

Unica Link V12.1.5 Installation Guide



Contents

- Chapter 1. Introduction to Unica Link..... 1**
 - Apache Kafka-link..... 1
 - Unica Link Docker, Native, and Microsoft Windows installation..... 1
 - Before you install Unica Link..... 2
- Chapter 2. Unica Link prerequisites.....3**
 - Unica Link upgrade.....3
 - Installation prerequisites for Linux..... 4
 - Docker installation prerequisites..... 5
 - Native installation prerequisites..... 5
 - Microsoft Windows installation prerequisites..... 6
 - Link installation prerequisites.....6
 - Enabling authentication with REST API..... 6
 - Link to Proxy Server Documentation..... 7
- Chapter 3. Microsoft Windows installation..... 8**
 - Licensing..... 8
 - Installation prerequisites.....8
 - Starting and stopping Unica Link..... 9
 - Accessing Unica Link..... 9
 - Customizing the installation..... 9
 - Customizing the Redis and Link Database..... 10
 - Customizing Link URL and Selecting HTTPS protocol..... 10
 - Installing connectors..... 11
 - Setting up Campaign..... 11

Setting up Journey.....	11
Uninstalling Link.....	12
Verify that the following ports are available on your machine.....	12
Installing Apache Kafka-link.....	15
Post-installation steps.....	15
Starting Kafka-link.....	15
Kafka-link logs.....	15
Installing Link Offline.....	16
Troubleshooting the installation.....	16
Chapter 4. Linux Installation.....	17
Link Utility.....	17
configure Command.....	17
install Command.....	18
start Command.....	19
status Command.....	19
stop Command.....	20
logs Command.....	21
uninstall Command.....	21
Installation of Unica Link in Native Mode.....	22
Installation of Unica Link in Docker Mode.....	26
Chapter 5. Setting up Journey and Campaign for Unica Link.....	28
Setting up Journey from HCL Link.....	28
Setting up Campaign.....	28
Setting up Journey to send messages to Kafka topics.....	29
Chapter 6. Logging.....	30

Link Logs.....	30
Runtime REST API logs.....	30
Kafkalink logs.....	30
Making logs verbose.....	31
Chapter 7. Enabling Unica Link to run on HTTP protocol for Windows and Linux	
Installation.....	32
Chapter 8. Configuring SSL Certificates for Link.....	39
Configuring SSL certificates for Link web UI.....	40
Configuring SSL certificates for the Link Apache Tomcat application server.....	41
Configuring SSL certificates for Link Java WEB Client.....	43
Configuring SSL certificates for Link using command line tool.....	46
Chapter 9. Encrypting the password in the properties file.....	49
Chapter 10. Troubleshooting.....	50
Troubleshooting: Installation.....	50
Licensing.....	51
Runtime REST API server missing in Link.....	51
Chapter 11. Known issues.....	53
Deleting functionality for Journey and Campaign applications.....	53

Chapter 1. Introduction to Unica Link

Unica Link is a component that can be installed separately from the Unica application.

Unica Link provides both design-time and runtime services. It is invoked at design-time when an administrator configures a Link connection, or when you configure a process box in the Unica Campaign application, or a touchpoint in the Unica Journey application. These configurations are stored in the Unica Link MongoDB repository.

The Campaign application communicates with Unica Link in these ways:

- Passes input and output data by use of **CSV** files written to a shared location.
- Invokes the Link APIs to send data to a delivery engine or other resource.

Apache Kafka-link

The Journey application communicates with Link by using Apache Kafka messages.

The Kafka component of Unica Link is named 'Kafka-link'. Kafka-link is installed and configured only if you are using Unica Link with Journey.

Unica Link Docker, Native, and Microsoft Windows installation

Unica Link can be installed and deployed by using either the Native installer, the Docker installer, or the Microsoft Windows installer. The same components are installed in either case.

If you are using Docker for other applications or components, use the Docker installer to create and run the Unica Link containers. If Docker is not available, or if it is not currently used, then install Unica Link by using the native installer. Later sections in this documentation describes the installation in Native mode, Docker mode, and Windows mode.

Unica Link must be configured to point to the Unica installation, to set the security methods to use for API calls (for example, HTTP and HTTPS), and the Kafka server details (if you are using Unica Journey). Some of these parameters need to be specified before installation,

but they can also be modified after the installation is complete. The configuration of Link differs between Docker, Native, and Windows installation and details are provided in this documentation, under separate sections for each mode.

Before you install Unica Link

These conditions must be met before you install Unica Link:

- Determine whether you want to install on Docker or by using the native installer.
- Check that pre-requisites are satisfied.
- Perform pre-installation configuration to set parameters for the installation.
- Run the Docker or native installer.
- Optionally modify configuration parameters once installation is complete.
- Configure Unica applications to reference the Link installation.

Chapter 2. Unica Link prerequisites

Ensure that these prerequisites are met before installing Unica Link.

Unica Link upgrade

To upgrade Link, the user must uninstall the previous version of Link using the following commands or steps:

Platform	Commands/Steps
Native	./Link uninstall
Docker	./Link uninstall
Microsoft Windows	<p>To uninstall Link, complete the following steps:</p> <ol style="list-style-type: none">1. Open the command line interface.2. Change the current folder to <code>C:\HCL\Link_<version_number>\DesignServer</code>.3. To stop and uninstall Link, run the following command <code>clean.bat</code>.4. Manually delete the folder <code>DesignServer</code> from the location <code>C:\HCL\Link_<version number>\</code>.5. Uninstall the Windows Link installation completely (Link comes with Link runtime installation, uninstall runtime component also) by performing the following steps:<ul style="list-style-type: none">• Click Windows Start button.• In the Search box, type <code>Add or Remove Programs</code>.• Select HCL Link <Version_Number> uninstall.

After successful uninstallation, install the new version of Link. For more details, see [Installation of Unica Link in Native Mode \(on page 22\)](#) and [Installation of Unica Link in Docker Mode \(on page 26\)](#).



Note: Unica Link is currently supported only on the Chrome browser.

Installation prerequisites for Linux

You need Linux for Native and Docker Installation. The system requirements for Linux are as follows:

Disk space and memory:

- **Disk space** – The minimum amount of disk space required is 8 GB. The amount of disk space required in a production environment varies based upon the solution implementation.
- **Memory** – The minimum amount of memory required is 8 GB. The amount of memory required in a production environment varies based upon the solution implementation.

Recommended development/production environment:

- 4 CPUs,
- 8 GB RAM per CPU so 4 CPUs x 8 GB/CPU = 32 GB RAM
- local disk of 40 to 60 GB

The Design server and the Runtime server can be on the same box.

For production: The amount of disk space or memory required in a production environment varies based upon the solution implementation.

For MongoDB: Minimum version requirement for MongoDB is 4.0.5 and Link only stores design artifacts there, so the required storage is minimal.

Docker installation prerequisites

Ensure that the following system requirements are met prior to performing the Docker installation of Unica Link.

