Unica Link V12.1 Installation Guide

Contents

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.

Upgrade from Unica Link 12.1.0.3 to 12.1.0.4

To upgrade Link, the user must uninstall the previous version of Link using the following command:

- **Native**: ./native-install.sh uninstall
- **Docker**: ./link_docker_install.sh uninstall
- **Windows**:

Instructions for uninstalling the Design Server

To uninstall the Design Server, complete the following steps:

1. Open the command line interface.
2. Change the current folder to `C:\HCL\Link_<version_number>\DesignServer`.
3. To stop and uninstall the Design Server, run the following command `clean.bat`.
4. Manually delete the folder `DesignServer` from the location `C:\HCL\Link_<version number>\`.
5. Uninstall the Windows Link installation completely (Link Design Server comes with Link runtime installation, uninstall runtime component also) by performing the following steps:
   - Click Windows Start button.
   - In the **Search box**, type `Add or Remove Programs`.
   - 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 17) and Installation of Unica Link in Docker Mode (on page 20).

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 environment:

- 4 CPUs,
- 8 GB RAM
- local disk of 40 to 60 GB

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_docker_install.sh script**.

Native installation prerequisites

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

- Linux Red Hat 7 - REHL 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 Design Server on Microsoft Windows, install Mongo DB Server version 4.0.5 (or higher) on the same computer. By default, Design Server and MongoDB server is expected to run on the same computer.

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

Design Server installation prerequisites

To use the Unica Link Design Server, 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 Design Server 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 Design Server 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>/tx-rest-native.properties`
- For docker installation: `<Link_install>/tx-rest-docker.properties`
- For Windows installation: `C:\HCL\Link_<version number>\tx-rest.properties`

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

authentication.enabled=true

Chapter 3. Microsoft Windows installation

This section describes how to install the Design Server 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. Install Link Design Server:
    a. Change the current folder to `C:\HCL\Link_<version number>\DesignServer`.
    b. Extract the compressed Design Server installation by executing the script `unzipDesignServer.bat`.
    c. Type `install.bat` and press `Enter`.
    d. Follow the messages on the screen. In the case of an installation error, check the `install.log` file. The log file displays error messages and possible actions that you can perform to fix the errors.

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

The Unica Link Design Server requires the Link Runtime server to be installed.

If you want to 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 the Unica Link Design Server

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

1. In the Command window, change the current folder to `C:\HCL\Link_<version-number>\DesignServer`.
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 the Design Server, run the stop.bat command. The server should stop in a few seconds.

Accessing the Unica Link Design Server

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

Use the default credentials (`admin/admin`) to login. 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 Design Server by editing settings in the file `C:\HCL\Link_<version-number>\DesignServer\hip-server-env.bat`.

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 the Design Server by running `clean.bat` followed by `install.bat`.

Customizing the Redis and Design Server Database

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

Redis customization instructions are available in the batch execution script (`hip-server-env.bat`) located in the DesignServer installation folder (`C:\HCL\Link_<version-number>\DesignServer\`). Modify the REDIS host and port settings to match your environment.

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

Customizing Design Server URL and Selecting HTTPS protocol

The Design Server default URL: `https://localhost/login` uses HTTPS protocol on port 443.

If you want to use the non-default port, and HTTPS protocol, you must update these settings:

- **HIP_HTTP**
- **HIP_HTTPS_PORT**
- **HIP_HTT_PORT**

For detailed instruction see the `hip-server-env.bat` 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\<Link Version>\` to the location mentioned for the `HIP_MODULES_DIR` parameter in the `%DTXHOME%\DesignServer\hip-server-env.bat` script.
2. Restart the Design Server.

Setting up Campaign

Steps to set up Campaign.

1. Edit the **campaign.properties** file. For more information, see Setting up Campaign *(on page 22)*.

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 22)*.

