e. In the **Configuration properties** pane, expand **Azure** and select **HTTP Gateway**.

Result

The settings for **HTTP Gateway** appear.

- f. Select **Edit Settings**.

- **Base URL** - End point of deployed cloud azure service. For Example: https://<xxxx>.cognitiveservices.azure.com
- **contentBaseUrl**
- **bypassProxy**

- g. Click **Save changes**.

- h. In the **Configuration properties** pane, expand **HTTP Gateway** and select **REST**.

Result

The **Settings for REST** page appears

- i. Select **Edit settings**

Result

The **(REST) Edit Settings** page appears.

j. For the **Authentication Type** field, select the value **None**

k. Click **Save changes**.

8. Setting up HCL Commerce

a. Select **Settings > Configuration**.

Result

The **Configuration** page appears.

b. From the **Configuration categories** panel, expand **Offer > partitions > partition <n> > Content Integration > Data Sources > HCL Commerce**

Result

The **Create category from template** for the **HCL Commerce** appears.

c. Provide values for the following fields and click **Save Changes**

- **New category name** - Specify an appropriate name to identify the new CMS. For example, Emerald.
- **System Identifier** - Commerce Emerald.



Note: Multiple stores can be onboarded by prefixing the identifier with **Commerce**, followed by a space and the store name. Example: Commerce Sapphire, Commerce Store 1, Commerce Store 2 and so on.

- **User credentials**
- **defaultUserCredentials**
- **Data source name for credentials**
- **Anonymous Content Access**
- **Additional Parameters** - Each parameter should be separated by new line. Make sure to add a space after colon [:] while editing the values. Contact commerce admin to obtain the required values.
 - storeId: **11**
 - contractId: **11005**
 - langId: **-1**
 - currency: **USD**

d. Click **Save changes**

e. In the **Configuration properties** pane, expand **HTTP Gateway** and select **REST**.

Result

The **Settings for REST** page appears.

Select Edit settings.

f. Select **Edit settings**.

Result

The **(REST) Edit Settings** page appears.

- g. For the **Authentication Type** field, select the value **None**.
- h. Click **Save changes**.



Note: For commerce products, deep search is not supported at category level. Commerce is looking into this limitation (ticket no: HC-8872) and it will be addressed in future versions.

9. Setting up Snap-CAST

- a. Select **Settings > Configuration**.

Result

The **Configuration** page appears.

- b. From the **Configuration categories** panel, expand **Offer > partitions > partition <n> > Content Integration > Data Sources > Snap-CAST**.

Result

The **Create category from template** for the **Snap-CAST** appears.

- c. Provide values for the following fields and click **Save changes**:

- **New category name** - Specify an appropriate name to identify the new CMS. For example, *Snap-CAST*.
- **System Identifier** - *Snap-CAST*.
- **User credentials**
- **defaultUserCredentials**
- **Data source name for credentials**
- **Anonymous Content Access** - When this field is set to **No**, then under CMS configuration, the content preview will not show the download icon. When set to **Yes**, it will show the download icon.
- **Additional Parameters** - You must configure the following additional key-value parameters. The value for the key-value parameters are examples and can be customized as per your requirement:

```
commandCode: "InterDependentTaggingRequest"
dbSchemaName: "GigBazaar"
version: "v1_0_0_1"
textType: 1
adInputId: 1
score: 100
textSourceOrigin: "Unica Generic"
textProcessorService: "Snap-CAST"
imageSourceOrigin: "Unica Generic"
imageProcessorService: "AWS.Rekognition v1"
blocking: 1
```

If you face issues with the key-value parameters, contact your Snap-CAST service provider.

Asset picket will get notified by the user event with user data which will in turn send user data to journey.

- d. In the **Configuration properties** pane, expand **Snap-CAST** and select **HTTP Gateway**.

Result

The **Settings for 'HTTP Gateway'** page appears.

- e. Select **Edit settings**.

Result

The **(HTTP Gateway) Edit Settings** page appears.

- f. For the Snap-CAST **Base URL** field, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

- g. Click **Save changes**.