- Linux dockerized container - "Docker Engine" is required in order to run Docker containers.
- Tomcat Version 9.0.14 - Automatically downloads.
- MongoDB Version 4.0.5 - Automatically downloads.
- Redis cache - Automatically downloads.

These prerequisites must be satisfied before you perform a Docker installation of Unica Link:

- Kafka must be installed and request and reply topics must be created with the appropriate topic names specified in the **kafalink.properties** file.
- Docker must be installed and running.
- Any previous installation of Unica must be fully removed by using the **uninstall** command of the **./Link uninstall script**.

Native installation prerequisites

Ensure that these system requirements are met before you perform the Native installation of Unica Link:

- Linux Red Hat 7 - RHEL 7+
- Tomcat version 9.0.26 - Prepackaged in the installer. You do not need to download or acquire.
- MongoDB version 4.0.5 - You must perform a manual installation and configuration of MongoDB. See the instructions on the MongoDB web site.

In addition, the following are required:

- Configuration of `hip-server`
- Configuration of `hip-rest`
- Kafkalink (Native)

Microsoft Windows installation prerequisites

Before you install Link on Microsoft Windows, install Mongo DB Server version 4.0.5 (or higher) on the same computer. By default, Link and MongoDB server is expected to run on the same computer.

If you want Link and MongoDB Server to run on different computers, you must configure the Link installation configuration settings. To change the MongoDB server host and port configuration settings, edit the `config.yaml` file, in the `C:\HCL\Link_<version_number>\config\config.yaml` location, to match your environment.

Link installation prerequisites

To use Unica Link, you must install Link Runtime server.

If you want to use the non-default installation, customize the installation using the instructions described in the Customizing installation section of this documentation.

Enabling authentication with REST API

Before installing Link or REST API, first enable authentication for the REST API execution time.

This enables authentication and enforces security checking before the flow is executed. The REST API call is authorized by accessing the Link user repository using the REST API supplied user and password, or security token. To enable the authentication mechanism, edit the file:

- For native installation: `<Link_install>/config.yaml`
- For docker installation: `<Link_install>/config.yaml`
- For Windows installation: `C:\HCL\Link_<version number>\config
\config.yaml`

and then change the settings in **authentication.enabled** as shown below:

authentication.enabled=true

Link to Proxy Server Documentation

Click on the following link to access Proxy Server documentation in Link 1.1.5:

https://help.hcltechsw.com/hcllink/1.1.5/proxy_server/concept/c_proxy_server_top_lvl.html

Chapter 3. Microsoft Windows installation

This section describes how to install Link using the Windows Installer.

1. Install Unica Link runtime using the Windows Link installer (do not provide the Flexnet license).
2. Open command prompt with administrator privileges, access the Link installation location, and execute the following command:

```
Copy C:\<Link installation location>\.passport.platform to C:\<Link installation location>\.passport
```

3. Enable the Unica application to embed the Link UI. Customize the Link configuration by changing the value of the **HIP_XCONTENT_POLICY_URLS** parameter in `C:\HCL\Link_<version-number>\config\config.yaml`. The value must be set to a coma separated list of URLs that point to Unica Journey and Unica Campaign WEB application servers. For example, the URLs can be `http://ipaddress:port` or `http://hostname:port` which point to Unica Journey and Unica Campaign WEB application servers. To apply the changes, run the Link client update utility `C:\HCL\Link_<version-number>\DesignServer\updateclient.bat` after customization. For more information, see [Customizing the installation \(on page 9\)](#).
4. If the URLs are not set correctly Unica cannot access Link in iframe and you will see error while loading Link in iframe.

Licensing

1. From the `<Link installer>\hipmodules`, edit the `platform.properties` file.
2. Set to the base URL of the Unica Platform server. For example, `http://myserver:7001`.
3. Provide the credentials.

Installation prerequisites

Unica Link requires the Link Runtime server to be installed.

If you want to use the non-default installation, customize the installation using the instructions described in the Customizing installation section of this documentation.

Starting and stopping Unica Link

After installation, to start and stop Link, complete the following steps:

1. In the Command window, change the current folder to `C:\HCL\Link_<version-number>\config`.
2. To start the server run the `start.bat` command. The server may require several minutes, depending on the performance of your computer.
3. To stop Link, run the `stop.bat` command. The server should stop in a few seconds.

Accessing Unica Link

After the Link is installed, user can access the server by pointing the Web browser to local URL `https://localhost:<port>/login`.

To login for the first time, use username `admin` and the initial administrator password which is stored in `c:\HCL\hipfiles\adminpassword.txt`.

After you login for the first time, change the default password of the admin account for security reasons.

Customizing the installation

You can customize Link by editing settings in the file `C:\HCL\Link_<version-number>\config\config.yaml`.

Review the file content for detailed description of all customization options. Post installation, if you change any of the configuration settings to activate the change, you must reinstall Link by running `clean.bat` followed by `install.bat`.

You can modify the Link UI Client configuration after installing Link without the need to reinstall the product by editing settings in this file: `C:\HCL\Link_<version-number>\config\config.yaml`.

To change any of the Link UI Client specific parameters, modify the related settings in the `config` file and save the changes. Then run the Link UI client update utility `C:\HCL\Link_<version-number>\config\updateclient.bat` to activate the changes.

For example, to enable HTTP transport for the Link UI:

1. Change the Link UI specific parameters as defined in the file `config\config.yaml`.
2. Change the `HIP_HTTP` parameter value to true:

```
set HIP_HTTP=true
```

3. Open an Administrative terminal window and change current folder to DesignServer:

```
cd C:\HCL\Link_<version>\DesignServer\
```

4. Run the Link UI client update utility.

```
updateclient.bat
```

5. Access the Link UI running on `<my Link Server hostname>` server using the following http URL:

```
http://<my Link Server hostname>/login
```

Customizing the Redis and Link Database

Link installation includes Redis server. If you want to use a preinstalled Redis server, you must customize your Link installation.

Redis customization instructions are written in the `config.yaml` file. Locate the `redis:` section, and modify Redis host and port settings to match your environment. Then restart the Link application.

If you want to customize Redis or MongoDB database, change `REDIS` or `MONGO` settings.

Customizing Link URL and Selecting HTTPS protocol

The Link default URL: `https://localhost:4443/login` uses HTTPS protocol on port 4443.

If you want to use a non-default port and HTTPS protocol, you must update the client inbound settings in the configuration file `config.yaml`.

- port
- protocol
- host

For detailed instruction see the `config.yaml` file.

Installing connectors

Use these steps to install connectors:

1. Copy the connector ZIP files and the content of the `apps` folder, from the location `C:\HCL\\` to the location mentioned for the `modules` parameter in the `%DTXHOME%\config\config.yaml` script.
2. Restart Link.
3. Refer the file `config.yaml`, present in the `<link installation location>\config`, to see the folder path that is set for the `persistence:files` parameter. The default path is `C:\data`. If the default path is `C:\data`, the tmp folder file path is `C:\data\tmp`.

Setting up Campaign

Steps to set up Campaign.

