Unica Journey
V12.1.1 Administrator's Guide

Contents

Chapter 1. An Introduction to Unica Journey

Unica Journey is a goal-based orchestration solution to craft, execute, and visualize context-driven, personalized, multi-step omnichannel customer experiences.

Marketers can use Unica Journey to:

- Define goals for customer experience
- Easily adjust journeys in real time to achieve them
- Craft and visualize entire customer journey across channels/touchpoints and events with a sleek and intuitive Journey Canvas

Customer journeys are completely automated and synchronized with every step of your customer's brand engagement. Use the real-time Insights within Journey to understand customer behavior with insights that reflect things as they happen in their Journey.

Features of Unica Journey

The features of Unica Journey are as follows:

- **Goal driven Experiences**: Define goals for your customer experience and easily adjust your journeys in real time to achieve them.
- **Orchestration Canvas**: Craft and visualize your entire customer journey across channels/touchpoints and events with a sleek and intuitive Journey Canvas.
- **Always on Engagement**: Completely automated execution that is in sync with every step of your customer's brand engagement.
- **Real-time Insights**: Understand your customer behavior with insights that reflect things as they happen in their Journeys.
- **Choice of Touchpoints**: Leverage the out of the box native touchpoints for digital channels or craft a custom touchpoint and seamlessly orchestrate the journey across your eco system.

- **Dynamic Data Framework**: Flexible data definition and entry sources to augment customer journey with contextual data and events from multiple touchpoints and in variety of formats (File, API, etc.)

Benefits of Unica Journey

The benefits of Unica Journey are as follows:

- **Increased Brand Loyalty**: Strengthen your brand following with targeted and automated journeys that acquire, nurture, convert and retain customers.
- **Amplified Omni Channel Engagement**: Deliver a consistent customer experience across channels with native integration for outbound (Unica Campaign) and inbound engagement (Unica Interact, Unica Discover, and Unica Deliver).
- **Shorten your Customer Conversion Cycle**: Be a step ahead and drive your customer to their goals with timely next best actions.
- **React to the Moment**: You will not miss any opportunity to know where your customer is on their journey and delight them with relevant experience.
- **Lower Marketing TCO**: Reduce your marketing TCO with automated flows and plug and play integration to your MarTech ecosystem through an open and flexible framework powered by the Unica Link.

Chapter 2. Unica Journey integrations

Unica Journey execution engines for email

Unica Journey supports Unica Deliver and Unica Link for email delivery. You can use either for integration with Journey.

Unica Journey integration with Unica Link

Unica Link provides capabilities to send communications across email, SMS, CRM, ADTECH and JDBC channels. Unica Link provides the following reference connectors to deliver communications to email, SMS, CRM, ADTECH and JDBC channels.

Install the following reference connectors as per your preference:

- **MailChimp** – for email
- **Mandrill** – for email
- **Twilio** – for SMS
- **Salesforce** – for CRM

Integration with Unica Link allows Journey to integrate with any third-party vendors for email, SMS, CRM, ADTECH and JDBC executions only.

Table 1. Installation and Configuration of Unica Link

Task	Documentation
Installation and configuration of Unica Link	See *Unica Link V12.1 Installation Guide*.
Installing Unica Link connector app for Journey	See *Unica Link V12.1 Installation Guide*.
Installing Unica Link connector – MailChimp	See *Unica Link Mailchimp Connector User Guide*.
Installing Unica Link connector – Mandrill	See *Unica Link Mandrill Connector User Guide*.

Task	Documentation
Installing Unica Link connector – Twilio	See *Unica Link Twilio Connector User Guide*.
Installing Unica Link connector – Salesforce	See *Unica Link Salesforce Connector User Guide*.

Note: HCL does not provide the account or access to these delivery channel vendors. Based on your preference you can get the entitlements or accounts from these vendors.

Unica Journey integration with Unica Deliver

Unica Journey utilizes the capabilities of Unica Deliver for sending email communications. This also helps in capturing the email responses in real-time and process audience Journey. For more details on enabling Unica Deliver integration with Unica Journey, see Unica Journey Installation Guide.

Unica Journey integration with Unica Campaign and Unica Interact

Unica Journey seamlessly integrates with Unica Campaign and Unica Interact. Unica Campaign and Unica Interact sends audience data to Unica Journey on a specific Kafka topic. The audience data is sent through a Kafka entry source and pushed across all journeys consuming data from these entry sources.

For more information on Unica Campaign and Unica Interact integration with Unica Journey, see the guides mentioned in the following documentation map.

Journey supports data from multiple partitions of Campaign

Journey support data from multiple partitions of campaign.

1. Journey application does not support multiple partitions.
2. Only data from multiple partitions of Campaign/Interact/Deliver can be processed in Journeys. For this journey, will run on single partition.

You need to make changes in configuration platform and user roles and permission:

- Campaign flowchart details displayed under the entry sources come from multiple partitions.
- Based on the partition the Deliver templates are displayed in email / SMS / WhatsApp touchpoints.

Table 2. Integration of Unica Campaign with other HCL products

Task	Documentation
Integration of Unica Campaign and Unica Journey	See *Unica Campaign Administration Guide* and *Unica Campaign User Guide*.
Integration of Unica Campaign and Unica Interact	See *Unica Interact Administration Guide*.

Unica Journey integration with Unica Discover

Unica Journey seamlessly integrates with Unica Discover. Unica Discover sends audience struggle data to Unica Journey. The audience data is sent through REST entry source and pushed across all journeys consuming data from these entry sources. Four scripts will be provided, after installing Journey you need to immediately run the scripts, this will create two entry sources and two data definition called Discover Entry source for CART and Discover Entry source for Form.

DD Name	Cart
Description	When Customer abandons any kind of cart or set of selected offerings this event can be triggered.

