Unica Deliver V12.1 Startup and Administrator's Guide
Chapter 1. Hosted messaging using Unica Campaign and Unica Deliver

Establishing a hosted email account with Unica

Overall view of the startup process

Before you begin working with Unica Deliver

Chapter 2. Configuring the local HCL Unica environment for Deliver

Confirming Deliver registration

Registering Deliver manually

Enable Deliver functions in Campaign

Displaying Deliver menu options

Specifying Deliver system table characteristics

Configuring access to local Deliver system tables

Required mapping for Deliver system tables in Campaign

Required restart of the web application server for Campaign

Chapter 3. Connections to message services

Requirements for configuring connection to HCL Unica hosted services

Requirements for uploading data to HCL Unica hosted services

Connection and port requirements

Whitelisting IP

Default upload connection by explicit FTP

Data upload with implicit FTP

Connection through an HTTP proxy

Data download frequency and port setting

Configuring a system user to access HCL Unica hosted services
Configuring the system user that accesses HCL Unica hosted services.............. 30
To configure addresses for connecting to HCL Unica Hosted Services.................. 32
IP address of Deliver host names........................................................................ 33
Configuring secure communication for hosted email............................................. 33
Generating a trusted keystore.................................................................................. 34
Configuring SSL when using WebLogic................................................................. 36
Configuring SSL when using WebSphere............................................................... 39

Chapter 4. Response and Contact Tracker operation.............................................. 42
Manual operation of the Response and Contact Tracker....................................... 42
Adding the Response and Contact Tracker as a service....................................... 43
Removing the Response and Contact Tracker service......................................... 44

Chapter 5. Startup verification.................................................................................... 45
Confirmation for system configurations................................................................. 45
Testing upload to HCL Unica hosted services...................................................... 48
Testing download from HCL Unica hosted services............................................ 49
Testing the connection to the hosted messaging interface.................................... 49

Chapter 6. About configuring Unica Deliver.......................................................... 50
Configuring access to additional mailing execution history................................. 52
Support for Campaign offer integration................................................................. 53
Configuring support for dimension tables............................................................ 53
To configure access to local Deliver system tables.............................................. 54
Configuring Journey - Deliver integration............................................................ 55
Configuration properties for Deliver...................................................................... 56
Campaign | partitions | partition[n] | Deliver.......................................................... 56
Campaign | partitions | partition[n] | server | internal...................................................... 59
Chapter 7. Configurations for implementing Push notifications

Configure your Apple developer account

Configure Firebase Cloud Messaging

Configure your Unica app

Integrate the SDK

iOS Swift

Introduction

Integration

Configure your app capabilities and entitlements

Registering your CRM

Advanced features

Android

Configure your Apple developer account

Integration

Registering with your CRM

Advanced features

Troubleshooting

React Native
Messaging permissions for Deliver ............................................................................. 144
Assigning Deliver roles ............................................................................................... 144
Controlling email domains and short link domains ...................................................... 145
Maintenance of hosted email domains ....................................................................... 146
Configuring default sender address and display names ................................................. 147
Controlling access to the list of sent messages ............................................................. 149
Granting access to the list of sent messages ................................................................. 150
Denying access to the list of sent messages ................................................................. 150
Enabling the restriction to the sent message list ......................................................... 152
About permissions for Deliver reports ...................................................................... 153

Chapter 11. Technote (troubleshooting) ...................................................................... 154
Connecting to message services through a proxy ....................................................... 155
Changes required for routing FTPS and HTTPS traffic through a SOCKS proxy ... 156

Chapter 12. Deliver channel vendor account configuration ...................................... 163
Karix SMS account configuration ............................................................................... 163
RML SMS account configuration ............................................................................... 165
RML WhatsApp account configuration ....................................................................... 166
Twilio SMS account configuration ............................................................................... 167

Index ............................................................................................................................. a
Chapter 1. Hosted messaging using Unica Campaign and Unica Deliver

When Unica Campaign is integrated with Unica Deliver, you can use Deliver to conduct highly personalized digital marketing campaigns.

Note: Deliver supports the following channels along with email. In this guide, the term message applies to all channels.

• SMS
• WhatsApp
• Push

Deliver provides access to resources hosted by Unica so you can design, send, and monitor individually customized messages based on information stored in your customer datamart.

• In Campaign, use flowcharts to create lists of message recipients and select personalization data for each recipient.
• In Deliver, use message design, transmission, and deliverability resources hosted by HCL to conduct digital marketing campaigns.

Establishing a hosted email account with Unica

When you purchase a message subscription, Unica creates a hosted email account on your behalf and sends you the account credentials that you will need to use message features. You apply these credentials when you configure your local HCL Unica applications to access the hosted email environment over secure connections.

You must have a valid account to access the message resources that Unica provides as a software service. If your HCL Unica installation includes multiple partitions and you plan to use message in more than one partition, you need a hosted email account and at least one service provider for WhatsApp and SMS for each partition. You cannot share email accounts across installations or partitions.
Establishing a hosted email account is the beginning of the startup process, which lasts for approximately 90 days. You can subscribe to SMS or WhatsApp as per the requirements. For a general description of the process, see the next topic.

Overall view of the startup process

You can activate message features in Unica Campaign to conduct highly targeted and trackable digital marketing campaigns. Campaign uses message functions that are provided by Unica Deliver through resources that are hosted in data centers in the US and Europe. An account to access the email resources is included with your Deliver subscription. You can also opt for the WhatsApp, Push, or SMS channels as per the requirements.

Unica begins the startup process after it creates your hosted email account. Unica helps you to become familiar with Deliver, connect to message resources, and establish your reputation as a legitimate digital marketer among leading Internet Service Providers (ISPs).

The process proceeds in three phases. The Unica Professional Services and Deliver Services teams guide you along the way.

The Professional Services consultant is your primary point of contact with Unica during the startup process. When the account startup process completes, the Professional Services consultant transfers primary support responsibility to the Unica Product Support team.
A dedicated Deliver Services consultant provides special assistance for message-related issues. Creating a favorable message reputation among major Internet Service Providers (ISPs) is critical to ensuring that your digital marketing campaigns consistently reach their target recipients. When you start to run mailings, the EAS consultant reviews the mailing deliverability performance and suggests the best ways to gradually build your message reputation.