1. Edit the `campaign.properties` file. For more information, see [Setting up Campaign \(on page 28\)](#).

Setting up Journey

Steps to set up Journey.

1. Edit the `journey.properties` file. For more information, see [Setting up Journey from HCL Link \(on page 28\)](#).

Uninstalling Link

To uninstall Link, complete the following steps:

1. Open the Command window, and change the current folder to `C:\HCL\Link_<version-number>\DesignServer`.
2. Run the **clean.bat** command to stop and uninstall Link.
3. Uninstall Link runtime by performing the following steps:
 - Click Windows Start button.
 - In the **Search box**, type `Add or Remove Programs`.
 - Select **HCL Link <Version_Number> uninstall**.
4. Manually delete the `C:\HCL\Link_<version number>\DesignServer` folder.

Verify that the following ports are available on your machine

The following table lists the required ports and the corresponding Link configuration properties:

Default port	Used by Link features	Configuration file location (relative to the Link installation folder)	Configuration property name
4443	Link server	<code>config\config.yaml</code>	<code>/client/inbound/port</code> used with <code>/client/inbound/protocol</code> Formerly: <code>HIP_HTTPS_PORT</code> , <code>HIP_HTTP_PORT</code>
80	Link server	<code>config\config.yaml</code>	<code>/client/inbound/port</code> used with /

Default port	Used by Link features	Configuration file location (relative to the Link installation folder)	Configuration property name
			<code>client/inbound/protocol</code> Formerly: <code>HIP_HTTPS_PORT, HIP_HTTP_PORT</code>
8080	Runtime and Link server	<code>config\config.yaml</code>	<code>/tomcat/inbound/ports/http</code> <code>/rest/inbound/ports/http</code> <code>/keycloak/serverUrl</code> (used only if the Keycloak feature is enabled, <code>/keycloak/enabled</code>) Formerly: <code>authentication.keycloak.serverUrl</code> <code>HIP_KEYCLOAK_SERVER_URL</code>
8443	Runtime and Link server	<code>config\config.yaml</code>	<code>/tomcat/inbound/ports/https</code> <code>/server/inbound/ports/https</code> <code>/client/outbound/server/port</code>

Default port	Used by Link features	Configuration file location (relative to the Link installation folder)	Configuration property name
			<code>/rest/inbound/authentication/server</code> Formerly: <code>authentication.server</code> <code>HIP_SERVER_HOST</code>
8005	Runtime and Link server	<code>config\config.yaml</code>	<code>/tomcat/inbound/ports/shutdown</code>
15901	Event Reporter	<code>config\config.yaml</code>	<code>/rest/eventReporter/port</code> Formerly: <code>event.logger.port</code>
		<code>eventreporter.properties</code>	<code>LISTENING_PORT</code>

The Link Windows installation opens a dialog that allows you to configure the Link design time (Link UI) ports. Typically, the Link UI ports are ports 4443 and 80. The Windows installation verifies that the ports are available on the machine. The installation does not continue if these ports are not available.

Also, the Link Windows installation opens a dialog that allows you to configure Link application server ports (typically ports 8080, 8443 and 8005). The Windows installation verifies that the selected ports are available on the machine. The installation does not continue if those ports are not available.

The Event Reporter port (port 15901) must be manually verified to ensure that it is available.

Installing Apache Kafka-link

These instructions describe how to install Apache Kafka-link on Windows.

Post-installation steps

Steps for editing the Kafka-link properties file

Follow these steps to edit the Kafka-link properties file:

1. Go to this location to edit the files: `C:\<Link installation location>\kafka-link`
2. Provide broker IP with port. For example: `brokers=localhost:9092`
3. Provide the Hip URL of the Link for your Windows local computer Link installation: `hip-url=https://localhost:8443/hip-rest`
4. The name of the request topic: `request-topic=<Topic Name>` For example - `request-topic=OUTGOING_MESSAGES`
5. The name of the response topic: `response-topic=<Topic Name>` For Example - `response-topic=INCOMING_RESPONSES`
6. Save the file.

Starting Kafka-link

Steps for starting Kafka-link.

Follow these steps to start Kafka-link:

1. Open a command prompt, and cd to the following location: `C:\<Link installation location>\kafka-link\`
2. Press **Enter**.
3. Enter the following command: `"kafkalink.bat -P kafkalink.properties"`
4. Press **Enter** to start Kafka-link.

Kafka-link logs

Location information about Kafka-link logs.

Installing Link Offline

For installing Link offline, Internet access is not required. Link offline installation can be done, only with MongoDB preinstalled.

Troubleshooting the installation

In the case of Link Install, Start, or Stop errors, review the logs files from the following folders:

Folder type	File / Folder Path	Description
Main Folder	<pre><Link-install>\Design-Server\install<Time-DateStamp>.log</pre> <pre><Link-install>\Design-Server\start<TimeDate-Stamp></pre> <pre><Link-install>\Design-Server\stop<TimeDateS-tamp></pre>	Contains information about the issues and steps required for resolving issues
Additional Folders	<pre>C:\HCL\Link_<ver-sion-number>\restapi\tomcat\server\logs</pre>	Link, Tomcat Application server files
	<pre>c:\HCL\Link_<ver-sion-number>\Design-Server\client\daemon*.log</pre>	Link, WEB NodeJS Application server files
	<pre>C:\HCL\Link_<ver-sion-number>\restapi\redis\logs</pre>	Link, Redis server files

Chapter 4. Linux Installation

You can install Unica Link in two modes:

- [Native Installation \(on page 22\)](#)
- [Docker Installation \(on page 26\)](#)

There is a consistent command line interface for the installation and management in either modes. The Link script is the entry point for the CLI.

Link Utility

After extracting the HCL Link package, you can execute the Link script to begin installation.

For more information on installation, see [Installation of Unica Link in Native Mode \(on page 22\)](#) and [Installation of Unica Link in Docker Mode \(on page 26\)](#).

The Link script supports the following commands:

- configure
- install
- start
- status
- stop
- logs
- uninstall

configure Command

The configure command is required before using any other command. This command is used to specify the installation type, the Unica integration, and the user or group that the installation is completed as. You must specify the correct option for `--integration` at a minimum.

Options

- `--type <native|docker>` - Specify the installation type. Default: `native`.
- `--integration <unica>` - Specify your integration.
- `--license-file` - This option is intended for standalone customer use only. Do not use if you have received Link as an embedded solution.
- `--user` - The user to install Unica Link as. This will determine which user has ownership of the processes and directories. This must be an already existing user. Default: `current user`.
- `--group` - The group to install Unica Link as. This provides you with flexibility to share Unica Link directories with a group of admins. Default: `current group`.

Notes

You cannot change the integration type after you run install. If you notice an error after installation, you must extract the Unica Link package into a new directory and start over.

install Command

The behavior of the install and uninstall commands is extremely similar. Both commands will act upon the default components if none are specified as options. You can finetune the scope of each action by listing individual components.



CAUTION: Embedded installations generally require all default components running for proper functionality. Do not use these options if you do not understand them fully.