Table 3. Attributes to be sent across

Name	Type	Length	Note
Email*	TEXT	200	It is a mandatory field.
Name	TEXT	200	

Name	Type	Length	Note
DiscoverSessionId	TEXT	50	Discover Session id required to link it back.
CartId	TEXT	50	Unique id to identify cart.
CartValue	NUMBER		
EventDateTime	TIMESTAMP		Date and time of the event in UTC Longitude
EventType	TEXT		Event Type can be CART_ABANDONED
CookieID	TEXT	1024	
TLT_BROWSER	TEXT	50	Browser details
TLT_MODEL	TEXT	50	Device Details
HTTP_ACCEPT_LANGUAGE	TEXT	50	Language

DD Name	Form
Description	When Customer fills any webform this event can be published.

Table 4. Attributes to be sent across

Name	Type	Length	Note
Email*	TEXT	200	It is a mandatory field.
Name	TEXT	200	

Name	Type	Length	Note
DiscoverSessionId	TEXT	50	Discover Session id required to link it back.
FormId	TEXT	50	Unique id to identify form
FormName	TEXT	100	
EventDateTime	TIMESTAMP		Date and time of the event in UTC Longitude
CookieID	TEXT	1024	
TLT_BROWSER	TEXT	50	Browser details
TLT_MODEL	TEXT	50	Device Details
HTTP_ACCEPT_LANGUAGE	TEXT	50	Language
EventType	TEXT		Event Type can be FORM_SUBMITTED, FORM_ABANDONED

Note: From Fixpack 3 onwards Unica Journey integration with Unica Discover feature is available.

An Introduction to Unica Deliver

Unica Deliver is a web-based, enterprise scale marketing message solution that you can use to conduct outbound bulk messaging and transactional messaging campaigns. Deliver integrates with Unica Campaign and with secure message composition, transmission, and tracking resources that are hosted by Unica.

You can use Deliver to create, send, and track personalized email communication. As Deliver installs and operates with Campaign, you can use Campaign flowcharts to precisely select and segment recipient information to customize each message.

Select your audience

Use Campaign to select message recipients and data about each person that you can use to personalize each message.

With Deliver, you can reach large numbers of email recipients quickly and personally. However, you can also configure a mailing to automatically send a single email message in response to a transaction.

Create a message

The Deliver Document Composer provides editing tools that you can use to design, preview, and publish personalized message content. You can create messages with content that you upload to the Document Composer or link to external content when Deliver builds and transmits messages. Deliver provides several ways to design messages that display content conditionally, based on personal data for each recipient.

Send the message and track responses

Depending on your goals, you can schedule a messaging campaign to run as soon as possible or schedule it to run later. Deliver monitors message delivery and tracks recipient responses. The system returns contact and response data to the Deliver system tables that are installed as part of the Campaign database schema.

How to get started

To get started, you must install Campaign and have a hosted messaging account.

System administrators must request a hosted messaging account and work with Unica to configure secure access to the remote messaging and tracking systems. Some messaging features are available only upon request to Unica. For more information about establishing a hosted messaging account and configuring access to Unica hosted messaging, see the Unica Deliver Startup and Administrator's Guide.

Unica Deliver integration

To integrate Unica Deliver with Unica Journey, you must make the following configurations inUnica Platform.

1. In Unica Platform, navigate to **Settings > Configuration**.
   The **Configuration categories** page appears.
2. Select **Journey**.
   The **Settings for 'Journey'** page appears.
3. Select **Edit settings**.
   The **(Journey)** page appears.
4. Perform the following steps:
   a. For the **Deliver_Configured** field, select **Yes**.
   b. Click **Save changes**.
5. In the expanded Journey node, select **Deliver_Configurations**.
   The **Settings for 'Deliver_Configurations'** page appears.
6. Select **Edit settings**.
   The **(Deliver_Configurations)** page appears.
7. Perform the following steps:

   a. Provide values for the following fields:
      - **Deliver_URL**: The URL cofigured for Deliver.
      - **Deliver_Partition**: The partition in which the credentials to access the **Deliver_URL** is stored.

   b. Click **Save changes**.

Kafka integration

You must configure Kafka in Unica Platform for the Journey node.

Accessing Kafka_Configurations in Unica Platform

To access Kafka_Configurations, complete the following steps:

1. On Unica Platform, navigate to **Settings > Configuration**.
2. Expand the **Journey** node.
3. Select **Kafka_Configurations**.
4. Select **Edit settings**.

Mandatory configurations based on CommunicationMechanism value

In the **(Kafka_Configurations)** page, you can select one of the following values for the CommunicationMechanism field:

- NO_SASLPLAINTEXT_SSL
- SASL_PLAINTEXT
- SSL
- SASL_PLAINTEXT_SSL

Based on your selection, the following fields become mandatory:

Field name	NO_SASLPLAIN TEXT_SSL	SASL_PLAIN TEXT	SSL	SASL_PLAIN TEXT_SSL
KafkaBrokerURL	Yes	Yes	Yes	Yes
TopicName	Yes	Yes	Yes	Yes
sasl.mechanism		Yes		Yes
UserForKafkaData		Yes	Yes	Yes

Field name	NO_SASLPLAIN TEXT_SSL	SASL_PLAIN TEXT	SSL	SASL_PLAIN TEXT_SSL
sasl.jaas.config.data Source		Yes		Yes
truststore.location			Yes	Yes
truststore.password.data Source			Yes	Yes
keystore.location			Yes	Yes
keystore.password.data Source			Yes	Yes
key.password.dataSource			Optional	Optional
ssl.endpoint.identification. algorithm			Yes	Yes

Make the necessary configurations and click **Save changes**.

Note: Running out of disk storage and abruptly shut down the kafka server due to the large size of kafka logs file.