- h. In the **Configuration properties** pane, expand **HTTP Gateway** and select **REST**.

Result

The **Settings for 'REST'** page appears.

- i. Select **Edit settings**.

Result

The **(REST) Edit Settings** page appears.

- j. For the **Authentication Type** field, select the value `None`. If your Snap-CAST instance is set up to use a different authentication method, use a different value. For more information about the available values, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

- k. Click **Save changes**.

- l. In the **Configuration properties** pane, expand **QOS** and select **Retry Policy**.

Result

The **Settings for 'Retry Policy'** page appears.

- m. Select **Edit settings**.

Result

The **(Retry Policy) Edit Settings** page appears.

- n. For the **Retry policy** configurations, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

- o. Click **Save changes**.

10. Setting up GoogleVision

- a. Select **Settings > Configuration**.

Result

The **Configuration** page appears.

- b. From the **Configuration categories** panel, expand **Offer > partitions > partition <n> > Content Integration > Data Sources > GoogleVision** .

Result

The **Create category from template** for the **GoogleVision** appears.

c. Provide values for the following fields and click **Save changes**:

- **New category name** - Specify an appropriate name to identify the new CMS. For example, `GoogleVision`.
- **System Identifier** - `GoogleVision`.
- **User credentials**
- **defaultUserCredentials**
- **Data source name for credentials**
- **Anonymous Content Access** - When this field is set to **No**, then under CMS configuration, the content preview will not show the download icon. When set to **Yes**, it will show the download icon.
- **Additional Parameters** - You must configure the following additional key-value parameters. The value for the key-value parameters are examples and can be customized as per your requirement:

```
LandmarkAnnotationsScoreThreshold: 0.50
LabelAnnotationsScoreThreshold: 0.50
LocalizedObjectAnnotationsScoreThreshold: 0.50
LogoAnnotationsScoreThreshold: 0.50
```

If you do not configure the above parameters, `0.80` will be considered as the default value.

For more information about the fields, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

d. In the **Configuration properties** pane, expand **GoogleVision** and select **HTTP Gateway**.

Result

The **Settings for 'HTTP Gateway'** page appears.

e. Select **Edit settings**.

Result

The **(HTTP Gateway) Edit Settings** page appears.

f. For the GoogleVision **Base URL** field, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

g. Click **Save changes**.

h. In the **Configuration properties** pane, expand **HTTP Gateway** and select **REST**.

Result

The **Settings for 'REST'** page appears.

i. Select **Edit settings**.

Result

The **(REST) Edit Settings** page appears.

j. For the **Authentication Type** field, select the value `None`. If your GoogleVision instance is set up to use a different authentication method, use a different value. For more information about the available values, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

k. Click **Save changes**.

l. In the **Configuration properties** pane, expand **QOS** and select **Retry Policy**.

Result

The **Settings for 'Retry Policy'** page appears.

m. Select **Edit settings**.

Result

The **(Retry Policy) Edit Settings** page appears.

n. For the **Retry policy** configurations, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

o. Click **Save changes**.

11. Setting up GoogleLanguage

a. Select **Settings > Configuration**.

Result

The **Configuration** page appears.

b. From the **Configuration categories** panel, expand **Offer > partitions > partition <n> > Content Integration > Data Sources > GoogleLanguage**.

Result

The **Create category from template** for the **GoogleLanguage** appears.

c. Provide values for the following fields and click **Save changes**:

- **New category name** - Specify an appropriate name to identify the new CMS. For example, `GoogleLanguage`.
- **System Identifier** - `GoogleLanguage`.
- **User credentials**
- **defaultUserCredentials**
- **Data source name for credentials**
- **Anonymous Content Access** - When this field is set to **No**, then under CMS configuration, the content preview will not show the download icon. When set to **Yes**, it will show the download icon.
- **Additional Parameters**

For more information about the fields, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

d. In the **Configuration properties** pane, expand **Snap-CAST** and select **HTTP Gateway**.

Result

The **Settings for 'HTTP Gateway'** page appears.

e. Select **Edit settings**.