Startup activities and milestones

**Planning**

**List of planning activities**

<table>
<thead>
<tr>
<th>What happens</th>
<th>Who is responsible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send email account credentials and the Welcome Kit, including the Email Startup Worksheet.</td>
<td>Unica Deliver Services</td>
</tr>
<tr>
<td>Schedule a conference call to introduce all involved parties, review the startup schedule, and understand the email marketing objectives.</td>
<td>Unica Professional Services</td>
</tr>
<tr>
<td>Complete the Email Startup Worksheet to specify your email domain requirements and mailing projections.</td>
<td>Your organization</td>
</tr>
</tbody>
</table>

**Create your email reputation**

**Actions required to establish a favorable email reputation**

<table>
<thead>
<tr>
<th>What happens</th>
<th>Who is responsible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provision the email account using information provided during the conference call and in the Email Startup worksheet.</td>
<td>Unica email operations</td>
</tr>
<tr>
<td>Begin warm up mailings to selected test accounts with major ISPs. This phase requires approximately 30 days to complete.</td>
<td>Unica email operations</td>
</tr>
</tbody>
</table>
### What happens | Who is responsible

<table>
<thead>
<tr>
<th>What happens</th>
<th>Who is responsible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate Deliver in Unica Campaign.</td>
<td>Your organization (with support from Unica)</td>
</tr>
<tr>
<td>Configure access to hosted email resources. Consult with the EAS consultant about which data center to specify.</td>
<td>Your organization (with support from Unica)</td>
</tr>
<tr>
<td>Start sending mailings. To build a favorable email reputation, send small mailings initially, followed over time by larger and more frequent mailings. ISPs often try to limit spam by blocking large or frequent mailings from email domains they do not recognize as legitimate.</td>
<td>Your organization (with support from Unica)</td>
</tr>
<tr>
<td>Provide deliverability results and reputation guidance as mailing volumes and frequency gradually increase.</td>
<td>Unica Deliver Services</td>
</tr>
</tbody>
</table>

### Production

**Activities required to reach and maintain desired production volumes**

<table>
<thead>
<tr>
<th>What happens</th>
<th>Who is responsible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send mailings at typical volume and frequency.</td>
<td>Your organization</td>
</tr>
<tr>
<td>Transfer primary contact responsibility to the Unica Support team.</td>
<td>Unica Professional Services</td>
</tr>
<tr>
<td>Maintain engagement for consultation on email issues. Make contact on a regular basis for continuing email account support.</td>
<td>Unica Deliver Services</td>
</tr>
</tbody>
</table>

---

**Before you begin working with Deliver**

Before you begin the messaging startup process, consider the following issues.
• Some configurations require restarting the web application server. Plan Deliver configuration activity to avoid interfering with large flowchart runs and other activities in Campaign.

• Unica asks you to name one individual to serve as the primary contact point during the startup process.

• Request hosted email account credentials before you begin the startup process. You use these credentials to configure your systems to access the account.

• Consult with your network administration staff. Deliver requires specific port ranges when communicating with HCL Unica.

• Confirm that you have the appropriate network permissions to make configuration changes.
Chapter 2. Configuring the local HCL Unica environment for Deliver

Using Deliver to send messages requires changes in the local installation of HCL Unica. Complete the steps described in the following sections.

- Enable Deliver functions in Campaign (on page 7)
- Registering Deliver manually (on page 7)
- Specifying Deliver system table characteristics (on page 9)
- Required mapping for Deliver system tables in Campaign (on page 11)
- Configuring access to local Deliver system tables (on page 10)
- Required restart of the web application server for Campaign (on page 12)

If your environment contains multiple partitions, repeat these steps for each Campaign partition in which you use Unica Deliver. For more information about creating and working with multiple partitions, see the Unica Campaign Installation Guide.

Confirming Deliver registration

Unica Deliver must be registered with Unica Platform. To confirm that Deliver is registered successfully, you must examine the configuration for Platform.

1. Log in to HCL Unica.

2. Navigate to Settings > Configuration.

3. Look for the Deliver configuration category.

   Unica Deliver is registered with Platform when the Deliver category appears in the configuration properties hierarchy.

If the Deliver category does not appear in the properties hierarchy, see the Unica Campaign Installation Guide for information about how to register Deliver manually.
If the Deliver category is available, you must enable Deliver functions in Campaign.

Registering Deliver manually

During the installation process if the Deliver installer cannot access the Platform system tables, you must run the configTool utility to register it manually.

By default, the Campaign installer automatically registers Deliver with the Platform system tables without enabling Deliver. In some situations, the Campaign installer does not connect with the Platform system tables to automatically register Deliver.

If the installer does not register Deliver automatically, you must register Deliver manually with the configTool utility that is provided with the HCL Unica installation. The configTool utility is in the tools\bin directory under your Platform installation.

To register Deliver manually, use the following command to run the configTool utility:

```
configTool -r Deliver -f "full_path_to_Deliver_installation_directory\conf \Deliver_configuration.xml"
```

The Deliver installation directory is a subdirectory of the Campaign installation directory.

Enable Deliver functions in Campaign

When you install Campaign, the installer also installs Deliver in the default partition, but does not enable it. Deliver functions are not available until you enable Deliver.

You enable Deliver with the following configuration property, in the Unica Platform.

```
Campaign > partitions > partition[n] > server > internal > deliverInstalled
```

To enable Deliver, change the value to yes.

Registration requirement

Registering Deliver with Unica Platform is required to operate Deliver. You register Deliver with Platform when you install Unica Campaign.
After you have enabled Deliver, confirm that Deliver is properly registered with Unica Platform. For details, see Confirming Deliver registration (on page 6).

Displaying Deliver menu options

To use Unica Deliver, you must update the system configuration so that menu options for Deliver display in the Unica Platform interface. When you install Campaign, the installer also installs Deliver menus in the default partition. In case Campaign installer do not connect with the Platform system tables, you must configure them manually using the following steps. To display the required options, use the configTool utility that is supplied with your HCL Unica installation.

You must run configTool with specific parameters for each Deliver menu option. Running configTool updates system configuration settings. You must restart the web application server to apply the changes. Although Deliver is installed with Campaign, menu options for Deliver do not appear until after you run configTool and restart the web application server. In the tools directory of the Platform installation, the configTool utility is located in the bin folder.

⚠️ Note: You must specify a path to the Deliver installation directory as a configTool parameter. The Deliver installation directory is a subdirectory of the Campaign installation directory.

- To display Deliver Settings in the Settings menu.

  configTool.bat -v -i -p "Affinium|suite|uiNavigation|settingsMenu" -f "full_path_to_Deliver_installation_directory\conf\deliver_op_odsettings_navigation.xml"

- To display Deliver Mailings in the Campaign menu.

  configTool.bat -v -i -p "Affinium|suite|uiNavigation|mainMenu|Campaign" -f "full_path_to_Deliver_installation_directory\conf\deliver_op_mailings_navigation.xml"

- To display Quick Builder in the Campaign menu.
unica deliver v12.1 startup and administrator's guide

2 - configuring the local hcl unica environment for deliver

configtool.bat -v -i -p "affinium|suite|uinavigation|mainmenu|campaign" -f "full_path_to_deliver_installation_directory\conf\deliver_op_new_documents_navigation.xml"

• To display deliver documents in the campaign menu.

configtool.bat -v -i -p "affinium|suite|uinavigation|mainmenu|campaign" -f "full_path_to_deliver_installation_directory\conf\deliver_op_documents_navigation.xml"

• To display deliver analytics in the analytics menu.

configtool.bat -v -i -p "affinium|suite|uinavigation|mainmenu|analytics" -f "full_path_to_deliver_installation_directory\conf\deliver_op_analytics_navigation.xml"

To verify that you successfully added the menu options, after you restart the web application server, log in to hcl unica and open the settings, campaign, and analytics menus to verify that the deliver options appear.

specifying deliver system table characteristics

unica deliver requires information that describes the type, schema, and jdbc connection for the deliver system tables in your installation. the deliver system tables are created in the campaign schema as part of the campaign installation process.