Chapter 3. Process of Starting and Stopping Journey

Starting Process

1. Starting Process Web
    a. Configure Kafka and Zookeeper
        i. IP – on which Zookeeper/Kafka is running
        ii. PORT- Kafka(default 9092), Zookeeper default port 2181
        iii. Log path
        iv. auto.create.topic.enable = true, this property should be set to true for Engine Publish service to work.
    b. Start Zookeeper, wait 10 sec
    c. Start kafka
    d. configure Journey.xml --(Refer Doc, Doc2)
    e. Configure Log4j2.xml under conf folder
    f. Start Webserver (JBOSS/TOMCAT/WebSphere)
    g. Start Journey Web application

2. Starting Engine
    a. Configure application.porperties
        i. Add DB Details
        ii. Add Kafka details(eg: spring.kafka.bootstrap-servers=127.0.0.1:9092, 127.0.0.2:9092)
            - Path for Ignite Storage, spring.ignite.storage.path, User thru which engine is executed should have Read and Write access to the path of Ignite folder
        iii. Configure Log4j2.xml under conf folder
        iv. Configure property spring.ignite.ipFinder.List as below:
            -

```
spring.ignite.ipFinder.List=127.0.0.1:63501,127.0.0.1:63502,
127.0.0.1:63503,127.0.0.1:63504
```

v. Start Engine (java –jar journeyEngine.jar)

Stopping Process (Steps for No Data Loss)

a. Stop Webserver

b. Stop Engine (grep and kill Pid)or use Director

c. Stop kafka

d. Stop Zookeeper

Chapter 4. Journey user roles and permissions

Before you begin using Unica Journey, you should assign roles and permissions to users.

Note: Any changes in configuration requires a restart of Unica Journey. For more information related to security configurations, see *Unica Platform Administrator's Guide*.

Assigning permissions to Journey Roles

Before assigning a role to a user, you should assign permissions to the available roles.

Journey offers two user roles:

- **JourneyAdmin**
- **JourneyUser**

To assign permissions to both roles, complete the following steps:

1. From the Unica Platform home page, select **Settings > User roles and permissions**.

   The **User roles and permissions** page appears.

2. In the left panel, expand **Unica Journey > partition1**.

   The **partition1** page appears.

3. Select **Assign Permissions**.

   The **(Properties for administrative roles)** page appears.

4. Click **Save and edit permissions**.

The **(Permissions for partition1)** page appears.

5. Expand **Application**.
6. Set values for the following fields:

Operations	**JourneyAdmin Default setting**	**JourneyUser Default setting**
Create Data Definition	Yes	No
Edit Data Definition	Yes	No
Delete Data Definition	Yes	No
Create Entry Sources	Yes	No
Edit Entry Sources	Yes	No
Delete Entry Sources	Yes	No
Create Journey	Yes	Yes
Edit Journey	Yes	Yes
Delete Journey	Yes	No
Publish Journey	Yes	Yes
Complete Journey	Yes	Yes
Pause Journey	Yes	Yes
Goal add/modify/delete	Yes	No
Goal view	Yes	Yes
Settings add/modify/delete	Yes	No
Settings view	Yes	Yes

Note:

- For the **JourneyAdmin** role, we recommend that you do not reduce the permissions and retain the default permissions. By default, **JourneyAdmin** has all permissions.
- For the **JourneyUser** role, provide permissions that you feel is appropriate. You can give the **JourneyUser** all permissions, but it is not recommended.

7. After providing the permissions, click **Save changes**.

Assigning JourneyAdmin role to a user

To assign **JourneyAdmin** role to a user, complete the following steps:

1. From the Marketing Platform home page, select **Settings > User roles and permissions**. The **User roles and permissions** page appears.
2. In the left panel, expand **Unica Journey**.
3. Select **partition1 > JourneyAdmin**. The **JourneyAdmin** page appears.
4. In the **Users** section, select a user. For example, `asm_admin`. The **asm_admin (asm_admin)** user details page appears.
5. Select **Edit roles**. The **Edit roles** page appears.
6. From the **Available roles** list, select **JourneyAdmin (Unica Journey)** and click the **>>** button to move the role to the **Selected roles** list.
7. Click **Save changes**.

Assigning JourneyUser role to a user

To assign **JourneyUser** role to a user, complete the following steps:

1. From the Marketing Platform home page, select **Settings > User roles and permissions**. The **User roles and permissions** page appears.
2. In the left panel, expand **Unica Journey**.
3. Select **partition1 > JourneyUser**. The **JourneyUser** page appears.
4. In the **Users** section, select a user. For example, `journey_example`. The **journey_example (journey_example)** user details page appears.
5. Select **Edit roles**. The **Edit roles** page appears.
6. From the **Available roles** list, select **JourneyUser (Unica Journey)** and click the **>>** button to move the role to the **Selected roles** list.
7. Click **Save changes**.

Chapter 5. Journey interaction logging

Interaction Logging for Journey is executed as a scheduled job. The scheduling parameters are set in `application.properties` file of the Journey Engine. The following is an example of the setting:

```
engine.logging.cron=0 15 3 * * ?
```

The scheduled job exports data into an alternative schema which is defined again in the `application.properties` files of the Journey Engine.

```
journey.report.datasource.url =

journey.report.datasource.username =

journey.report.datasource.password =

journey.report.datasource.driver-class-name=
```

Interaction Logging captures the movement of every contact that enters the Journey application as they move through each Journey, either Published or Completed. Even journeys that are Published but Paused are considered for the Interaction Logging.

All Touchpoints, Email, SMS, or CRM, are considered for Interaction Logging as the audience data is sent using the configured integrations through the respective channels. The responses received, from every contact, is also captured.

Log4j2

Both Journey Web and Journey Engine uses the standard for logging. The `log4j2.xml` file, for both Journey Web and Journey Engine, is placed within the `conf` folder in the installation location.

Both Journey Web and Journey Engine produce regular application logs as well as performance logs. For Journey Web, the default location of the logs is within the `logs` folder. For Journey Engine, the default location of the logs is within the `performancelogs` folder. For both Journey Web and Journey Engine, the mentioned folders are placed within the installation location.