Uninstalling the Design Server

To uninstall the Design Server, 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 the Design Server.
3. Manually delete the `C:\HCL\Link_<version number>\DesignServer` folder
4. 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**.

Troubleshooting the installation

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

Folder type	**File / Folder Path**	**Description**

Main Folder	`install<TimeDateStamp>.log`
	`start<TimeDateStamp>.log`
	`stop<TimeDateStamp>.log`
		Contains information about the issues and steps required for resolving issues

Additional Folders	`C:\HCL\Link_<version-number>\restapi\tomcat\server\logs`	Design Server, Tomcat Application server files

Folder type	File / Folder Path	Description
	`c:\HCL\Link_<version-number>\Design-Server\client\daemon\*.log`	Design Server, WEB Node-JS Application server files
	`C:\HCL\Link_<version-number>\restapi\redis\logs`	Design Server, Redis server files

Chapter 4. Linux Installation

You can install Unica Link in two modes:

- Native Installation *(on page 17)*
- Docker Installation *(on page 20)*

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 17) and Installation of Unica Link in Docker Mode (on page 20).

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.
- `--design-server` - Design 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.
- `--design-server` - Design 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.
- `--design-server` - Design 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.
- `--design-server` - Design 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.
- `--design-server-app` - Design Server - Backend.
- `--design-server-client` - Design 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.
- `--design-server` - Design 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

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 tempaorary 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. To specify the new directory, add a new line to `hip-server-native.env` For example:

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

4. For the long and short workers, modify the existing lines in `hip-server-native.env`. For example:

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


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

5. To copy the new configuration into the Link installation, execute the following command:

   ```
   cp hip-server-native.env tomcat-context/install/hip-server.env
   ```

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

You must configure Unica Link before running any installation commands. If you choose not to, you may have to reinstall for the updated configurations to take effect.

There are two files to configure before installation (use ‘native’ post-fixed versions):

- `hip-server-native.env`
- `tx-rest-native.properties`

To reconfigure after installation, update those files, and run the install command. This will overwrite the configuration files in the install directory.

You will observe several directories specified in those configuration files. If the user or group you install as, does not have read/write access to these directories, you will need a system administrator to create them and transfer ownership.

If you are installing as `non-root` (recommended), you must change the `HIP_HTTPS_PORT` specified in `hip-server-native.env` to a number above 1024. The standard Linux kernel prevents regular users from starting processes that bind to any port below 1024. Alternatively, you can use setcap to overcome this limitation:

Note: Install Unica Link before using setcap

```
sudo setcap ‘cap_net_bind_service=+ep’
 <link_package>/node-context/install/nodejs/bin/npm
sudo setcap ‘cap_net_bind_service=+ep’
 <link_package>/node-context/install/nodejs/bin/node
```

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

d.
```
./Link install
```

e. Change the kafkalink properties under `<Link-install>/integration-context/kafka-link/`.

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

```
./Link start
```

2. After Link is up and running, run any flow from swagger to generate `hch-config.properties` under `location /opt/link/restapi/config/`. Provide existing location for the parameter `results_dir`.

Installation of Unica Link in Docker Mode

Dependencies

- Docker

Configuration

You should configure Unica Link before running any installation commands. If you choose not to, you might need to reinstall for your updated configuration to take effect.

There are two files to configure before installation (DO NOT use ‘native’ post-fixed versions):

- `hip-server.env`
- `tx-rest-docker.properties`

To reconfigure properties from `hip-server.env` after installation, run the `update_envs.sh` script after making your changes.

To reconfigure properties from `tx-rest-docker.properties`, you have to copy the file into the `hip-rest` container and then restart it:

```
docker cp ./tx-rest-docker.properties
 hip-rest:/opt/runtime/tx-rest.properties
```

```
./Link stop --runtime
```

```
./Link start --runtime
```

You will observe several directories specified in those configuration files. If the user or group you install as, does not have read-write access to these directories, you will need a system administrator to create them and transfer ownership.