Result

The **(HTTP Gateway) Edit Settings** page appears.

f. For the GoogleLanguage **Base URL** field, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

g. Click **Save changes**.

h. In the **Configuration properties** pane, expand **HTTP Gateway** and select **REST**.

Result

The **Settings for 'REST'** page appears.

i. Select **Edit settings**.

Result

The **(REST) Edit Settings** page appears.

j. For the **Authentication Type** field, select the value `Basic`. If your GoogleLanguage instance is set up to use a different authentication method, use a different value. For more information about the available values, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

k. Click **Save changes**.

l. In the **Configuration properties** pane, expand **QOS** and select **Retry Policy**.

Result

The **Settings for 'Retry Policy'** page appears.

m. Select **Edit settings**.

Result

The **(Retry Policy) Edit Settings** page appears.

n. For the **Retry policy** configurations, see [Configuring a third-party CMS for integration with Unica Content Integration on page 17](#).

o. Click **Save changes**.

12. Setting up Mailchimp

a. Select **Settings > Configuration**.

Result

The **Configuration** page appears.

b. From the **Configuration categories** panel, expand **Unica Product > Content Integration > dataSources > Mailchimp**.

Result

The **Settings for Mailchimp** page appears.

c. Provide values for the following fields and click **Save changes**:

- **System Identifier** - Mailchimp.
- **User credentials** - Default User

- **defaultUserCredentials** - asm_admin
- **Data source name for credentials** - Name added in this field should be same as user datasouce name. Credentials will be user name - user (all lower case) and for password refer -
- **Anonymous Content Access**

d. In the **Configuration properties** pane, expand **Mailchimp** and select **httpGateway**.

Result

The **Settings for 'httpGateway'** page appears.

e. Select **Edit settings**.

Result

The **(httpGateway) Edit Settings** page appears.

f. For the Mailchimp **Base URL** field, the URL comprises of datasource followed by mailchimp login. For example - `https://<data_center>.api.mailchimp.com/3.0/`. Refer

g. Click **Save changes**.

h. In the **Configuration properties** pane, expand **HTTP Gateway** and select **REST**.

Result

The **Settings for 'REST'** page appears.

i. Select **Edit settings**.

Result

The **(REST) Edit Settings** page appears.

j. For the **Authentication Type** field, select the value `Basic`. If your Mailchimp instance is set up to use a different authentication method, use a different value.

k. Click **Save changes**.

l. Disable all the security checks for the mailchimp webhook url from **Unica Platform > Security > API**

management > Unica Content Integration > Create a New Security Filter > /webhook/Mailchimp/events/*

- **API URI** - /webhook/Mailchimp/events/*
- **Block API access** - Disabled
- **Secure API access over HTTPS** - Enabled/Disabled as per your environment requirements.
- **Required authentication for API access** - Disabled
- **Authentication mode** - Not Required
- **Data source credential holder** - Not Required
- **Data source** - Not Required

Mailchimp Integration with Unica

To integrate Mailchimp application with Unica system, complete the following list of procedures:

Signing Up and Signing into Mailchimp account

- i. Access the following URL:

<https://login.mailchimp.com/>

- ii. To create an account, use either your official email address, Gmail account or Facebook. If required, you can continue with your Google account or Apple account.

API key generation

- i. For generating API key, navigate to **Profile > Extra > API keys**.
- ii. The screen will have a link that will help in creating a key.
- iii. Click **Create a Key**.
- iv. Your API key will be generated.
- v. User can generate multiple API keys and integrate them with different data sources in Platform.



Note: This key will be the password while setting up mailchimp data source in platform.

Create data source in Platform (Mailchimp)

[Configuring the out-of-the-box integrations offered by Content Integration on page 24](#)

Set Audience

After Mailchimp account is created successfully, user can set one or multiple audiences. This is an optional step, user can continue with existing audiences. For setting Audience follow the below steps:

- i. Navigate to **Audience > Manage Audience > settings**.
- ii. Click **Audience Fields and *|MERGE|* tags**.
- iii. Provide values to the visible fields
- iv. Click on **Add a Field** for adding new fields (if required)

Create Webhook

- i. Before creating a webhook security checks needs to be disabled. Refer
- ii. Navigate to **Manage Audience > settings**.
- iii. Click **Webhook**
- iv. Add URL in **Configured webhooks** fields
- v. Click **Create New Webhook**



Note: The added url (https://<unica-domain>/asset-viewer/api/AssetPicker/webhook/Mailchimp/events/webhook_listener) should be publicly available on internet. Update the Unica domain as per your environment.