Chapter 6. Journey GDPR

Accessing Journey GDPR

You can access the GDPR tool from the Journey application folder. The location is as follows:

```
<Journey_Home>\Journey\tools\GDPR\
```

GDPR supports > MariaDB, **MS Sql server**, **OneDb data bases** along with **Oracle**

Executing Journey GDPR

To execute Journey GDPR, complete the following steps:

1. Make changes to the following properties in the `gdpr.properties` file:

Property Name	**Example value**	**Notes**
`Journey.audience.DBType`	`ORACLE`	Currently, Journey supports only Oracle.
`Journey.audience.Db.Schema.Name`	`JourneyUser`	Schema name used in the Journey database.
`Journey.audience.Field`	`email/mobileNumber`	Field name in the input `CSV` file.
`Journey.audience.Csv`	`<GDPR_HOME>/sample/JourneyAudiences.csv`	Replace `<GDPR_HOME>` with the current directory path. This is the input CSV file containing records that you need to opt out from the Journey.

Property Name	Example value	Notes
`Journey.audience.Output`	`<GDPR_HOME>/JourneyAudiences.sql`	The `JourneyAudiences.sql` is the output file name containing all the SQL queries used to drop all the records from the Journey application. Replace `<GDPR_HOME>` with the current directory path.
`Journey.audience.Output.FileSizeLimit`	`10`	Value is in MBs. When the file size exceeds the entered value, it will generate multiple files with the following suffixes: `JourneyAudiences _0`, `JourneyAudiences _1`, and so on.

2. **Note:** If you see any errors, you can trace it using this log file.
3. To execute the file, perform one of the following steps:
   a. For Windows, locate and execute the `gdpr_purge.bat` file. For example, if the file `gdpr_purge.bat` is in the `D:\workspace\HCL_GDPR\dist\journey\` location, run the `gdpr_purge.bat` file.
   b. For UNIX-based systems, locate and execute the `gdpr_purge.sh` file. For example, if the file `gdpr_purge.sh` is in the `\workspace\HCL_GDPR\dist\journey\` location, run the command ./gdpr_purge.sh.
4. After running `gdpr_purge.bat` (for Windows) or `gdpr_purge.sh` (For Linux), output files *"JourneyAudiences 0"*, *"JourneyAudiences _1"*, *"JourneyAudiences _2"* and so on will be generated at location `<GDPR_HOME>` specified in above steps. Number of files generated will depend on filesize specified.

5. *"JourneyAudiences_x"* file will have delete queries for records mentioned in JourneyAudiences.csv
6. These queries need to be run manually in "Journey" database as required to have the records deleted from the journeyaudiences table.

GDPR utility removes records from following table: JourneyAudiences, AudienceResponse, AudienceResponseMetaData, AudienceResponseInteraction, JourneyAudienceMilestone and JourneyAudienceGoal. However, it does not delete the data from respective tables, where aggregated counts are stored. For example, tables like journeyFlow, journeyAudienceFlow, JourneyGoalContactTransaction etc. Hence, there will be count mismatch on UI.

With GDPR tool, user would not be able to delete the customer data from either Publish Kafka topic or from the files available on the file system. User need to Delete this data manually as per their requirement.

With GDPR tool user will not able to delete customer data exported by JDBC connector.

Chapter 7. Kafka authentication using SSL

If you are using your organization's Kafka instance, you can use certificates configured for that Kafka instance. You are not required to generate SSL key and certificates and obtain the client certificates to configure in Journey application properties.

If you do not have the certificates, you can generate self-signed certificate authority (CA), which is simply a public-private key pair and certificate.

You must add the same CA certificate to each Kafka client and broker's trust store.

Generate SSL key and certificate for each Kafka broker

To generate self-signed certificates for Kafka server, complete the following steps.

Prerequisites

- You must have Java keytool and OpenSSL to generate certificates and trust store.
- Optionally, you can use any SSL certificate generation utility instead of OpenSSL.

1. To deploy SSL, generate the key and the certificate for each machine in the cluster. Generate the key into a temporary keystore initially so that you can export and sign it later with CA.

   ```
   keytool -keystore kafka.server.keystore.jks -alias localhost -validity 365
   -genkey
   ```

   - keystore: The keystore file that stores the certificate. The keystore file contains the private key of the certificate; therefore, it needs to be kept safely.
   - validity: The valid time of the certificate in days.
2. Create your own CA (certificate authority)

   ```
   openssl req -new -x509 -keyout ca-key -out ca-cert -days 365
   ```

   The generated CA is simply a public-private key pair and certificate, and it is intended to sign other certificates.
3. Add the generated CA to the clients' trust store so that the clients can trust this CA.

-
  ```
  keytool -keystore kafka.server.truststore.jks -alias CARoot -import -file ca-cert
  ```

-
  ```
  keytool -keystore kafka.client.truststore.jks -alias CARoot -import -file ca-cert
  ```

4. Sign all certificates in the keystore with the CA generated.
    a. Export the certificate from the keystore:

    ```
    keytool -keystore kafka.server.keystore.jks -alias localhost -certreq -file cert-file
    ```

5. Sign it with CA.

    ```
    openssl x509 -req -CA ca-cert -CAkey ca-key -in cert-file -out cert-signed -days 365 -CAcreateserial -passin pass:<password>
    ```

6. Import both the certificates of the CA and the signed certificate into the keystore.

    ```
    keytool -keystore kafka.server.keystore.jks -alias CARoot -import -file ca-cert
    ```

    ```
    keytool -keystore kafka.server.keystore.jks -alias localhost -import -file cert-signed
    ```

7. Create client keystore and import both certificates of the CA and signed certificates to client keystore. These client certificates will be used in application properties.