• Navigate to settings > configuration > deliver > partitions > partition[n] > datasources > system tables.

• Review and update information for the following parameters.

  note: the campaign system table information is expected here.

  database type

  schema name

  jdbcbatchsize
jdbcClassName
jdbcURI
asmDataSourceForDBCredentials - must be UA_SYSTEM_TABLES

• Provide the required information in the following configuration properties. See the Platform online help for each property to learn more about setting the configuration properties.

  ◦ Deliver > partitions > partition [n] < dataSources > systemTables > type
  ◦ Deliver > partitions > partition [n] < dataSources > systemTables > schemaName
  ◦ Deliver > partitions > partition [n] < dataSources > systemTables > jdbcBatchSize
  ◦ Deliver > partitions > partition [n] < dataSources > systemTables > jdbcClassName
  ◦ Deliver > partitions > partition [n] < dataSources > systemTables > jdbcURI

For additional information about configuration properties and configuring Deliver, see Configurations for Unica Deliver (on page 50).

Configuring access to local Deliver system tables

Unica Deliver requires access to the Deliver system tables in the Campaign schema. To allow Unica Deliver components to access system tables in the Campaign schema without requesting a manual database login, you must specify an Deliver system user to provide the necessary database access credentials.

The system user that accesses the database is associated with an Unica Platform data source that contains the login credentials for the database that hosts the Campaign schema.
For more information about system table configuration properties, see Deliver | partitions | partition[n] | dataSources | systemTables (on page 67).

1. Specify the system user that you defined in the Unica Platform. Edit the following configuration property.

   Deliver > partitions > partition [n] < dataSources > systemTables > asmUserForDBCredentials

2. Specify the login credentials for the database that contains the Campaign schema and Deliver system tables. Edit the following configuration property.

   Deliver > partitions > partition [n] < dataSources > systemTables > amDataSourceForDBCredentials

Required mapping for Deliver system tables in Campaign

You must map Deliver system tables in the Campaign schema to corresponding Deliver database tables. The Deliver system tables have Deliver in the table name.

In Campaign, map the following Deliver system tables.

- Deliver Output List Table
- Deliver Output List Audience Fields Mapping Table
- Deliver Mailing Table
- Deliver Mailing Instance Table
- Deliver Data Table Column Mapping Table
- Deliver Personalization Field Mapping Table
- Deliver Personalization Field Usage Table

For information about mapping tables, see the Unica Campaign Administrator's Guide.
Required restart of the web application server for Campaign

After making changes to the Campaign and Deliver configurations, you must restart the web application server that hosts Campaign.

Consult the documentation for your web application server for restart instructions.
Chapter 3. Connections to message services

To access message services provided by Unica, you must configure a connection between the local HCL Unica installation and HCL Unica.

Marketers access Deliver features through the Campaign interface. Working with Deliver requires that you establish a secure, automatic Internet connection that Campaign can use to upload message recipient lists to HCL Unica. Deliver components installed with Campaign also use this connection to download contact and response data to the Deliver system tables in the Campaign schema.

⚠️ Note: Each instance of Campaign requires a unique connection to HCL Unica. If the Campaign installation includes multiple partitions, each partition requires a separate hosted email account. The accounts can share the IP connection to HCL Unica.

All communication between HCL Unica and HCL Unica hosted services is done over SSL. Each communication from HCL Unica hosted services is a response to a request from the local environment. HCL Unica hosted services never attempts to initiate a connection with your corporate network. All communication with HCL Unica hosted services originates from behind your corporate firewall.

Requirements for configuring connection to HCL Unica hosted services

Configuring a connection to HCL Unica hosted services requires administrative permissions and information about the hosted email account established for your organization.

To configure a hosted email connection, you need the following.

- User name and password provided by Unica for the hosted email account
- Permissions to create or modify system users in the Unica Platform
- Administrative access to configuration properties maintained in the local Unica Platform installation
• Administrative access to the web application server on which the Unica Platform and Campaign are deployed

You must know, or be able to consult with people who know, your corporate data security requirements. Before you begin, review these procedures to understand how to create the necessary connection in compliance with your corporate firewall restrictions.

You must be familiar with how to configure trusted connections on your web application server, IBM WebSphere®, Oracle WebLogic, Apache Tomcat, and JBOSS.

Requirements for uploading data to HCL Unica hosted services

A Deliver component called the Recipient List Uploader (RLU) is part of your Unica Campaign installation. The RLU uses FTP in passive mode to manage the upload of email mailing lists and associated metadata to HCL Unica hosted services.

Deliver uses passive FTP to upload data. When using passive FTP, the RLU initiates all upload connection requests as the local client. HCL Unica hosted services never initiates a connection request to your network.

Deliver supports two passive FTP methods, explicit FTP and implicit FTP. Explicit FTP is the default method used for uploading recipient lists. To use implicit FTP you must make changes in the Deliver configuration properties.

For more information about FTP in passive mode and using FTP over SSL, see RFC959 and RFC2228.

Connection and port requirements

To communicate with HCL Unica hosted services, you must have a connection to the Internet. HCL Unica hosted services use specific ports.

The local HCL Unica installation and HCL Unica hosted services use the following ports to communicate.

**HTTPS**: port 443
FTP command port:

- Explicit FTP: port 21
- Implicit FTP: port 990

FTP data upload ports

- U.S. data center
  - Explicit FTP: ports 15393 to 15443
  - Implicit FTP: ports 15600 to 15650
- Europe data center
  - Explicit FTP: ports 15393 to 15443
  - Implicit FTP: ports 15600 to 15650

HCL Unica hosted services never initiates a connection with your local network. It only responds to connection requests initiated from behind your firewall.

Whitelisting IP

To upload the recipient list (OLT) to FTP Deliver server, the external IP of the server on which Campaign Web is running must be whitelisted at Deliver On Demand server side.

You must get the external IP using the following commands and provide it to the Onboarding team. The Onboarding team will request to whitelist the IP so that FTP requests from your server to FTP Deliver server is allowed.

- On Unix system, run `curl ifconfig.me` command to get the external IP of your server.
- On Windows system, you can access `http://ifconfig.me` to get the external IP of your server.
Default upload connection by explicit FTP

By default, the Recipient List Uploader (RLU) uses explicit FTP when the system uploads an email recipient list. The recipient list is also referred to as the Output List Table (OLT)

The RLU establishes a connection with HCL Unica hosted services through the default FTP command port (port 21). It issues a request to encrypt the session over SSL. Over the secure connection, the RLU negotiates with the FTP server to establish a separate SSL link over a port that is randomly selected by the RLU. For more information about explicit FTP, see RFC 2228.

The following diagram illustrates the default method for uploading recipient data from Campaign to HCL Unica hosted services.