- vi. Select all the available option
- vii. Click **Update**

Add Subscriber (Optional)

- i. Navigate to **Manage Audience > Add a subscriber**.
- ii. Provide values for the following fields:
 - Email Address
 - First Name
 - Last Name

- Address
 - so on
- iii. Click **Update**



Note: Refer Mailchimp system to invite a subscriber and update profile.

Content Integration Framework will get notified by the user event with user data which will in turn send user data to Journey.

13. Setting up Journey

- a. Access the Unica Journey application with administrator privileges.
- b. Open the *Unica Journey Administrator's Guide* and perform the steps mentioned in **Settings > REST Integration > Creating a new REST integration**.
- c. Access the Unica Platform application.
- d. Select **Settings > Users**.
- e. Select the username you want to register for app credentials. For example, `asm_admin`.
- f. Click **Edit data sources**.
- g. Click **Add new**.
- h. For the **Data source** field, provide the value `JOURNEY_DS`.
- i. For the **Data source login** field, provide the value of **Client ID** that you copied from the app.
- j. For the **Data source password** and **Confirm password** fields, provide the value of **Client Secret** that you copied from the app.
- k. Click **Save changes**.
- l. From the **Configuration categories** panel, expand **Unica Product > Content Integration > dataSources > Journey**.

Result

The **Settings for Journey** page appears.

- m. Provide values for the following fields and click **Save changes**:
 - **System Identifier** - `Journey`.
 - **User credentials** - `Default User`
 - **defaultUserCredentials** - `asm_admin`
 - **Data source name for credentials** - `JOURNEY_DS` (as set up in *Step h*).
 - **Anonymous Content Access** - `Yes`
- n. Select **HTTP Gateway** and click **Edit settings**.

- o. For the **Base URL** field, provide the value `http://<unica_domain>/journey` and click **Save changes**, where: `<unica_domain>` is the domain name or host name where the Journey is hosted.
- p. Select **REST** and click **Edit settings**.
- q. For the **Authentication Type** field, provide the value `None` and click **Save changes**.

14. Setting up Slack

Configuring Slack will involve configurations that must be done on Slack and configurations that will be done on Unica. For more information, see [Configuring Slack for Integration with Unica on page 41](#).

Configuring Slack for Integration with Unica

In version 12.1.2, Slack Integration is compatible only with Unica Plan. You must first configure the settings on Slack and then use those details and configure on Unica. To integrate Slack application with Unica Plan, complete the following list of procedures:

Slack configuration: Signing up and Signing into Slack

1. Access the following URL:
<https://slack.com/intl/en-in/get-started#/createnew>
2. To create an account, use your official email address. If required, you can continue with your Google account or Apple account.
3. After successfully creating your account and a workspace, use the following URL to sign in:
<https://slack.com/signin#/signin>

Slack configuration: Accessing workspace and sending invitations

1. After a successful sign in, you will see your workspace.
2. If required, you can create multiple workspaces. The security token and member ID are specific to a workspace. To create a new workspace, select your workspace name and select **Add workspaces > Create a new workspace**.



Note: In case of Unica, all users (including administrator) must reside on the same workspace because the `OAuth` tokens are limited only to a single workspace.

3. To invite team members to a specific workspace:
 - a. Access the required workspace.
 - b. Select the workspace name and select **Invite people to <workspace-name>**.
4. The **Invite people to <workspace-name>** dialog appears. Enter the emails IDs separated by semicolon and click **Send**.
5. The Slackbot provides a notification when invitees accept the invitation and join Slack.
6. You can view the members who have joined by accessing their profiles. Select the three horizontal dots and select **Copy member ID** to fetch their member ID.