    ```
    keytool -keystore kafka.client.keystore.jks -alias localhost -validity 365 -genkey
    ```

    ```
    keytool -keystore kafka.client.keystore.jks -alias localhost -certreq -file cert-file
    ```

    ```
    openssl x509 -req -CA ca-cert -CAkey ca-key -in cert-file -out cert-signed -days 365 -CAcreateserial -passin pass:<password>
    ```

    ```
    keytool -keystore kafka.client.keystore.jks -alias CARoot -import -file ca-cert
    ```

    ```
    keytool -keystore kafka.client.keystore.jks -alias localhost -import -file cert-signed
    ```

Configuring Kafka server, Journey and Link components with SSL

The server certificates to be used for Kafka server and client certificates must be used by any application connecting to Kafka server including Journey Web, Journey Engine, Unica Link – Kafka-link or any other tools you require to connect to this Kafka server.

To configure Kafka Server, Journey components, and Link component with SSL authentication, execute the procedures provided in the following sections.

Configuring Kafka Server with SSL authentication

You must use the following server certificates for Kafka server only. Share these certificates on the required machines and make a note of password.

- `kafka.server.keystore.jks`
- `Kafka.server.truststore.jks`

Update the following server.properties in Kafka server config directory.

```
listeners=SSL://<KAFKA_HOST>:<KAFKA_PORT>
ssl.keystore.location=/PATH/kafka.server.keystore.jks
ssl.keystore.password= password
ssl.key.password= password
ssl.truststore.location= /PATH/kafka.server.truststore.jks
ssl.truststore.password= password
ssl.endpoint.identification.algorithm=
ssl.client.auth=required
security.inter.broker.protocol=SSL
```

Configure Journey engine with Kafka SSL

Use the following client certificates and share these certificates on the required machines and make a note of password.

- `Kafka.client.keystore.jks`
- `kafka.client.truststore.jks`

1. Update Journey Engine `log4j2.xml` file from `<JOURNEY_HOME>/Engine/conf/` directory. Uncomment the following lines in `log4j2.xml`.

```
<Property name="security.protocol" >${sys:security.protocol}</
Property>
<Property name="ssl.truststore.location">
${sys:ssl.truststore.location}</Property>
<Property name="ssl.truststore.password">
${sys:ssl.truststore.password}</Property>
<Property name="ssl.keystore.location">${sys:ssl.keystore.location}</
Property>
<Property name="ssl.keystore.password">${sys:ssl.keystore.password}</
Property>
<Property name="ssl.key.password">${sys:ssl.key.password}</Property>
<Property name="ssl.endpoint.identification.algorithm">
${sys:ssl.endpoint.identification.algorithm}</Property>
```

2. Update `journey_engine_master.config` from `<JOURNEY_HOME>/Engine/` directory.
3. Update the following property values.

```
kafka.security.enabled=Y
kafka.security.protocols.enabled=SSL
security.protocol=SSL
ssl.truststore.location= /PATH/kafka.client.truststore.jks
ssl.truststore.password=<ENCYPTED PASSWORD WITH JOURNEY ENCRYPTION TOOL>
ssl.keystore.location= /PATH/kafka.client.keystore.jks
ssl.keystore.password=<ENCYPTED PASSWORD WITH JOURNEY ENCRYPTION TOOL>
ssl.key.password=<ENCYPTED PASSWORD WITH JOURNEY ENCRYPTION TOOL>
ssl.endpoint.identification.algorithm=
```

Configuring Journey web with Kafka SSL

1. Update Journey Web `application.properties` file from `<JOURNEY_HOME>/Web/properties/` directory.
2. Update the following property values.

```
kafka.security.enabled=Y
kafka.security.protocols.enabled=SSL
ssl.truststore.location= /PATH/kafka.client.truststore.jks
ssl.truststore.password= <ENCYPTED PASSWORD WITH JOURNEY ENCRYPTION TOOL>
ssl.keystore.location= /PATH/kafka.client.keystore.jks
ssl.keystore.password= <ENCYPTED PASSWORD WITH JOURNEY ENCRYPTION TOOL>
ssl.key.password= <ENCYPTED PASSWORD WITH JOURNEY ENCRYPTION TOOL>
ssl.endpoint.identification.algorithm=
```

Configuring Unica Link component with SSL

Update the following property values in Unica Link installations - `kafkalink.properties` file.

```
security.ssl=true
security.protocol=SSL
ssl.truststore.location= /PATH/kafka.client.truststore.jks
ssl.truststore.password=password
security.authentication=username
ssl.keystore.location= /PATH/kafka.client.keystore.jks
ssl.keystore.password=password
ssl.key.password=passwordssl.endpoint.identification.algorithm=
```

Configuring Kafka server, Journey and Link Components with SASL

To configure Kafka Server, Journey components, and Link component with SASL authentication, execute the procedures provided in the following sections.

Configuring Kafka Server with SASL authentication

1. Specify JVM parameter in `kafka-run-class.bat/sh`.

   set `JAVA_OPTS=%JAVA_OPTS%`

   `-Djava.security.auth.login.config=/PATH/kafka_server_jaas.conf`

   `set COMMAND=%JAVA% %JAVA_OPTS% %KAFKA_HEAP_OPTS%`

   `%KAFKA_JVM_PERFORMANCE_OPTS% %KAFKA_JMX_OPTS% %KAFKA_LOG4J_OPTS% -cp`

   `"%CLASSPATH%" %KAFKA_OPTS% %*`

   Sample `jaas.config` file:

   ```
   KafkaServer {
      org.apache.kafka.common.security.plain.PlainLoginModule required
      username="admin"
      password="admin-secret"
      user_admin="admin-secret"
      user_alice="alice-secret";
   };
   ```

```
KafkaClient {
  org.apache.kafka.common.security.plain.PlainLoginModule required
  username="alice"
  password="alice-secret";
};
```

2. Update the following Kafka server properties file from `KAFKA_SERVER/config/server.properties`.

```
listeners=SASL_PLAINTEXT:// <KAFKA_HOST>:<KAFKA_PORT>
security.inter.broker.protocol=SASL_PLAINTEXT
sasl.mechanism.inter.broker.protocol=PLAIN
sasl.enabled.mechanisms=PLAIN
```