The following table describes the connection sequence.
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial client FTP connection request</td>
<td>From behind the corporate firewall, the RLU initiates a data upload session by using FTP over explicit SSL. The RLU sends the SSL connection request to the address for HCL Unica hosted services. You must configure this address in advance. To start the session, the RLU opens a randomly selected port on the client side as its FTP command port. HCL Unica hosted services accept FTP command connections on port 21.</td>
</tr>
<tr>
<td>2</td>
<td>Remote FTP server response</td>
<td>In response to the RLU request a secure SSL session, the FTP server designates the FTP data port to use for the recipient list upload.</td>
</tr>
<tr>
<td>3</td>
<td>Recipient List upload</td>
<td>The RLU commences the list upload on the specified data port. When the upload finishes, the RLU drops the FTP connection.</td>
</tr>
</tbody>
</table>

For the range of data ports the FTP server can specify, see [Connection and port requirements](#) (on page 14).

Configuring the explicit FTP connection

No additional configuration is required. The RLU uses explicit FTP by default.

Data upload with implicit FTP

You can configure the RLU to use implicit FTP to upload email recipient lists. Implicit FTP uses port 990 to establish a connection to HCL Unica hosted services.

(For explanation of steps and configuration details)
When you enable the RLU for implicit FTP, the entire session is encrypted from the start. Implicit FTP is unlike connection through explicit FTP, where the RLU explicitly requests a secure link.

To use implicit FTP to upload recipient lists, you must reveal and configure a system configuration property in the Deliver configuration properties.

The following diagram illustrates how the RLU uploads recipient data to HCL Unica hosted services when the system is configured to use implicit FTP.

![Diagram showing the connection sequence between Local Network, Recipient list uploader (RLU), FTP Server, and HCL Unica hosted services.]

The following table describes the connection sequence.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial client FTP connection request</td>
<td>From behind the corporate firewall, the RLU initiates a data upload session by using FTP over SSL. The RLU sends the SSL connection request to the address for HCL</td>
</tr>
<tr>
<td>Step</td>
<td>Action</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>Unica hosted services. You must configure this address in advance. To start the session, the RLU opens port 990. HCL Unica hosted services accept encrypted FTP command connections only on port 990.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Remote FTP server response</td>
<td>If HCL Unica hosted services recognize the request as a valid implicit FTP request, the FTP server accepts the connection request. It designates the FTP data port to use for the recipient list upload. See <strong>Connection and port requirements (on page 14)</strong> for the range of data ports the FTP server can specify.</td>
</tr>
<tr>
<td>3</td>
<td>Recipient List upload</td>
<td>The RLU commences the list upload on the specified data port. When the upload finishes, the RLU drops the FTP connection.</td>
</tr>
</tbody>
</table>

**Accessing the configuration parameters for implicit FTP upload**

To configure the system to use implicit FTP for uploading data, you must edit configuration properties that are hidden by default. You run a script to reveal the properties.

To configure the system to use implicit FTP for uploading data, make the following configuration properties visible. These properties are hidden by default.

- `Deliver > serverComponentsAndLocations > hostedServices > ftpPort`
- `Deliver > serverComponentsAndLocations > hostedServices > useFTPImplicitSSL`
This procedure reveals the configuration properties but does not configure them. To enable implicit FTP, you must access these properties in the Deliver configuration and configure them. For information about how to configure these properties, see Enabling implicit FTP upload (on page 20).

To reveal ftpPort and useFTPImplicitSSL, in the Tools directory of your Deliver installation, run the switch_config_visibility script from the command line script, as follows.

**Windows™**

```
\switch_config_visibility.bat -p "Affinium|Deliver|serverComponentsAndLocations|hostedServices|ftpPort" -v true
\switch_config_visibility.bat -p "Affinium|Deliver|serverComponentsAndLocations|hostedServices|useFTPImplicitSSL" -v true
```

**UNIX™**

```
/switch_config_visibility.sh -p "Affinium|Deliver|serverComponentsAndLocations|hostedServices|ftpPort" -v true
/switch_config_visibility.sh -p "Affinium|Deliver|serverComponentsAndLocations|hostedServices|useFTPImplicitSSL" -v true
```

You must restart the web application server to make these properties visible in the Deliver configuration.

**Enabling implicit FTP upload**

To enable implicit FTP, you must update the Deliver configuration.

To complete this task, you must reveal two configuration properties. For more information, see Accessing the configuration parameters for implicit FTP upload (on page 19).

1. **Navigate to** Settings > Configuration > Deliver > serverComponentsAndLocations > hostedServices.

2. **Click Edit Settings.**
   - **Confirm that useFTPImplicitSSL is set to true.**
• **Set `ftpPort` to 990.**

3. Save your changes.

The changes do not take effect until you restart the web application server and the Campaign listener. You can do this now or wait until you complete all of the startup configuration steps.

**Connection through an HTTP proxy**

If your corporate firewall rules prohibit direct communication with the public Internet, you can connect to HCL Unica through an HTTP proxy server. Deliver supports connecting through a SOCKS proxy server that allows both HTTPS and FTPS traffic.

Deliver supports SOCKS Protocol Version 5.

The following diagram illustrates communication between the local and hosted environments when you use a SOCKS proxy.
You configure the SOCKS proxy server in the local On Premises environment. Before you begin configuring the proxy server, confirm that you satisfy the following requirements.

- The proxy server must be a SOCKS proxy server.
- The proxy server must be able to access the Deliver OD environment. The server must allow traffic to and from the ports that are configured for the data center that is used by your hosted email account. Unica maintains data centers in the United States and in the Europe.
- The Deliver OP environment must be able to access the SOCKS proxy server.
Configuring routing for FTPS and HTTPS traffic through a SOCKS proxy

To use a SOCKS proxy to access the hosted email resources, you must update the web application server where you deployed Campaign. You must also modify the startup scripts for the Deliver RCT and RLU.

• For FTPS traffic, apply the following configurations to the RLU and the web application server.

**RLU and webservice configurations for FTPS.**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Dhcl.unica.deliver.ftps.proxy.host = &lt;socksHost&gt;</td>
<td>Host name or IP of the SOCKS proxy.</td>
</tr>
<tr>
<td>-Dhcl.unica.deliver.ftps.proxy.port = &lt;socksPort&gt;</td>
<td>The port on which SOCKS proxy is running.</td>
</tr>
<tr>
<td>-Dhcl.unica.deliver.ftps.proxy.match.hosts= &lt;comma separated list of host names and IP addresses&gt;</td>
<td>Host names and IP addresses that are used when routing traffic through the SOCKS proxy. Provide values specific to the data center used by your account.</td>
</tr>
</tbody>
</table>

When the local and hosted environments establish a data connection, the IP address that is specified for `-Dhcl.unica.deliver.ftps.proxy.match.hosts` is the IP address that the remote FTP server sends to the local FTP client.