Slack configuration: Creating an App

About this task

An administrator must create a Slack app with a set of permissions so that the app is ready for distribution to the team members.

The app, upon installation, uses the oAuth token creation for the administrator of the app as well as the team members.

To create an app, complete the following steps:

1. Open a new tab in the browser on which your Slack session is running.
2. Access the following URL:

<https://api.slack.com/apps>

3. Click **Create New App**.

Result

The **Create an App** dialog appears.

4. Select the option **From scratch**.

Result

The **Name app & choose workspace** dialog appears.

5. Provide an appropriate value for **App Name** and select the required workspace for **Pick a workspace to develop your app in**.
6. Click **Create App**.

Result

The **App Summary** page appears indicating successful creation of the app.

7. Scroll down to the **App Credentials** section and copy the values of the following fields:
 - **Client ID**
 - **Client Secret**

Slack configuration: Additional configurations on Slack

About this task

You must complete the following additional configurations on Slack:

1. From the left pane, within the **Settings** section, select **OAuth & Permissions**.
2. Scroll down to the **Redirect URLs, Scopes, and IP Address Ranges** section and click **Edit via Manifest**.

Result

The **Manifest** screen appears.

3. Switch to the **JSON** tab.
4. Append the following code, within the existing code, in the JSON tab:

```
"oauth_config": {
  "redirect_urls": [
    "https://<<unica-domain.com>>/asset-viewer/api/AssetPicker/webhook/Slack/auth/user"
  ],
  "scopes": {
    "user": [
```

```

        "channels:write",
        "chat:write",
        "groups:history",
        "groups:read",
        "groups:write",
        "im:read",
        "im:write",
        "mpim:history",
        "mpim:read",
        "mpim:write"
    ]
}
}

```



Note: In the earlier code, for the `redirect_urls` parameter, ensure that you replace `<<unica-domain.com>>` with an appropriate domain name, and that the protocol must be HTTPS.

Result

The complete entry in the **JSON** tab should be as follows:

```

{
  "_metadata": {
    "major_version": 1,
    "minor_version": 1
  },
  "display_information": {
    "name": "slack_app_demo"
  },
  "oauth_config": {
    "redirect_urls": [
      "https://unica-domain.com/asset-viewer/api/AssetPicker/webhook/Slack/auth/user"
    ],
    "scopes": {
      "user": [
        "channels:write",
        "chat:write",
        "groups:history",
        "groups:read",
        "groups:write",
        "im:read",
        "im:write",
        "mpim:history",
        "mpim:read",
        "mpim:write" ]
    }
  },
  "settings": {
    "org_deploy_enabled": false,
    "socket_mode_enabled": false,
    "token_rotation_enabled": false
  }
}

```

5. Click **Save Changes**.

Slack configuration: Retrieving Administrator's OAuth Token

About this task

To retrieve administrator's OAuth token, the administrator must complete the following steps:

1. From the left pane, within the **Settings** section, select **OAuth & Permissions**.

Result

The **OAuth & Permissions** page appears.

2. In the **OAuth Tokens for Your Workspace** section, click **Install to Workspace**.
3. Note down the **User OAuth Token**.

Slack configuration: Distributing the Slack App

About this task

Before distributing Slack app to team members, admin must ensure that their own account and the Unica team members must be configured for the data source `SLACK_DS` using their respective Slack **Member IDs**. In case of the administrator, the password will be the `OAuth` token retrieved in [Slack configuration: Retrieving Administrator's OAuth Token on page 44](#).

In case of other users, Password can be a dummy value as the system internally updates team member's OAuth token at the time of app installation by the team members. Unica admin must complete this step before distributing the Slack app.

Before distributing the Slack app, ensure that you complete [Unica configuration: Disabling Authentication for Inbound URL on page 45](#).

To distribute the Slack app, complete the following steps:

1. From the left pane, within the **Distribution** section, select **Manage Distribution**.

Result

The **Manage Distribution** page appears.

2. Copy the **Sharable URL** and paste it in an editor.
3. Share it with the required team members.
4. When a team member tries to install Slack, the Slack bot will notify you about the installation request. You must approve the request by clicking **Approve for Workspace** in the notification window.