Configure Journey Engine with Kafka SASL

1. Update Journey Engine `log4j2.xml` file from `<JOURNEY_HOME>/Engine/conf/` directory. Uncomment the following lines in `log4j2.xml`.

```
#<!-- Kafka SASL configuration -->
<Property name="security.protocol">${sys:security.protocol}</Property>
<Property name="sasl.mechanism">${sys:sasl.mechanism}</Property>
```

2. Update `journey_engine_master.config` from `<JOURNEY_HOME>/Engine/` directory. Update the following property values.

```
kafka.security.enabled=Y
kafka.security.protocols.enabled=SASL_PLAINTEXT
security.protocol=SASL_PLAINTEXT
sasl.mechanism=PLAIN
java.security.auth.login.config=./kafka_client_jaas.conf
```

Configuring Journey Web with Kafka SASL

Update Journey Web `application.properties` file from `<JOURNEY_HOME>/Web/properties/` directory.

```
kafka.security.enabled=Y
kafka.security.protocols.enabled=SASL_PLAINTEXT
#java.security.auth.login.config=/PATH/kafka_client_jaas.conf
```

Configuring Unica Link component with Kafka SASL

Update the following property values in Unica Link installation - `kafkalink.properties` file.

```
security.sasl =true
security.protocol=SASL_PLAINTEXT
security.sasl.auth.login.config =/PATH/kafka_client_jaas.conf
sasl.mechanism=PLAIN
```

Configuring Kafka server and Journey Components with SASL_SSL configuration

To configure Kafka Server and other Journey components with SASL authentication, complete the procedures provided in the following sections.

Note: Unica Link does not support connecting to Kafka-link using SASL_SSL authentication mechanism. You must either use SASL or SSL authentication mechanism.

Configuring Kafka Server with Kafka SASL_SSL

Update the following server.properties in Kafka server configuration directory.

```
listeners=SASL_SSL:// <KAFKA_HOST>:<KAFKA_PORT>
security.inter.broker.protocol=SASL_PLAINTEXT
sasl.mechanism.inter.broker.protocol=PLAIN
sasl.enabled.mechanisms=PLAIN
ssl.keystore.location=/PATH/kafka.server.keystore.jks
ssl.keystore.password=password
ssl.key.password= password
ssl.truststore.location=/PATH/kafka.server.truststore.jks
ssl.truststore.password= password
ssl.endpoint.identification.algorithm=
ssl.client.auth=required
security.inter.broker.protocol=SSL
```

Configuring Journey Engine with Kafka SASL_SSL

1. Update Journey Engine `log4j2.xml` file from `<JOURNEY_HOME>/Engine/conf/` directory.

   Uncomment the following lines in `log4j2.xml`.

   ```
   <Property name="sasl.mechanism">${sys:sasl.mechanism}</Property>
   <Property name="security.protocol" >${sys:security.protocol}</
   Property>
   <Property name="ssl.truststore.location" >
   ${sys:ssl.truststore.location}</Property>
   <Property name="ssl.truststore.password">
   ${sys:ssl.truststore.password}</Property>
   <Property name="ssl.keystore.location">${sys:ssl.keystore.location}</
   Property>
   <Property name="ssl.keystore.password">${sys:ssl.keystore.password}</
   Property>
   <Property name="ssl.key.password">${sys:ssl.key.password}</Property>
   ```

```
<Property name="ssl.endpoint.identification.algorithm">
${sys:ssl.endpoint.identification.algorithm}</Property>
```

2. Update the following `journey_engine_master.config` from `<JOURNEY_HOME>/Engine/` directory.

   Update the following property values.

```
kafka.security.enabled=Y
kafka.security.protocols.enabled=SASL_SSL
ssl.truststore.location=/PATH/kafka.client.truststore.jks
ssl.truststore.password=<ENCYPTED PASSWORD WITH JOURNEY ENCRYPTION
 TOOL>
ssl.keystore.location=/PATH/kafka.client.keystore.jks
ssl.keystore.password=<ENCYPTED PASSWORD WITH JOURNEY ENCRYPTION TOOL>
ssl.key.password=<ENCYPTED PASSWORD WITH JOURNEY ENCRYPTION TOOL>
ssl.endpoint.identification.algorithm=
java.security.auth.login.config=/PATH/kafka_client_jaas.conf
```

Configuring Journey Web with Kafka SASL_SSL

Update the following Journey Web `application.properties` file from `<JOURNEY_HOME>/Web/properties/` directory.

```
kafka.security.enabled=Y
kafka.security.protocols.enabled=SASL_SSL
ssl.truststore.location=/PATH/kafka.client.truststore.jks
ssl.truststore.password=<ENCYPTED PASSWORD WITH JOURNEY ENCRYPTION TOOL>
ssl.keystore.location=/PATH/kafka.client.keystore.jks
ssl.keystore.password=<ENCYPTED PASSWORD WITH JOURNEY ENCRYPTION TOOL>
ssl.key.password=<ENCYPTED PASSWORD WITH JOURNEY ENCRYPTION TOOL>
ssl.endpoint.identification.algorithm=
java.security.auth.login.config=/PATH/kafka_client_jaas.conf
```

Chapter 8. Configure Web Application servers Tomcat for SSL

On every application server on which a Unica application is deployed, configure the web application server to use the certificates you have decided to employ.

See your web application server documentation for details on performing these procedures.

Ensuring cookie security

Some cookies may not be properly secured in the client browser. Not securing cookies leaves the application vulnerable to man-in-the-middle and session hijacking attacks. To fix this issue, take the following precautions.

- Enforce the use of SSL at all times to reduce the risk of cookies being intercepted on the wire.
- In the web application server, set the `secure` and `httponly` flags on all cookies.
    - The `secure` flag tells the browser to send the cookie only over an HTTPS connection. You must enable SSL on all applications that communicate with each other if you set this flag.
    - The `httponly` flag prevents cookies from being accessed through a client side script.