Set `-Dhcl.unica.deliver.ftps.proxy.match.hosts` to one of the following values. The value that you enter depends on the data center that is used by your hosted email account.
Host name and IP addresses for the US data center:

-Dhcl.unica.deliver.ftps.proxy.match.hosts=
ftp-em.unicadeliver.com

Host name and IP addresses for the Europe data center:

-Dhcl.unica.deliver.ftps.proxy.match.hosts=
ftp-eu.unicadeliver.com

- For HTTPS traffic, apply the following configurations to the RCT and the web application server.

**Configuration settings for HTTPS to SOCKS proxy**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Dhcl.unica.deliver.https.proxy.host=&lt;socksHost&gt;</td>
<td>Host name or IP of the SOCKS proxy</td>
</tr>
<tr>
<td>-Dhcl.unica.deliver.https.proxy.port=&lt;socksPort&gt;</td>
<td>The port on which SOCKS proxy is running</td>
</tr>
<tr>
<td>-Dhcl.unica.deliver.https.proxy.type=SOCKS</td>
<td>The type of proxy server. You must use a SOCKS proxy server.</td>
</tr>
</tbody>
</table>

Configuring authentication to access a SOCKS proxy

If your SOCKS proxy requires authentication, you must configure the web application server, RLU, and RCT to provide the access credentials.

Configure the following for the web application server, RLU, and RCT. The values for **username** and **password** must be the credentials that are required to authenticate against the proxy.
Connections to message services

-Dhcl.unica.deliver.proxy.auth.user = <username>
-Dhcl.unica.deliver.proxy.auth.password = <password>

Configuring the RCT to use a SOCKS proxy

You must modify the RCT to communicate through a SOCKS proxy server. The required settings depend on your operating system.

- For the RCT in Windows environments, add the following proxy arguments to common.bat.

  The common.bat file is in the \deliver\bin directory of your local Deliver installation.

```batch
set RCT_PROXY_ARGS=
-Dhcl.unica.deliver.https.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.https.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.https.proxy.type=SOCKS
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUTH_USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUTH_PASSWORD>
set RCT_JAVA_ARGS=%BASE_VM_ARGS% %RCT_MEM_ARGS% %RCT_EXTRA_VM_ARGS% %RCT_PROXY_ARGS%
```

- For the RCT in UNIX environments, add the following proxy arguments to common.sh.

  The common.sh file is in the /deliver/bin directory of your local Deliver installation.

  Note: Do not directly modify rlu.sh, rct.sh, or setenv.sh. The system overrides the changes.

```sh
RCT_PROXY_ARGS=""
-Dhcl.unica.deliver.https.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.https.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.https.proxy.type=SOCKS
```
Configuring the RLU to use a SOCKS proxy

You must modify the RLU to communicate through a SOCKS proxy server. The required settings depend on your operating system.

- For the RLU in Windows environments, add the following proxy arguments to common.bat.

```bash
set RLU_PROXY_ARGS=
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUTH_USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUTH_PASSWORD>

RLU_JAVA_ARGS="${BASE_VM_ARGS} ${RCT_MEM_ARGS} ${RCT_EXTRA_VM_ARGS} ${RCT_PROXY_ARGS}"
```

- For the RLU in UNIX environments, add the following proxy arguments to common.sh.

```bash
set RLU_PROXY_ARGS=
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUTH_USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUTH_PASSWORD>

RLU_JAVA_ARGS="${BASE_VM_ARGS} ${RLU_MEM_ARGS} ${RLU_EXTRA_VM_ARGS} ${RLU_PROXY_ARGS}"
```

**Note:** Do not directly modify rlu.sh, rct.sh, or setenv.sh. The system overrides the changes.
-Dhcl.unica.deliver.ftps.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.ftps.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.ftps.proxy.match.hosts=<comma separated list of host names and IP addresses>
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUTH_USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUTH_PASSWORD>

RLU_JAVA_ARGS="${BASE_JAVA_ARGS} ${RLU_MEM_ARGS} ${RLU_EXTRA_VM_ARGS} ${RLU_PROXY_ARGS}"

Configuring the web application server to use a SOCKS proxy

To connect to HCL Unica through a SOCKS proxy, you must modify the configuration of the web application server. For Unica WebSphere servers, you modify the generic JVM arguments. For Oracle Weblogic servers, you modify the SetDomainEnv script.

• If your web application server is Unica WebSphere, add the following to the WebSphere generic JVM arguments.

-Dhcl.unica.deliver.https.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.https.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.https.proxy.type=SOCKS
-Dhcl.unica.deliver.ftps.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.ftps.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.ftps.proxy.match.hosts=<comma separated list of host names and IP addresses>
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUTH_USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUTH_PASSWORD>
• If your web application server is Oracle Weblogic, modify the `setDomainEnv` script. The required settings depend on your operating system.

In Windows environments, make the following changes:

```bash
JAVA_OPTIONS =${JAVA_OPTIONS}
-Dhcl.unica.deliver.https.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.https.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.https.proxy.type=SOCKS
-Dhcl.unica.deliver.ftps.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.ftps.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.ftps.proxy.match.hosts=<comma separated list of host names and IP addresses>
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUTH_USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUTH_PASSWORD>
```

In UNIX environments, make the following changes:

```bash
JAVA_OPTIONS ='${JAVA_OPTIONS}
-Dhcl.unica.deliver.https.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.https.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.https.proxy.type=SOCKS
-Dhcl.unica.deliver.ftps.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.ftps.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.ftps.proxy.match.hosts=<comma separated list of host names and IP addresses>
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUTH_USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUTH_PASSWORD>,'
```
Data download frequency and port setting

A Deliver component called the Response and Contact Tracker (RCT) is installed as part of your Unica Campaign installation. The RCT regularly requests email response and tracking data from HCL Unica. By default, the RCT issues a data request every 5 seconds.

The RCT issues data requests over HTTPS (HTTP over SSL). HCL Unica hosted services accept HTTPS connection requests on port 443, and only from hosts that you have specified during the hosted email account startup process.

Configuring a system user to access HCL Unica hosted services

Deliver components must be able to access the HCL Unica hosted services without requiring manual entry of login credentials. To establish automatic login, define a system user in Platform that can provide the required access credentials.

To simplify user administration and troubleshooting, you can modify an existing system user to access hosted services and local system tables. You can configure a single system user to provide credentials for multiple systems. For example, modifying the configuration of the Campaign system user creates a single user that can automatically access HCL Unica hosted services and the Deliver system tables in the Campaign schema.

The credentials required to access HCL Unica hosted services are the user name and password that Unica provides for your hosted messaging account. The credentials that you use depend on whether you are connecting to the Unica US data center or the data center that Unica maintains in the Europe. Consult with Unica to determine which data center you use.

For specific information about how to configure a system user to communicate with HCL Unica hosted services, see the Unica Deliver Startup and Administrator's Guide.

For general information about how to create system users and data sources, see the Unica Platform Administrator's Guide.
Configuring partition access to HCL Unica hosted services

Unica Deliver components in the partition must be allowed to automatically provide valid login credentials when attempting to communicate with HCL Unica hosted services. To do this, you must add the HCL Unica hosted services login credentials to a Platform user. This user becomes the Deliver system user.