Options

- `--runtime` - Runtime REST API.
- `--server` - Link Server.
- `--kafka-link` - Kafka Link.
- `--apps` - Apps in `integration-context/apps`.
- `--connectors` - Connectors in `integration-context/connectors`.

Notes

Connectors are not included in the Link package by default. You must create the `<link_package>/integration-context/connectors` directory and place your connector files there before connector installation. You can update app property files by re-running the installation (with the `--apps` option).

start Command

The behavior of the start and stop commands is extremely similar. Both commands will act upon the default components if none are specified as options. You can fine tune the scope of each action by listing individual components.



CAUTION: Warning: Embedded installations generally require all default components running for proper functionality. Do not use these options if you do not understand them fully.

Options

- `--runtime` - Runtime REST API.
- `--server` - Link Server.
- `--kafka-link` - Kafka Link.

status Command

The status command retrieves the status of each Unica Link component. Though the usage is similar between the native and docker installation types, the possible states are different.

Options

- `--runtime` - Runtime REST API.
- `--server` - Link Server.
- `--kafka-link` - Kafka Link.

Possible States - Native

- Not Installed
- Stopped
- Running
- Unknown

Possible States - Docker

- Not Installed
- Created
- Dead
- Exited
- Paused
- Running
- Restarting

stop Command

The behavior of the start and stop commands is extremely similar. Both commands will act upon the default components if none are specified as options. You can fine tune the scope of each action by listing individual components.



CAUTION: Warning: Embedded installations generally require all default components running for proper functionality. Do not use these options if you do not understand them fully.

Options

- `--runtime` - Runtime REST API.
- `--server` - Link Server.
- `--kafka-link` - Kafka Link.

logs Command

The logs command prints the primary log of the chosen component. Because the log files may increase in size, you should pipe the output of this command into a tool so you can navigate better.

Options

- `--runtime-app` - Runtime REST API.
- `--server-app` - Link Server - Backend.
- `--client` - Link Server - User Interface.
- `--kafka-link` - Kafka Link.

uninstall Command

The behavior of the install and uninstall commands is extremely similar. Both commands will act upon the default components if none are specified as options. You can finetune the scope of each action by listing individual components.



CAUTION: Embedded installations generally require all default components running for proper functionality. Do not use these options if you do not understand them fully.

Options

- `--runtime` - Runtime REST API.
- `--server` - Link Server.
- `--kafka-link` - Kafka Link.
- `--apps` - Apps in `integration-context/apps`.
- `--connectors` - Connectors in `integration-context/connectors`.

Notes

Connectors are not included in the Link package by default. You must create the `<link_package>/integration-context/connectors` directory and place your

connector files there before connector installation. You can update app property files by re-running the installation (with the `--apps` option).

Installation of Unica Link in Native Mode

Dependencies

- `bash` (Bourne Again SHell)
- `ksh` (KornShell)
- `libnsl`
- GNU `sed`
- MongoDB server

Accessing Unica Link on HTTP through load balancer HTTPS URL

This section describes the accessing of Unica Link on HTTP through load balancer HTTPS URL and internally routed to link by using nginx or any other load balancer.

For accessing Unica Link through load balancer URL, you need to set three context rules as follows:

- `hip-rest` - It must be redirected to the rest port.
- `tx-server` - It must be redirected to the client/node port.
- `link client` - It must use the client port.

See note for the `tx-server` context rules:

- The nginx `tx-server` context communicate to link node via the Link node/client port, not directly to link tomcat through the tomcat port (Link server).
- The connection from the Link node/client to the backend server will run through the settings which we have under `client` → `outbound` in `config.yaml` (Link node/client redirection to tomcat/link server).

Native System Configuration

Native system configuration includes:

- changing the temporary directory, or
- removing `noexec` from the `/tmp` directory.

Changing the Link Temporary Directory

To change the Link temporary directory, complete the following steps:

1. Create a new directory. Ensure that the `noexec` option is not used for mounting of the location.
2. Check the output of the mount. For example, run the following commands:

```
mkdir /opt/hiptmp
```

```
chmod 777 /opt/hiptmp
```

3. Add a new line to `tomcat-context/install/restapi/tomcat/setenv.sh` to specify the new directory. For example:

```
export JAVA_OPTS="$JAVA_OPTS -Djava.io.tmpdir=/opt/hiptmp"
```

4. For the long and short workers, modify the existing lines in `config.yaml`. For example:

```
longTaskProcessJvmOptions="-Xmx2g -Djava.io.tmpdir=/opt/hiptmp"
```

```
shortTaskProcessJvmOptions="-Djava.io.tmpdir=/opt/hiptmp"
```

5. Restart Link.

Changing the Mount Option for `/tmp`

The `/tmp` directory must be mounted with `noexec`. To verify, run the following commands:

```
mount | grep /tmp
```

```
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,noexec,relatime)
```

If `/tmp` directory is already mounted with `noexec`, you can change the mount option for `/tmp` directory. To change the the mount option for `/tmp` directory, complete the following steps:

1. Ensure that you are using root user privileges.
2. Edit `/etc/fstab` and remove `noexec` from the mount options of `/tmp`.
3. Either:
 - a. Reboot the system, or
 - b. Run the following command:

```
mount -o remount /tmp
```

Configuration

To configure most properties of Link, use the file `config.yaml`.

Review the `config.yaml` file before you perform the installation to confirm that all default options are as expected.

To reconfigure after installation, update those files, and restart Link.

You will notice several directories specified in the configuration files. If you, or the group you install as, lack Read/Write access to these directories, contact a system administrator to create the directories and transfer ownership.

If you are installing as a non-root user, use ports above 1024 to avoid permission issues.

The default ports are:

```
Tomcat (REST API + Server): HTTPS 8443, HTTP 8080  
UI Client: HTTPS 4443
```

Installation

Read the Link Utility documentation before following the installation steps. You may need to adjust commands according to your use-case.

1. The following commands are intended for a quick-start scenario:
 - a. Use the following command to generate configuration:

```
./Link configure --generate
```

- b. Change the Unica Journey/Unica Campaign/Unica Platform properties under `<Link installer>/integration-context/apps/` and place

the connectors under `<Link installer>/ integration-context/ connectors`.

- c. Enable the Unica application to embed the Link UI. Customize the Link configuration by changing the value of the **contentSecurityPolicy** parameter in `<Link installer>/Config.yaml` file. The value is list of URLs that point to Unica journey and Unica Campaign web application servers. For example, the URLs can be `http://ipaddress:port` or `http://hostname:port` which point to Unica journey and Unica Campaign web application servers.

If the URLs are not set correctly, Unica cannot access link in iframe and you will see error while loading link in iframe.

- d.

```
./Link configure --type native --integration unica
```

- e.

```
./Link install
```

- f. Change the kafkalink properties under `<Link-install>/integration-context/install/`.

- g. Either start all components or continue to the troubleshooting section for any errors.

```
./Link start
```

2. When Link is up and running, navigate to the location that is set for the environment variable files and create a new folder `tmp`.