Note: The first user getting the approval needs to continue the installation of Slack. All subsequent users will not have to go through the approval workflow.

Unica configuration: Registering App Credentials on Unica for the Administrator

About this task



Note: The administrator and the users must have the the Content Integration role assigned to them. For more information on assigning roles to users or user groups, see *Unica Platform Administrator Guide*.

After receiving the Client ID and Client Secret from [Slack configuration: Creating an App on page 42](#), you must register the credentials on Unica for Slack integration. To register the app credentials, complete the following steps:

1. Access the Unica application.
2. Select **Settings > Users**.
3. Select the username you want to register for app credentials. For example, `asm_admin`.
4. Click **Edit data sources**.
5. Click **Add new**.
6. For the **Data source** field, provide the value `SLACK_APP_DS`.
7. For the **Data source login** field, provide the value of **Client ID** that you copied from the app.
8. For the **Data source password** and **Confirm password** fields, provide the value of **Client Secret** that you copied from the app.
9. Click **Save changes**.

Unica configuration: Disabling Authentication for Inbound URL

About this task

You must disable authentication for the inbound URL `/webhook/Slack/auth/*` and create a security filter by completing the following steps:

1. Select **Settings > Configuration**.
2. Expand **Unica Platform > Security > API management > Unica Content Integration**.
3. Select **Slack_inbound**.
If you do not see the **Slack_inbound** option, contact your Unica administrator.
4. Click **Edit settings**.
5. For the API URI field, provide the value `/webhook/Slack/auth/*`.
6. Ensure that:
 - **Block API access** is `Disabled`.
 - **Require authentication for API access** is `Disabled`.
 - **Secure API access over HTTPS** is either `Enabled` or `Disabled` as per your environment requirements.
7. Do not modify any other values and click **Save changes**.

Unica configuration: Registering the Administrator and Other Slack Invited Users on Unica

About this task

The admin can now use the member IDs, as retrieved in the section [Slack configuration: Accessing workspace and sending invitations on page 41](#), and the User OAuth Token, as retrieved in the section [Slack configuration: Retrieving Administrator's OAuth Token on page 44](#) to register members on Unica by completing the following steps:

1. Select **Settings > Users**.
2. Select the username you want to register for app credentials. For example, `demo_user`.
3. Click **Edit data sources**.
4. Click **Add new**.
5. For the **Data source** field, provide the value `SLACK_DS`.
6. For the **Data source login** field, provide the value of **Member ID** that you copied.
7. For the **Data source password** and **Confirm password** fields:

In case of the administrator, the password will be the `OAuth` token retrieved in [Slack configuration: Retrieving Administrator's OAuth Token on page 44](#). In case of other users, Password can be a dummy value as the system internally updates team member's `OAuth` token at the time of app installation by the team members.

8. Click **Save changes**.

Unica configuration: Configuring Additional Parameters for Slack Data Source in Unica

About this task

To configure the additional parameters, complete the following steps:

1. Select **Settings > Configuration**.
2. On the left pane, expand **Plan > Content Integration > dataSources > cmsConfigurationTemplates**.



Note: Plan does not support partition.

3. Click **Slack**
4. Select **HTTPS Gateway**.
5. Click **Edit settings**.
6. For the **Base URL** field, provide the value `https://slack.com/api`.
7. For the **bypassProxy** field, set the value `No`.
8. Click **Save changes**.
9. On the left pane, select **Slack**.
10. Click **Edit settings**.
11. For the **Additional parameters** field, add the following values:

```
unicaAdminUserName: "asm_admin"
dataSourceForSlackApp: "SLACK_APP_DS"
alwaysCreatePublicChannel: true/false
numberOfThreadsToRemoveChannelMembers: 4
channelBaseUrl: "https://app.slack.com/messages/"
```

12. Click **Save changes**.
13. Expand **HTTPS Gateway** and select **REST**.
14. Click **Edit Settings**.
15. Ensure that the value of **Authentication Type** is set to `None`.
16. Click **Save changes**.