You can add the platform data source containing the HCL Unica hosted services credentials to the Deliver system user. This user can be the same system user that accesses the Campaign system tables in the partition.

The steps for configuring system users for a partition are the same as those followed during the initial Deliver installation, which created the first partition. For details about how to add HCL Unica hosted services login credentials to a system user, see the Unica Deliver Startup and Administrator's Guide.

The credentials required to access HCL Unica hosted services are the user name and password that Unica provided during the initial startup process.

⚠️ **Important**: For each additional partition, you must request a separate user name and password from Unica.

Configuring the system user that accesses HCL Unica hosted services

Deliver components in Campaign must be able to access HCL Unica hosted services automatically, without prompting for a login. System users that are configured in Unica Platform can reference a data source that provides the required user name and password. You can add the data source to a new system user or to an existing system user. To simplify user administration, you can update a system user that is already configured to access the Campaign schema so that it can also access HCL Unica hosted services.

To complete this task, you must know the HCL Unica hosted services user name and password that Unica assigned to your hosted email account. Receiving the user name and password is part of the account startup process.
You must have the appropriate access permissions and know how to create system users and data sources in the Unica Platform.

⚠️ **Note:** If your installation contains multiple partitions, you must complete this task for each partition. You cannot share system users across partitions.

1. Create a Platform data source to contain the user name and password that is required to access HCL Unica hosted services. For best results and easier maintenance, name this data source UNICA_HOSTED_SERVICES. Configure this data source as follows.

   For **Data Source Login**, enter the user name that you received from Unica during account startup.

   For **Data Source Password**, enter the password that you received Unica during account startup.

2. Specify the data source in the Deliver configuration. Use the `amDataSourceForAcctCredentials` configuration property.

   The configuration property is at `Deliver > partitions > partition[n] > hostedAccountInfo > amDataSourceForAcctCredentials`.

   By default, the specified data source is **UNICA_HOSTED_SERVICES**.

3. Specify a system user to access HCL Unica hosted services. You can specify an existing user or create a user. In the Deliver configuration, use the `amUserForAcctCredentials` configuration property.

   The configuration property is at `Deliver > partitions > partition[n] > hostedAccountInfo > amUserForAcctCredentials`.

   By default, the specified user is **asm_admin**.

4. Add the data source that is configured in Step 1 to the system user specified in Step 3.

   You must restart the web application server for the configuration changes to take effect.
Configuring addresses for connecting to HCL Unica hosted services

To ensure proper connection to HCL Unica hosted services, you must enter the addresses as values for configuration properties in the Deliver configuration. The connection addresses that you enter depend on whether you are connecting to the Unica U.S. data center or to the Unica data center in the Europe.

Consult with Unica to confirm which data center your hosted email account uses.

In Unica Platform, navigate to **Settings > Configuration**. In the Deliver configuration, navigate to the following Deliver configuration properties and confirm or update the connection settings, depending on data center your account uses.

- **Deliver > serverComponentsAndLocations > hostedServices> uiHostName**
  
  To connect to Unica data center in US, change this value to `em.unicadeliver.com`.
  
  To connect to the Unica data center in Europe, change this value to `em-eu.unicadeliver.com`.

- **Deliver > serverComponentsAndLocations > hostedServices> dataHostName**
  
  To connect to Unica data center in US, change this value to `em.unicadeliver.com`.
  
  To connect to the Unica data center in Europe, change this value to `em-eu.unicadeliver.com`.

- **Deliver > serverComponentsAndLocations > hostedServices> ftpHostName**
  
  To connect to Unica data center in US, change this value to `ftp-em.unicadeliver.com`.
  
  To connect to the Unica data center in Europe, change this value to `ftp-eu.unicadeliver.com`.

If you change a configuration property, restart the web application server to apply the changes.
IP address of Deliver host names

If you require to whitelist IP address for Deliver host names on your corporate firewall, use the following IP addresses.

- `em.unicadeliver.com:13.248.215.130` and `76.223.84.165`
- `em-eu.unicadeliver.com:75.2.15.173` and `99.83.137.137`
- `ftp-em.unicadeliver.com:192.190.152.236`
- `ftp-eu.unicadeliver.com:192.190.153.236`
- `tms-us.unicadeliver.com:13.248.172.132` and `76.223.38.158`
- `tms-eu.unicadeliver.com:75.2.31.132` and `99.83.164.171`

Configuring secure communication for hosted email

Communications between the email marketer and HCL Unica hosted services occur over Secure Sockets Layer (SSL). You must change the web application server configuration to use SSL. Making the required changes requires using the Java™ keytool utility.

Configuring the secure communication involves the following actions.

- Generate a trusted keystore.
- Obtain a digital certificate from HCL Unica hosted services.
- Add the trusted keystore to the web application server.
- Import the HCL Unica hosted services digital certificate into the trusted keystore.

The exact steps and sequence required to configure SSL depend on the type and version of web application server (WebSphere, WebLogic, Tomcat, JBOSS) on which you deployed Unica Platform and Unica Campaign.

For WebLogic, see Configuring SSL when using WebLogic (on page 36).

For WebSphere, see Configuring SSL when using WebSphere (on page 39).
Generating a trusted keystore

Follow this procedure to create an identity keystore and a trusted keystore for configuring Unica Deliver to communicate with HCL Unica hosted services over SSL. You add the keystores to the web application server when you configure SSL.

HCL uses the following sample values in the procedures contained in this section.

- **Identity keystore**: HCLUnicaClientIdentity.jks
- **Alias for the identity keystore**: HCLUnicaClientIdentity
- **Password (-storepass) for the identity keystore**: clientPwd
- **The security key (-keypass) for the identity keystore**: clientPwd
- **Certificate based on the identity keystore**: ClientCertificate.cer
- **Trusted keystore**: HCLUnicaTrust.jks
- **Password (-storepass) for the trusted keystore**: trustPwd

The actual values that you enter must be specific to your installation.

To complete steps in this procedure, run the Java `keytool` utility from the command line.

1. Generate an identity keystore. Use the `genkey` command, as shown in the following example.

   The example creates an identity keystore named HCLUnicaClientIdentity.jks. You can use a different name for the identity keystore that you create.

   ```bash
   keytool -genkey -alias HCLUnicaClientIdentity -keyalg RSA -keystore <HCLUnicaClientIdentity.jks> -keypass <clientPwd> -validity 1000 -dname "CN=hostName, O=myCompany" -storepass <clientPwd>
   ```

   Note the following.

   - You use the values for alias, keystore, keypass, and storepass later in this procedure and when you configure SSL in the web application server.
   - For WebSphere, the keystore password (-storepass) and the key password (-keypass) must be the same.
• In the distinguished name (-dname) the common name (CN) is the same as the host name used to access HCL Unica hosted services. For example, if the URL for HCL Unica hosted services is https://hostName.example.com:7002/unica/jsp, then the CN is hostName.example.com. The CN portion of the distinguished name is the only required portion; Organization (O) and Organizational Unit (OU) are not required.