Refer to the file `config.yaml`, present in the installation location, to see the folder path that is set for the files parameter. The default path is `/opt/hip-rest`. If the default path is `/opt/hip-rest`, the `tmp` folder file path is `/opt/hip-rest/tmp`.

Installing Link Offline

For installing Link offline, Internet access is not required. Link offline installation can be done, only with MongoDB preinstalled.

Installation of Unica Link in Docker Mode

Dependencies

- Docker



Important: Unica Link does not offer Docker installation compatibility on Red Hat platforms version 8 and beyond.

Configuration

To configure most properties of Link, use the file `config.yaml`.

Review the `config.yaml` file before performing the installation to confirm all default options are as expected.

To reconfigure after installation, update those files and reinstall Link.

You will notice several directories specified in those configuration files. If you, or the group that you install as, lacks Read/Write access to these directories, contact a system administrator to create the directories and transfer ownership.

If you are installing as a non-root user, use ports above 1024 to avoid permission issues.

The default ports are:

```
REST API: HTTPS 9443, HTTP 8080
```

Installation

Read the Link Utility documentation before following the installation steps. You may need to adjust commands according to your use-case.

1. The following commands are intended for a quick-start scenario:
 - a. Use the following command to generate configuration:

```
./Link configure --generate
```

- b. Change the Unica Journey/Unica Campaign/Unica Platform properties under `<Link installer>/ integration-context/apps/` and place

the connectors under `<Link installer>/ integration-context/ connectors`.

- c. Enable the Unica application to embed the Link UI. Customize the Link configuration by changing the value of the `contentSecurityPolicy` parameter in `<Link installer>/Config.yaml` file. The value is list of URLs that point to Unica journey and Unica Campaign web application servers. For example, the URLs can be `http://ipaddress:port` or `http://hostname:port` which point to Unica journey and Unica Campaign web application servers.

If the URLs are not set correctly, Unica cannot access link in iframe and you will see an error while loading the link in iframe.

- d.

```
./Link configure --type docker --integration unica
```

- e.

```
./Link install
```

- f. Change the `kafkalink` properties under `/opt/hip-rest/config/`.

- g. Either start all components or continue to the troubleshooting section for any errors.

```
./Link start
```

Installing Link Offline

Internet access is required for the installation of Link.

Chapter 5. Setting up Journey and Campaign for Unica Link

The Unica system IPADDRESS contains the following setup:

- Kafka broker
- Unica Journey where Journey engine is running
- Unica Campaign

Setting up Journey from HCL Link

To set up Journey follow these steps:

1. Go to the apps folder: `<link_install>/link-context/apps`.
2. Update the `journey.properties` file as follows:
 - a. Set the file to point to the Journey server. Ensure that a slash does not exist at the end of the URL: `login_url=http:// IPADDRESS:PORT/journey/api/login` `data_definition_url=http:// IPADDRESS:PORT/journey/api/datadefinitions/point`
 - b. Specify Journey credentials: `username=<username>password=<password>`

Setting up Campaign

To set up Campaign follow these steps:

1. Go to the apps folder: **UnicaLink_<n.n.n.n>**.
2. Update the **campaign.properties** file as follows:
 - a. Set the file to point to the Campaign server. Ensure there is not a slash at the end of the URL: `login_url=http:// IPADDRESS:PORT/unica/api/manager/authentication/login` `data_definition_url=http:// IPADDRESS:PORT/Campaign/api/campaign/rest/v3/link/field-information?actionId`
 - b. Specify campaign `headers:m_user_name=<username>m_user_password=passwordapi_auth_mode=<manager>`

Setting up Journey to send messages to Kafka topics

In order for the Unica Journey application to send messages to Kafka topics, this setup is required.

For Kafkalink to work from Unica Link, perform these steps:

1. Update the **kafkalink.properties** file located in this location: `<link_install>/integration-context/install`.
2. Update the following:
 - List of one or more Kafka brokers= **IPADDRESS:9092**
 - The base URL for the HIP runtimehip-url=`https://IPADDRESS:<PORT>/hip-rest/`
 - The name of the request topicrequest-topic=**OUTGOING_MESSAGES**
 - The name of the response topicresponse-topic=**INCOMING_RESPONSES**
 - The directory where files are created before sending to HIPdata-directory=**/tmp**
3. Update the **platform.properties** file located under `<link_install>/integration-context/apps`. Set the following properties:
 - Set the base URL of the Unica Platform server. For example: `http://myserver:7001`. Do not specify a slash (/) after the URL.
 - base_url=`http://IPADDRESS:<PORT>`
 - Credentials: username=**asm_admin** password=**password**

Chapter 6. Logging

Logs are produced for your use by Link, the Runtime REST API, and Kafkalink.

Link Logs

These are the logs produced by Link:

- For Native Installer:
 - **./Link logs --runtime-app**
 - **./Link logs --client**
- For Docker Installer:
 - **docker logs hip-server**
 - **docker logs hip-rest**

Runtime REST API logs

These logs are produced by the Runtime REST API:

Native Install:

- **./Link logs --runtime-app**

Docker Install:

- **docker logs hip-rest**

Kafkalink logs

These logs are produced by Kafkalink:

Native Install:

- **./Link logs -kafka-link**

Docker Install:

- **docker logs kafka-link**

Making logs verbose

To make logs more verbose you can modify the **config.yaml** file. Modify the file as follows:

1. For Native or Docker:

- Native Install: Go to: `<install_dir>/config.yaml`. By default, `<install_dir>` is in `<installation_tar>/tomcat-context/install`
- Docker install: Edit **config.yaml** in the **hip-rest** Docker container by using the command:

```
docker exec -it hip-rest vi /opt/runtime/config.yaml
```

2. Change the value of **logging:level: TRACE**.

3. For Native or Docker:

- Native Install: `./Link stop` and `./Link start`
- Docker Install: **docker restart hip-rest**

Chapter 7. Enabling Unica Link to run on HTTP protocol for Windows and Linux Installation

Enabling Unica Link for Microsoft Windows

This documentation describes how to enable the Unica Link to run on `HTTP` protocol before running the link install command for the Microsoft windows installation.

To enable the Unica Link to run link server and rest on http port 8080 and client on http port 4443. Modify below files and complete the following steps:

1. Modify `config.yaml`.

```
set HIP_HTTP=true
set HIP_HTTPS_PORT=443
set HIP_HTTP_PORT=4443

::HIP server host location

set HIP_SERVER_HOST=localhost:8080

Modify config.yaml -

authentication.server=http://localhost:8080tomcat:
  inbound:
```

2. Run `install.bat` command
3. Set the environment variable `HIP_SERVER_USE_HTTP= True`, and complete the following steps:

- a. Click **This PC**.
 - b. Select properties and click **Advanced System Settings**.
 - c. Under the **Advance** tab, click **Environment Variables** and add a system variable as mentioned in *Step 3*.
4. Run Start.bat command.
 5. Check client log for the following message:

```
<Link installation
  folder>\DesignServer\client\daemon\hcllinkwebui.out.log

Server listening at http://0.0.0.0:4443
Using server backend at url: http://localhost:8080
```

6. Launch the Link UI and log in.

Enabling Unica Link for Linux Native and Docker for Port 8080

This documentation describes how to enable the Unica Link to run on HTTP protocol before running the link install command for the Native and Docker installation.