Setting the flags for SSL in Tomcat

To set the `secure` and `httponly` flags in Tomcat, perform the following changes in the `.xml` server of Tomcat.

```
<Connector port="7003"
 protocol="org.apache.coyote.http11.Http11NioProtocol"
maxThreads="150" SSLEnabled="true" scheme="https" acceptCount="100"
 clientAuth="false"
```

```
disableUploadTimeout="true" enableLookups="false" secure="true"
 sslProtocol="TLS" keystoreFile="/opt/v12.1/v12.1.0.1.1/Campaign/SSL_NEW/
PlatformClientIdentity.jks" keystorePass="password" >     </Connector>
```

Configure Unica Journey with SSL

To configure Unica Journey to use SSL, you must set some configuration properties. Use the procedures in this section that are appropriate for your installation of Unica Journey and the communications that you want to secure using SSL.

When you access your Unica installation over a secure connection, and when you set navigation properties for applications as described in the following procedures, you must use `https` and the secure port number in the URL. The default SSL port is`8443` for Tomcat.

Follow this procedure to configure Journey with SSL

1. Log in to Unica and click **Settings > Configuration**.
2. Set the value of the `Affinium | Journey | navigation` property to Unica Journey URL.

   For example: `https://`*`host.domain:SSL_port/`*`unica`

   where:

   - *`host`* is the name or IP address of the machine on which Unica Journey is installed
   - *`domain`* is your company domain in which your Unica products are installed
   - *`SSL_Port`* is the SSL port in the application server on which Unica Journey is deployed

   Note `https` in the URL.

Chapter 9. Settings

Use the settings menu to manage the Journey integrations like Email connectors, SMS connectors, CRM connections, and REST integrations.

Setting a default email connection

If you have multiple connectors to Unica Link for sending an email, you can set the default email connection in the **Settings** menu.

To set a default email connection, complete the following steps:

1. Select ⚙ **> Link > Email**.
   The **Email** page appears.
2. From the **Available Connections** list, select a connection.
   The available connection includes Mandril, Mailchimp, etc.
3. Click **Save**.
   You can also deselect an existing connection and click **Save**. This ensures that no default connection is set.

Setting a default SMS connection

If you have multiple connectors to Unica Link for sending an SMS, you can set the default SMS connection in the **Settings** menu.

To set a default SMS connection, complete the following steps:

1. Select ⚙ **> Link > SMS**.
   The **SMS** page appears.

2. From the **Available Connections** list, select a connection.

   Note:

   Phone number formats should be mentioned as per the specification of the delivery channel. Journey will send the phone number in the same format to delivery channel. For example, in reference Twilio connection phone number format supported with Journey is as follows:

   - *<plus sign><country-code><10-digit phone number>* - `+15403241212`.
   - *<plus sign> <country-code <(area-code)> <three-digit number><four-digit number>* - `+1 (540) 324 1212`.
   - *<plus sign>-<country-code>-<area-code>-<three-digit number>-<four-digit number>* - `+1-540-324-1212`.
   - *<plus sign> <country-code>-<area-code>-<three-digit number>-<four-digit number>* - `+1 540-324-1212`.

   Whatever format of phone number you provide, Unica Journey will save the number in the following format: <plus sign><country-code><10-digit phone number>. For example, if you provide phone number as `+1 540-324-1212`, Unica Journey stores the phone number as `+15403241212`.

   If you select Twilio as the default SMS connection, it will accept phone numbers only in the following format: *<plus sign><country-code><10-digit phone number>*. For example, `+15403241212`.

3. Click **Save**.

Setting a default CRM connection

If you have multiple CRM connections, you can set the default CRM connection in the **Settings** menu.

To set a default CRM connection, complete the following steps:

1. Select ⚙ **> Link > CRM**.
   The **CRM** page appears.
2. From the **Available Connections** list, select a connection.
3. Click **Save**.

Setting a default ADTECH connection

If you have multiple ADTECH connections, you can set the default ADTECH connection in the **Settings** menu.

To set a default ADTECH connection, complete the following steps:

1. Select ⚙ **> Link > ADTECH**
   ADTECH page appears
2. From the **Available Connections** list, select a connection.
3. Click **Save**.

Setting a default Database connection

If you have multiple Database connections, you can set the default database connection in the **Settings** menu.

To set a default Database connection, complete the following steps:

1. Select ⚙ **> Link > Database**
   Database page appears
2. From the **Available Connections** list, select a connection.

3. Click **Save**.

Manage connections

You can manage Unica Link connections from this menu.

You can create a connection with Unica Link connectors like Mailchimp, Mandrill, Salesforce, and Twilio. You can view all existing connections in the **Existing Connections** (n) panel, where n is the number of connections.

1. To create a Mailchimp connection, complete the following steps:

    a. Select ⚙ **> Link > Manage Connections > Create New**.
    The **Create New Connection** page appears.

    b. Provide values for the following fields:
        - **Name** - Mandatory
        - **Description** - Optional

    c. Click **Next**.

    d. From the **Choose Connection** panel, select **Mailchimp**.

    e. In the **Connection Properties** panel, provide values for the following mandatory fields:

    Note: To know about the fields and the values to be put, see *Unica Link Mailchimp Connector User Guide*.

        - **Base URL**
        - **User ID**
        - **API Key**
        - **Activity Fetch Frequency**
        - **Activity Fetch Units**

f. Click **Test** to the test the connection. If the provided values are correct, you will see a success message. If the provided values are incorrect, you will see an error message.

g. To save the connection, click **Save**.
The new connection is successfully saved and it appears in the **Existing Connections** panel.

2. To create a Mandril connection, complete the following steps:

a. Select ⚙ **> Link > Manage Connections > Create New**.
The **Create New Connection** page appears.

b. Provide values for the following fields:
- **Name** - Mandatory
- **Description** - Optional

c. Click **Next**.

d. From the **Choose Connection** panel, select **Mandrill**.

e. In the **Connection Properties** panel, provide values for the following mandatory fields:

Note: To know about the fields and the values to be put, see *Unica Link Mandrill User Guide*.