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

    d.
    ```
    ./Link install
    ```

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

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

    ```
    ./Link start
    ```

2. After Link is up and running, run any flow from swagger to generate `hch-config.properties` under location `/opt/link/restapi/config/`. Provide existing location for the parameter `results_dir`.

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:8080/journey/api/login` data_definition_url=`http:// IPADDRESS:8080/journey/api/datadefinitions/point`
   b. Specify Journey credentials:username=<*username*>password=<*password*>
3. Set up the **results_dir** attribute with an existing folder structure (The `hch-config.properties` files are within the `config` folder in HCH installer), which is utilized for intermediate file creation.
   - For Linux docker, use `/data/<any-folder>`.
   - For native installation, use `/root/hiprest/<any-folder>`.
   - For Microsoft Windows, use `C:\...\...\<any-folder>`.

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:7001/unica/api/manager/authentication/login` data_definition_url=`http://IPADDRESS:7002/Campaign/api/campaign/rest/v3/link/field-information?actionId`

b. Specify campaign headers:m_user_name=<*username*>m_user_password=*password*api_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*>**/link-context/kafka-link**
2. Update the following:
   - List of one or more Kafka brokers= **IPADDRESS:9092**
   - The base URL for the HIP runtimehip-url=`https:// IPADDRESS:8443/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*>**/link-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:1200`
   - Credentials:username=**asm_admin**password=**password**

Chapter 6. Logging

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

Design Server logs

These are the logs produced by Design Server:

- For Native Installer:
  - **native-install.sh logs -runtime**
  - **native-install.sh logs -client**
- For Docker Installer:
  - **docker logs hip-server**
  - **docker logs hip-server**

Runtime REST API logs

These logs are produced by the Runtime REST API:

Native Install:

- **native-install.sh logs -runtime**

Docker Install:

- **docker logs hip-rest**

Kafkalink logs

These logs are produced by Kafkalink:

Native Install:

- **native-install.sh logs -kl**

Docker Install:

- **docker logs kafka-link**

Making logs verbose

To make logs more verbose you can modify the **tx-rest.properties** file. Modify the file a follows:

1. For Native or Docker:
    - Native Install: Go to: <install_dir>/**tx-rest.properties**. By default, <install_dir> is in <installation_tar>**/tomcat-context/install**
    - Docker install: Edit **tx-rest.properties** in the **hip-rest** Docker container by using the command:

      docker exec -it hip-rest vi /opt/runtime/tx-rest.properties
2. Change the value of **trace.level**.
3. For Native or Docker:
    - Native Install: **native-install.sh restart -runtime**
    - Docker Install: **docker restart hip-rest**

Chapter 7. Modifying the configuration in Docker installation or Native installation

After you install Unica Link you might want to change the HTTP / HTTPS configuration.

To change the HTTP/HTTPS configuration in the Docker installation or the Native installation, complete the following steps:

1. Open the `ENV` file.
   a. For Docker installation, open the **hip-server.env** file.
   b. For Native installation, open the **hip-server-native.env** file.
2. Edit the HIP Design Client HTTP/HTTPS configuration to match your requirements as described in the `ENV` file.
   a. For Docker installation, edit the configurations in the **hip-server.env** file.
   b. For Native installation, edit the configurations in the **hip-server-native.env** file.
3. Save the `ENV` file.
   a. For Docker installation, save the **hip-server.env** file.
   b. For Native installation, save the **hip-server-native.env** file.
4. Perform one of the following steps depending on your installation:
   a. For Docker installation, run the script update_client_http.sh.
   b. For Native installation, restart the client by running the command native-install.sh restart -client.

Chapter 8. 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

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

Solution

Increase the inotify max_user_watches property.

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

```
sudo sysctl -p
```

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 Design Server. 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 Design Server

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 the Link Design Server 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: “HCH 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 DESIGN SERVER troubleshooting section if the connection fails.

Chapter 9. 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"
    ]
}
```