2. Generate a certificate based on the identity keystore. Use the export command, as shown in the following example.

   The example generates a certificate named ClientCertificate.cer. You can use a different name for the certificate that you create.

   The values for keystore, storepass, and alias must match the values you specified for the identity keystore.

   keytool -export -keystore <HCLUnicaClientIdentity.jks> -storepass <clientPwd> -alias HCLUnicaClientIdentity -file <ClientCertificate.cer>

3. Generate the trusted keystore. Use the import command, as shown in the following example.

   The example generates a trusted keystore named HCLUnicaTrust.jks. You can use a different name for the trusted keystore that you create.

   keytool -import -alias HCLUnicaClientIdentity -file <ClientCertificate.cer> -keystore <HCLUnicaTrust.jks> -storepass <trustPwd>

   Type Y when prompted to trust the certificate.

   Note the values that you defined for the following variables. Your values can be different from the values given in the example.

   • alias (in the example: HCLUnicaClientIdentity)
   • identity keystore (in the example: HCLUnicaClientIdentity.jks)
   • storepass (in the example: trustPwd) The storepass value for the trusted keystore can be different from the storepass value for the identity keystore and certificate.
Connections to message services

• keystore (in the example: HCLUnicaTrust.jks) Depending on your web application server, you also specify the identity keystore.

You specify these installation-specific values when you configure SSL on the web application server for your HCL Unica installation.

Configuring SSL when using WebLogic

This section describes the steps required to configure SSL if you deploy HCL Unica components on Oracle WebLogic. This change is required to allow Deliver components that operate inside Campaign to communicate with HCL Unica hosted services over SSL.

For specific guidance regarding navigation and working with the Oracle WebLogic user interface, consult the documentation for the specific Oracle WebLogic version you are using.

Complete the following tasks.

• Modify the WebLogic startup script
• Modify the WebLogic configuration
• Obtain a digital certificate from HCL Unica hosted services
• Create a trusted keystore and import the Unica digital certificate

Modifying the WebLogic startup script

If you deployed Campaign on WebLogic, you must modify the WebLogic startup script and the WebLogic configuration for SSL so that WebLogic recognizes and accepts secure communication between locally installed Deliver components and HCL Unica hosted services.

Add the following arguments to the JAVA_OPTIONS in the WebLogic startup script.

• -Dweblogic.security.SSL.allowSmallRSAExponent=true
• WebLogic version 12c or higher: -Dweblogic.security.SSL.protocolVersion=TLS1
Modify the WebLogic configuration

You must change the SSL configuration in WebLogic.

Use the WebLogic console to make the following change in the WebLogic SSL configuration for your domain.

Change the **Hostname Verification** setting to **None**.

Obtain a certificate from HCL Unica hosted services

To configure SSL communication, you must download a digital certificate from HCL Unica. The certificate details are saved to a file with the `.cer` extension that you can import into the web application server keystore.

You lose access to HCL Unica hosted services when your existing SSL certificate expires. Use this procedure to download a new certificate.

1. In Internet Explorer, log into the address for HCL Unica hosted services that has been configured for your hosted email account.
   - For the US data center, go to [https://em.unicadeliver.com](https://em.unicadeliver.com)
   - For the Europe data center, go to [https://em-eu.unicadeliver.com](https://em-eu.unicadeliver.com)

   The login attempt results in a failed login, but allows you to use the browser to submit the certificate request.

2. Click the lock icon and select **View Certificate**.

3. Select the Details tab and select **Copy to File**.

   Save the file with a `.cer` extension to a location that is accessible to the web application server. The file that you create is the digital certificate that you insert into the keystore on the web application server.

   For example, save the certificate as **HCLHosted.cer**.
Create a trusted keystore for WebLogic and import the Unica certificate

For Weblogic, you must create a trusted keystore that accepts the Unica certificate.

Before you begin, use a web browser to download the HCL Unica hosted services digital certificate and save it as a .cer file. For example, the certificate can be named HCLHosted.cer (your file name can be different). For additional details, see Obtain a certificate from HCL Unica hosted services (on page 37).

HCL uses the following sample values in the procedures contained in this section.

- **Identity keystore**: HCLUnicaClientIdentity.jks
- **Password for the identity keystore**: clientPwd
- **Trusted keystore**: HCLUnicaTrust.jks
- **Alias for the trusted keystore**: HCLUnicaHostedIdentity
- **Password (-storepass) for the trusted keystore**: trustPwd
- **Digital certificate (-file) from Unica**: HCLHosted.cer

The actual values that you enter must be specific to your installation.

To complete steps in this procedure, run the Java keytool utility from the command line.

1. Generate a trusted keystore for WebLogic.
   
   For details, see Generating a trusted keystore (on page 34).

   You specify the identity keystore and the trusted keystore in the WebLogic configuration.

2. Use the import command in the keytool utility to add the HCL Unica hosted services certificate to the trusted keystore created in Step 1, as shown in the following example. Use the digital certificate that you downloaded from Unica.

   In this procedure, you also define an alias for the trusted keystore.

   ```
   keytool -import -alias HCLUnicaHostedIdentity -file <HCLHosted.cer> -keystore <HCLUnicaTrust.jks> -storepass <trustPwd>
   ```

   Type Y when prompted to trust the certificate.
3. In the WebLogic administration console, configure the keystores for the server.

To specify the configuration rules, select the option for Custom Identity and Custom Trust keystores from the available choices. For the Custom Identity, you specify the identity keystore. For the Custom Trust, you specify the trusted keystore.

For example, in the administration console, specify the following (using the example values from the trusted keystore you created in Step 1).

- For the **Identity**: specify the identity keystore and associated password.
  
  For example, `HCLUnicaClientIdentity.jks` and `clientPwd`.

- For the **Trust**: specify the trusted keystore and associated password.
  
  For example, `HCLUnicaTrust.jks` and `trustPwd`.

Specify the full path to both keystores.

4. Restart WebLogic. WebLogic does not implement the configuration changes until you restart the web application server.

5. To test the SSL connection, log in to Unica Campaign and access various messaging features menus. Confirm that you can create email, landing pages, and mailings.

### Configuring SSL when using WebSphere

This section describes the general steps required to configure SSL if you have deployed HCL Unica components on WebSphere. This change is required to allow Deliver components that operate inside Campaign to communicate with HCL Unica hosted services over SSL.

Before you begin, you will need to know the value for the configuration property `uiHostName`. The value for `uiHostName` is the URL for HCL Unica hosted services. For details, see [Configuring addresses for connecting to HCL Unica hosted services](on page 32).

You must access the WebSphere security console to modify settings for SSL certificate and key management. This task requires a restart of the Campaign web application server to implement the changes.
If you have deployed Campaign on WebSphere, you must modify the WebSphere security configuration to retrieve the signer certificate from HCL Unica and add it to the WebSphere trust store. If you receive an error message indicating that your current signer certificate has expired, delete the current certificate and add a new one.