- **API Key**
- **Activity Fetch Frequency**
- **Activity Fetch Units**

f. Click **Test** to the test the connection. If the provided values are correct, you will see a success message. If the provided values are incorrect, you will see an error message.

g. To save the connection, click **Save**.

   The new connection is successfully saved and it appears in the **Existing Connections** panel.

3. To create a Salesforce connection, complete the following steps:

   a. Select ⚙ **> Link > Manage Connections > Create New**.
   The **Create New Connection** page appears.

   b. Provide values for the following fields:
      - **Name** - Mandatory
      - **Description** - Optional

   c. Click **Next**.

   d. From the **Choose Connection** panel, select **Salesforce**.

   e. In the **Connection Properties** panel, provide values for the following mandatory fields:

   Note: To know about the fields and the values to be put, see *Unica Link Salesforce User Guide*.

      - **Instance URL**
      - **Access Token**
      - **Version**

   f. Click **Test** to the test the connection. If the provided values are correct, you will see a success message. If the provided values are incorrect, you will see an error message.

   g. To save the connection, click **Save**.
   The new connection is successfully saved and it appears in the **Existing Connections** panel.

4. To create a Twilio connection, complete the following steps:

   a. Select ⚙ **> Link > Manage Connections > Create New**.
   The **Create New Connection** page appears.

b. Provide values for the following fields:
   - **Name** - Mandatory
   - **Description** - Optional

c. Click **Next**.

d. From the **Choose Connection** panel, select **Twilio**.

e. In the **Connection Properties** panel, provide values for the following mandatory fields:

   Note: To know about the fields and the values to be put, see *Unica Link Twilio User Guide*.

   - **Base URL**
   - **Account SID**
   - **Auth Token**
   - **From Number**
   - **Retry Interval**
   - **Retry Attempts**

f. Click **Test** to the test the connection. If the provided values are correct, you will see a success message. If the provided values are incorrect, you will see an error message.

g. To save the connection, click **Save**.
   The new connection is successfully saved and it appears in the **Existing Connections** panel.

REST Integration

REST keys are used for third-party login to the application. You can generate key-value pair and using the key value pair, you can login to Journey using third-party applications.

Creating a new REST integration

To create a new REST integration key pair, complete the following steps:

1. Select ⚙ **> REST**.

   The **REST** page appears.
2. Click **+ REST Integration**.

   The **New REST Integration** page appears.
3. Provide values for the following fields:
   - **App Name** - Mandatory.
   - **Description** - Optional.
4. Click **Generate Keys**.

   The system generates a **ClientID** and **ClientSecret**.
5. Use the toggle bar to change the **Status** to `Active` or `Inactive`. By default, the **Status** is `Active`.
6. To save the REST integration, click **Save**.

   To send audience data to Journey, follow the details mentioned on the REST Entry Source used for configuring the REST end point. Use the **ClientID** and **ClientSecret**, which you received when executing Step (4), for configuring the REST end point on Entry Source.

Viewing REST integration list

Unica Journey maintains a list of REST integrations created.

To view a list of REST integrations, complete the following steps:

1. Select ⚙ **> REST**.

   The **REST** page appears.

2. Perform any one of the following operations:
    a. To view the list of REST integrations in ascending order or descending order on the Name field, click **Name**.
    b. To view the list of REST integrations in ascending order or descending order on the Description field, click **Description**.

Modifying an existing REST integration

You can only modify the description and the status of an existing REST integration.

To modify an existing REST integration, complete the following steps:

1. Select ⚙ **> REST**.
   The **REST** page appears.
2. To modify a rest integration, you can either:
    - select the required REST integration from the list
    - select ⋮ > ✎

   The **Update REST Integration** page appears.
3. You can update only the following fields:
    - **Description**
    - **Status**
4. To save the modifications, click **Save**.

Deleting REST integrations

You can only delete inactive REST integrations that are no longer used or needed.

To change the status of a REST integration entry, see Modifying an existing REST integration (on page 42).

To remove existing inactive REST integrations, complete the following steps:

1. Select ⚙ **> REST**.
   The **REST** page appears.
2. Perform either of the following steps:
   - To delete a REST integration, select ⋮ **>** 🗑 succeeding the REST integration in the list.
   - To delete multiple REST integrations, select the checkboxes preceding the REST integrations, in the list, that you want to delete and click **Delete**.
3. A confirmation box appears. To proceed with the deletion, click **Ok**.

Journey Proxy Integration

Proxy server has been integrated into journey web and Engine Projects, this gives user an upper hand in adding security and keeping application server behind proxy servers. Proxy server will interact with Deliver, Link and Platform servers.

Journey Web – Communicates with Deliver, Link and Platform server for fetching configuration details and while integrating Email/SMS/AdTech Point in Journey.

Journey Engine – Uses proxy to communicate with Deliver/Link Server for submitted Email/SMS/Adtech details to end servers.

Proxy Supported by Journey Web

1. SOCKS
2. HTTP
3. HTTPS

Proxy Supported by Journey Engine

1. HTTP

Note: SOCKS and HTTPS proxy is not supported by SOAP (Apache Axis2) used by Engine to communicate with Deliver.

Property to be configured for Engine

- journey.proxy.type=NONE
- spring.proxy.host=[IP]
- spring.proxy.port=[PORT]
- spring.proxy.username=[username]
- spring.proxy.password=[password]

Property to be configured for Web

- journey.proxy.type=NONE
- spring.proxy.host=[IP]
- spring.proxy.port=[PORT]
- spring.proxy.username=[USERNAME]
- spring.proxy.password=[PASSWORD]
- server.use-forward-headers=true

Note: Default value of property journey.proxy.type is NONE, when set to NONE proxy is disabled.

Developer Tools

Displays the list of developer tools.

API Documentation

User can find the list of REST APIs for Journey.