To enable the Unica Link to run on [HTTP](#) 8080 protocol for the Native and Docker installation, complete the following steps:

1. Open the `config.yaml` file.
2. In client configuration section, for inbound and outbound connections, change the protocol properties from `https` to `http` as shown below in the following example:

```
client:
  # Settings for inbound connections to the client such as when
  accessing
  # via a web browser.
  inbound:
```

```

# Protocol for inbound connections. Valid options are [http,
https].

# Previously: HIP_HTTP
protocol: "http"

# Port to listen on for inbound connections. This value must be
>1000

# to run the client process as a non-root user.
# Previously: HIP_HTTPS_PORT, HIP_HTTP_PORT
port: 4443

# Settings for outbound connections from client
outbound:

# Connection from client to backend server
server:

# Protocol for connection. Valid options are [http, https].
# Previously: HIP_SERVER_USE_HTTP
protocol: "http"

```

3. In the Rest configuration section perform these steps:

a. Change the settings in `authentication.enabled` as follows:

```
authentication.enabled=true
```

b. Change the server property from `https://localhost:8443` to `http://localhost:8080` as shown in the following example:

```

rest:
  authentication:
    # Specify if inbound API calls should be authenticated.
    # Previously: authentication.enabled
    enabled: true
    # The authentication server MUST point to the Server
instance configured

```

```

    # for this installation. The format is
    {protocol}://{host}:{port}

    # [NATIVE] Default:
    'https://localhost:{.server.inbound.ports.https}'

    # [CONTAINERIZED] Default: 'https://hip-server:8443'

    # Previously: authentication.server

    server: "http://localhost:8080"

```

4. In the tomcat configuration section, the default configuration for `http` port is 8080 and `https` port is 8443. Any mismatch found in default configuration will require changing ports as shown in the following example:

```

tomcat:
  inbound:
    # [NATIVE-ONLY] Ports to listen on for connections. Neither
    protocol can be
    # disabled directly from this configuration file. To do so,
    complete
    # installation first then go to the Tomcat configuration files for
    advanced
    # options.
    # To set ports for docker installations use
    the .rest.inbound.ports and
    # .server.inbound.ports to configure on a per-container basis.
    Native
    # installations share the same Tomcat instance between rest and
    server.
  ports:
    http: 8080
    https: 8443
    shutdown: 8005"

```

5. Save the `config.yaml` file.

6. Execute ./Link install and ./Link start.



Note: The Link Runtime server should also use `HTTP` URL.

Enabling Unica Link for Linux Native for Port 8443

To enable the Unica Link to run on HTTP 8443 protocol for the Native installation, complete the following steps:

1. Open the `config.yaml` file.
2. In client configuration section, for inbound and outbound connections, change the protocol properties from `https` to `http` as shown below in the following example:

```
client:
  # Settings for inbound connections to the client such as when
  accessing
  # via a web browser.
  inbound:
    # Protocol for inbound connections. Valid options are [http,
    https].
    # Previously: HIP_HTTP
    protocol: "http"
    # Port to listen on for inbound connections. This value must be
    >1000
    # to run the client process as a non-root user.
    # Previously: HIP_HTTPS_PORT, HIP_HTTP_PORT
    port: 4443
  # Settings for outbound connections from client
  outbound:
    # Connection from client to backend server
    server:
      # Protocol for connection. Valid options are [http, https].
```



```
# Previously: HIP_SERVER_USE_HTTP
protocol: "http"
```

3. In the Rest configuration section perform these steps:

a. Change the settings in `authentication.enabled` as follows:

```
authentication.enabled=true
```

b. Change the server property from `https://localhost:8443` to `http://localhost:8443` as shown in the following example:

```
rest:
  authentication:
    # Specify if inbound API calls should be authenticated.
    # Previously: authentication.enabled
    enabled: true
    # The authentication server MUST point to the Server
instance configured
    # for this installation. The format is
{protocol}://{host}:{port}
    # [NATIVE] Default:
'https://localhost:{.server.inbound.ports.https}'
    # [CONTAINERIZED] Default: 'https://hip-server:8443'
    # Previously: authentication.server
    server: "http://localhost:8443"
```

4. In the tomcat configuration section, change the `http` port to `8443` and `https` port to `8444` as shown in the following example:

```
tomcat:
  inbound:
    # [NATIVE-ONLY] Ports to listen on for connections. Neither
protocol can be
```

```
# disabled directly from this configuration file. To do so,
complete
# installation first then go to the Tomcat configuration files for
advanced
# options.
# To set ports for docker installations use
the .rest.inbound.ports and
# .server.inbound.ports to configure on a per-container basis.
Native
# installations share the same Tomcat instance between rest and
server.
ports:
  http: 8443
  https: 8444
  shutdown: 8005"
```

5. Save the `config.yaml` file.
6. Execute `./Link install` and `./Link start`.



Note: The Link Runtime server should also use `HTTP` URL.

Chapter 8. Configuring SSL Certificates for Link

After installing Link, setup the custom SSL certificates for the Link web UI and the Link runtime application server.



Note: For the Link web UI to use the certificate, ensure that `/client/inbound/protocol` is set to `https` and it is the default setting.

Create the SSL certificates or use the SSL certificates from a third-party provider (example GoDaddy Inc). PEM is the preferred format for the certificates. If the certificates are not of the PEM format, convert the certificates to PEM format.

Split the SSL certificates into three files. Each file must contain one of the following SSL certificate components:

- Private key
- Server certificate key
- Certificate authority keys

The following table lists the GoDaddy-provided certificates in their corresponding files:

File	Description
Private key file	<p>The file format is PEM and the extension of the file is KEY.</p> <p>The content of the file starts and ends with the following lines:</p> <pre>-----BEGIN PRIVATE KEY----- ... -----END PRIVATE KEY-----</pre>

File	Description
Server certificate key file	<p>The file format is PEM and the extension of the file is KEY.</p> <p>The content of the file starts and ends with the following lines:</p> <pre data-bbox="834 520 1386 751">-----BEGIN CERTIFICATE----- ... -----END CERTIFICATE-----</pre>
Certificate authority (CA) keys	<p>You need these keys to sign off the private keys and public keys. Bundle the keys into a single file or multiple files. The files are not required if the certificates are self-signed.</p> <p>The CA certificates start and stop with the following lines:</p> <pre data-bbox="834 1163 1386 1394">-----BEGIN CERTIFICATE----- ... -----END CERTIFICATE-----</pre>

Configuring SSL certificates for Link web UI

To configure SSL certificates for Link web UI, complete the following steps:

1. Prepare the SSL certificate files.
2. Within the Link client installation folder, copy the certificate files to the `ssl-certificates/public-certificates/` folder. The relative path of the installation root folder for different platforms are as follows:

Microsoft Windows	The client installation folder is <code>DesignServer/client</code> .
Linux (Native)	The client installation folder is <code>node-context/install</code> .
Linux (Docker)	The client installation folder is within the <code>hip-client</code> container at <code>/usr/src/app</code> . Use the <code>docker cp</code> command to copy files from a local folder to the Docker <code>hip-client</code> container.

3. Setup the `config.yaml` `"/client/inbound/ssl"` section to point to the certificate files copied in Step 2.

Example:

```
key: "ssl-certificates/public-certificates/server.key"
cert: "ssl-certificates/public-certificates/server.crt"
ca: "
  ssl-certificates/public-certificates/gd1.crt,ssl-certificates/public
-certificates/gd2.crt,ssl-certificates/public-certificates/gd3.crt "
```

4. Restart the Link application. If your user profile has all the appropriate privileges, open a shell and change the working directory to the installation root.

Configuring SSL certificates for the Link Apache Tomcat application server

If you implement Link on a setup containing Apache Tomcat sever, the implementation works as a Tomcat application.

Apache Tomcat uses a Java keystore to store SSL certificates. To import certificates into the Link application server Java keystore, use the `openssl` utility. For more details related to `openssl` installation, see <https://www.openssl.org/>.

To configure SSL certificates for the Link Apache Tomcat application server, complete the following steps:

1. To export the SSL certificates into pkcs12 format, run the command `open ssl pkcs12`. The system prompts you to provide the password phrase (for example `<my passphrase>`) to protect the generated certificate file. The `-CAfile` command option supplies the CA bundle.

If GoDaddy provides multiple CA PEM files, bundle all CA files (concatenated as text files) into single file named `cabundle.crt`:

```
gd1.crt>> cabundle.crt
gd2.crt>> cabundle.crt
gd3.crt>> cabundle.crt
openssl pkcs12 -export -in sever.crt -inkey server.key -out
dtxtomcat.p12 -name dtxtomcat -CAfile cabundle.crt -caname root
```

2. Depending on the platform, access the keytool application from the provided location:

Microsoft Windows	<code>C:\HCL\Link_<version>\java\bin\keytool.exe</code>
Linux-based OS	<code>tomcat-context/install/java/bin/keytool</code>

To generate java keystore, run the following command:

```
keytool.exe -importkeystore -deststorepass <my passphrase>
-destkeypass <my passphrase> -destkeystore dtxtomcat.keystore
-srckeystore dtxtomcat.p12 -srcstoretype PKCS12 -srcstorepass
changeit -alias dtxtomcat keytool -v -list -keystore
dtxtomcat.keystore
```



Note: The value of *<my passphrase>* must match the passphrase value set for the configuration property */tomcat/keystore/password*.

- For Microsoft Windows and Linux-based operating systems, copy the generated Java keystore to the following location:

```
<Link installation folder>/restapi/tomcat/server/dtxtomcat.keystore
```

where *<Link installation folder>* is:

- For Microsoft Windows - C:\HCL\Link_<version>
 - For Linux-based operating systems - tomcat-context/install
- In case of Link installation on Docker environment, copy the keystore file from a local folder to the `hip-rest` container at `/usr/local/tomcat` using the `docker cp` command. The value of *<my pass phrase>* must match the passphrase value in the `/usr/local/tomcat/conf/server.xml` file within the container.
 - Restart the Link application. If your user profile has all the appropriate privileges, open a shell and change the working directory to the installation root.

- In case of Microsoft Windows, execute the following commands:

```
DesignServer\stop.bat
DesignServer\start.bat
```

- In case of Linux-based systems, execute the following commands:

```
./Link stop
./Link start
```

Configuring SSL certificates for Link Java WEB Client

This documentation explains how to configure an SSL certificate for the Link Java WEB client.

**Note:**

- Please note that if you encounter the following error while configuring the HCH runtime server, you will need to import the Link server certificate into the JRE keystore or truststore. This problem occurs when java web clients try to access web servers that have server certificates signed by unknown authority.

```
com.hcl.tx.server.impl.ServiceException: An
error was encountered when invoking an endpoint:
javax.net.ssl.SSLHandshakeException: PKIX path building
failed:
sun.security.provider.certpath.SunCertPathBuilderException:
unable to find valid certification path to requested target.
```

- Please note that if you encounter the following license-related SSL errors when accessing a UNICA platform running on HTTPS, you will need to import the Unica Link platform certificate into the JRE (Java Runtime Environment) keystore or truststore. This problem occurs when java web clients try to access web servers that have server certificates signed by unknown authority.

```
SEVERE [https-jsse-nio-8443-exec-5]
com.hcl.hip.ws.resources.AuthFilter.checkLicense Invalid
license, request GET
https://localhost:8443//tx-server/rest/v1/tx/license_check//pl
atform failed, this Link instance is not licensed to execute
the request.
```

```
SEVERE [https-jsse-nio-8443-exec-5]
com.hcl.hip.ws.resources.AuthFilter.checkLicense This Link
instance is not licensed to execute the REST requests. Error:
Internal Server Error
```

To configure SSL certificates for the Link Java WEB client, complete the following steps:

1. Collect the WEB server certificate by executing the following command:

For Link Server	openssl s_client -connect <Link server hostname>:<PORT>
For Unica Platform	openssl s_client -connect <UNICA platform hostname>:<PORT>

2. Copy the following content from the generated output after running the earlier command and save the certificate in either `.cer` file format:

```
-----BEGIN CERTIFICATE-----
.....
-----END CERTIFICATE-----
```

3. Import the certificate from the file (created in step 2) into JRE (Java Runtime Environment) keystore using the following command:

For Windows

```
sudo keytool -import -trustcacerts -keystore <Link installation
folder>/java/lib/security/cacerts -storepass changeit -noprompt
-alias <Alias> -file <Cert File path>
```

For Linux Native

```
sudo keytool -import -trustcacerts -keystore <Link installation
folder>/tomcat-context/install/java/lib/security/cacerts -storepass
changeit -noprompt -alias <Alias> -file <Cert File path>
```

4. Restart Link.

Configuring SSL certificates for Link using command line tool

To configure the SSL certificates for Link using command line tool to import certificate in PEM format, complete the following steps:

1. Prepare the SSL PEM certificate files.
2. Import SSL PEM certificates into Link environment using `importcertificate` tool.



Note:

- If the SSL certificates are in PEM format, then certificates can be imported into Link environment using `importcertificate` tool.
 - PEM is the preferred format for the certificates. If the certificates are not of the PEM format, convert the certificates to PEM format.
3. Split the prepared SSL certificates into three PEM files. Each file must contain one of the following SSL certificate components:
 - Public Server certificate PEM file
 - Private certificate key PEM file
 - Certificate authority PEM file
 4. Depending on the platform, after installing Link, open a command line with administrative privileges, then change directory to:

Microsoft Windows	cd to the DesignServer installation folder, for example: <pre>cd C:\HCL\Link_1.1.6\DesignServer</pre>
Linux-based OS	cd to Link installation folder, for example: <pre>cd /home/myuser/Link_1.1.6</pre>



Note: Log in to the system as the same user, that have Link installed.

5. Copy the SSL PEM files on the computer where Link is installed, for example:

<p>Microsoft Windows</p>	<p>Private key: <code>c:\ssl-certificates\self-signed-test\server-key.pem</code></p> <p>Public certificate: <code>c:\ssl-certificates\self-signed-test\server-crt.pem</code></p> <p>Certificate authority: <code>c:\ssl-certificates\self-signed-test\self_ca_root.pem</code></p>
<p>Linux-based OS</p>	<p>Private key: <code>/home/myuser/ssl-certificates/self-signed-test/server-key.pem</code></p> <p>Public certificate: <code>/home/myuser/ssl-certificates/self-signed-test/server-crt.pem</code></p> <p>Certificate authority: <code>/home/myuser/ssl-certificates/self-signed-test/self_ca_root.pem</code></p>

6. For Microsoft Windows and Linux-based operating systems, run the command from the following folder location:

- Command for Microsoft Windows from `C:\HCL\Link_1.1.6\DesignServer` folder:

```
.\importcertificates.bat
c:\ssl-certificates\self-signed-test\server-key.pem
```

```
c:\ssl-certificates\self-signed-test\server-crt.pem  
c:\ssl-certificates\self-signed-test\self_ca_root.pem
```

- Command for Linux-based operating systems from `/home/myuser/Link_1.1.6` folder:

```
./Link  
importcertificates /  
home/myuser/ssl-certificates/self-signed-test/server-key.pem /  
home/myuser/ssl-certificates/self-signed-test/server-crt.pem /  
home/myuser/ssl-certificates/self-signed-test/self_ca_root.pem
```

7. Restart the Link application. If your user profile has all the appropriate privileges, open a shell, and change the working directory to the installation root.

- In case of Microsoft Windows, execute the following commands:

```
DesignServer\stop.bat  
DesignServer\start.bat
```

- In case of Linux-based systems, execute the following commands:

```
./Link stop  
./Link start
```

Chapter 9. Encrypting the password in the properties file

To enhance security and safeguard sensitive information, encrypt the password stored in the properties file to add a layer of protection to the data.

To encrypt passwords in the properties file, complete the following steps:

1. In the Link installation directory, locate the `encrypt_password.sh` script.
2. Open the properties file and use the following syntax to encrypt the password:

```
<Field Name>=plain:<TEXT/PASSWORD>
```



Note: For the strings to be encrypted, add the prefix `plain:`.

Example:

```
password=plain:password
```

3. Run the encryption script `./encrypt_password.sh <Properties File Path>`.



Note: Replace `<Properties File Path>` with the absolute path to the properties file you want to encrypt the password in.

For example, if the properties file is available at `/opt/hipmodules/journey.properties`, the command would be as follows:

```
./encrypt_password.sh /opt/hipmodules/journey.properties
```

4. After executing the script, open the properties file. You will see that the encrypted string, or text, has the prefix `encrypted`. For example:

```
password=encrypted<Encrypted Sting>
```

5. Restart Link.

Chapter 10. Troubleshooting

Troubleshooting: Installation

Most installation issues will include an error message, either printed to your terminal, or written to `<command>.log`. If you do not see an explanation on the screen, check the log.

For any errors relating to start up of a component, you will likely find more information in the component's log. Use the `logs` command to view it.

Error 1

Link Client logs show `ENOSPC: System limit for number of file watchers reached.`

Solution 1

Increase the `inotify.max_user_watches` property.

```
echo fs.inotify.max_user_watches=524288 | sudo tee -a /etc/
sysctl.conf

sudo sysctl -p
```

Error 2

Link shows the following error message while installing Link on Redhat and CentOS:

```
sed: error while loading shared libraries: libsnl.so.1: cannot o
pen hared object file: No such file or directory
sed: error while loading shared libraries: libsnl.so.1: cannot o
pen hared object file: No such file or directory
sed: error while loading shared libraries: libsnl.so.1: cannot o
pen hared object file: No such file or directory
```

Solution 2

To install Link on Redhat and CentOS, complete the following steps:

1. Run the command `sudo yum install libnsl`.
2. Run the command `./Link install`.

Licensing

You can check to see if your Runtime REST API is licensing by executing this request through Swagger.

1. In your browser, navigate to this location: `https://<RUNTIME_REST_API_IP>:<PORT>/hip-rest/api-docs?url=openapi.json#/V2_Map_and_Flow_Deployment_API/listPackages`
2. Click on the 'unlocked /lock icon in the upper right corner of the page to specify a username and password for your request. Use the credentials that work for Link. Click **Authorize** then Close. The lock icon will show locked.
3. Click on **Try it out**.
4. Click on Execute.
5. Check the Response Code:
 - **200**=Valid license
 - **401 = with Response Body** "This Link instance is not licensed to execute the REST requests."

Invalid license. Check `<Link_Install>/tomcat-context/apps/platform.properties`.
Check entitlements through the Platform.
 - **401 with empty Response Body**.- Invalid credentials. Repeat step

Runtime REST API server missing in Link

The Runtime REST API server is configured during connector installation only if the **HIP_REST_SERVER_ADDRESS** environment variable exists in **hip-server-native.env** (native installation).

1. Open Link in your browser. The default configuration will be available with `https://<LINK_IP>:443`.
2. Open the Servers page from the **Deploy** menu.
3. Click on the plus icon in the upper right corner of the page to add a new server.

4. Configure the server details:

- Name: "Link Runtime"
- Type: "Web"
- Base URL: `https://localhost:8443/hip-rest` (native) or `https://hip-rest:8443/hip-rest` (docker).

5. Test the connection by clicking the Test button. Go to the UNABLE TO PING RUNTIME REST API FROM LINK troubleshooting section if the connection fails.

Chapter 11. Known issues

Summary of known issues in this version of the product.

Deleting functionality for Journey and Campaign applications

If a Unica application is removed, then it is necessary to run a script in Unica Link to clean up the artifacts that were created for that application.

This is done by running the script **remove_application.sh** which can be found in directory `<install_dir>/remove_application.sh` in the Unica Link install.

Run the command with the following 4 arguments:

remove_application.sh <base-url> <username> <password> <application>

where:

- **base-url** - the base URL for Link server. This should be of form: `https://<hostname>:8443`. If defaults were used during installation, then the default port is 8443.
- **username** and **password** – the credentials for an administrative user for Unica Link.
- **application** – the name of the application.

When successful, the command provides details about the deleted project and packages.

For example:

```
$ remove_application.sh https://localhost:8443 admin ***** journey
{
  "application": "journey",
  "deleted_project": {
    "_id": "5ed6b1de2ab79c0001a7e36d",
    "name": "_app_journey",
    "version": 1
  },
  "deleted_packages": [
```

```
    "_package__app_journey_act1" ,  
    "_package__app_journey_act23"  
  ]  
}
```