For specific guidance regarding navigation and working with the WebSphere user interface, consult the documentation for the specific Unica WebSphere version you are using.

1. Generate a trusted keystore.

   For additional details, see Generating a trusted keystore (on page 34).

   To configure SSL, you need to specify the values that you define for the following variables. The values shown are for example only. Your values can be different.

   - **alias**: `UnicaClientIdentity` (example)
   - **keystore**: `HCLUnicaTrust.jks` (example)
   - **storepass**: `trustPwd` (example)

2. Select the new keystore in the WebSphere security console.

   For example, if you followed the example in Step 1, select `HCLUnicaTrust.jks`.

3. Obtain a security certificate from HCL Unica and import it into WebSphere, as described in the following steps.

   a. In the WebSphere security console, navigate to SSL certificate and key management > Key stores and certificates > NodeDefaultTrustStore > Signer certificates. Select the option to Retrieve from port.

   b. Configure WebSphere to establish a test connection to retrieve the signer certificate from HCL Unica. Enter the following values for the HCL Unica signer certificate.

      - **Host** The value that is defined for Deliver >serverComponentsAndLocations > hostedServices >uiHostName
      - **Port** 443
      - **SSL configuration for outbound connection** NodeDefaultSSLSettings
      - **Alias** The value that you entered for Host
When you have finished, WebSphere communicates with HCL Unica hosted services to retrieve the information required to create a signer certificate for HCL Unica hosted services.

4. After WebSphere finishes creating the signer certificate, select the new certificate in the security console.

The web application server uses the new certificate when establishing connections to HCL Unica.

5. Restart WebSphere

WebSphere does not implement the configuration changes until you restart the web application server.

For additional information about supported WebSphere versions for deploying Unica products, see the Recommended Software Environments and Minimum System Requirements document for each product.
Chapter 4. Response and Contact Tracker operation

The Response and Contact Tracker (RCT) is installed in your local environment and communicates with HCL Unica hosted services to retrieve and process data for email contacts, email delivery, and recipient responses, such as link clicks and opens. The RCT must be running in order to retrieve link tracking and email delivery notification data from HCL Unica hosted services.

You can start the RCT in either of the following ways.

• Start the RCT manually.
• Start the RCT as a service

⚠️ Important: You must start the RCT manually the first time you use Deliver, even if you registered the RCT as a service.

You must restart the RCT when you make changes to configuration properties for Deliver. You can restart the RCT at any time, even if you have configured it to run as a service. HCL Unica hosted services continue to store tracking data if the RCT is shut down or restarting. When it resumes operation, the RCT downloads the queued information.

Manual operation of the Response and Contact Tracker

To operate the Response and Contact Tracker (RCT) manually, run the `rct` script in the `bin` directory in your Deliver installation.

• To start the RTC, run the `rct` script in the `bin` directory under your Deliver installation, as follows.

  `rct start`

• To stop the RCT, run the `rct` script as follows.
Adding the Response and Contact Tracker as a service

You can configure the Response and Contact Tracker (RCT) to start automatically by adding it as a service.

Register the RCT service by running the MKService_rct script that is provided with the Deliver software.

To add the Response and Contact Tracker (RCT) as a service, run the MKService_rct -install script from the bin directory under your Deliver installation.

The bin directory is created as a subdirectory in the Campaign installation directory when you install or upgrade to the latest version of Unica Campaign.

In UNIX or Linux™, run this script with a user that has root permissions or permissions to create daemon processes.

In Windows, the name of the service is Response & Contact Tracker.

After you run the MKService_rct script, start the RCT manually with the rct script. You manually restart the RCT only once. After you start the RCT manually the first time, the RCT restarts automatically every time you restart the operating system of the computer where you installed the RCT.

After configuring the RCT service, you can prevent the RCT from starting automatically by running the MKService_rct script with the -remove option.
Removing the Response and Contact Tracker service

If you have installed the Response and Contact Tracker (RCT) as a service, the RCT restarts each time you restart the system where you installed the RCT. To prevent the RCT from restarting automatically, you must remove the Response and Contact Tracker (RCT) service.

To remove the RCT as a service, run the MKService_rct script with the -remove option.

From a Windows command line, in your HCL Unica home directory, run Deliver\bin \MKService_rct.bat -remove.

In UNIX or Linux, in your HCL Unica home directory, run Deliver/bin/MKService_rct.sh -remove.

For more information about this script, see The MKService_rct script (on page 128)
Chapter 5. Startup verification

To ensure access to all hosted email features, test the configurations and connections for your Campaign and Deliver installations after you enable Deliver, expand your Deliver installation, or upgrade the Campaign installation.

Verify configurations and connections after you do any of the following.

- Enable Deliver for the first time
- Upgrade your current Unica Campaign installation
- Add a new partition to the Deliver configuration maintained in Unica Platform

Confirmation for system configurations

To ensure that startup preparations are complete, confirm that the following configuration properties are set and that the settings meet the requirements for your Deliver and Campaign installations.

<table>
<thead>
<tr>
<th>Configuration property</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Campaign</td>
<td>partitions</td>
</tr>
<tr>
<td>partition[n]</td>
<td>Deliver</td>
</tr>
<tr>
<td>Campaign</td>
<td>partitions</td>
</tr>
<tr>
<td>partition[n]</td>
<td>server</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration property</td>
<td>Setting</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>default partition. When you set this property to <strong>Yes</strong>, Deliver features become available in the Campaign interface.</td>
</tr>
<tr>
<td>Deliver</td>
<td>serverComponentsAndLocations</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Deliver</td>
<td>serverComponentsAndLocations</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Deliver</td>
<td>serverComponentsAndLocations</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration property</td>
<td>Setting</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Deliver</td>
<td>partitions</td>
</tr>
<tr>
<td></td>
<td>The HCL Unica user that references the data source that contains the HCL Unica hosted services access credentials. You configure this value when you create a system user to access the email resources that are hosted by Unica.</td>
</tr>
<tr>
<td>Deliver</td>
<td>partitions</td>
</tr>
<tr>
<td></td>
<td>The Platform data source that contains the HCL Unica hosted services login credentials. You configure this value when you create a system user to access the email resources that are hosted by Unica.</td>
</tr>
<tr>
<td>Deliver</td>
<td>partitions</td>
</tr>
<tr>
<td></td>
<td>Type of database that hosts the system tables. Provide the correct value for your database.</td>
</tr>
<tr>
<td>Deliver</td>
<td>partitions</td>
</tr>
<tr>
<td></td>
<td>Name of the database schema for the system tables. Set to the appropriate schema name for your database.</td>
</tr>
<tr>
<td>Deliver</td>
<td>partitions</td>
</tr>
<tr>
<td></td>
<td>JDBC driver for system tables. Provide the correct value for your environment.</td>
</tr>
<tr>
<td>Deliver</td>
<td>partitions</td>
</tr>
<tr>
<td></td>
<td>JDBC connection URI for system tables. Provide the correct value for your environment. Specify the database type, database driver, host, port, and database name. For example: jdbc:oracle:thin:@yourdb.example.com:1234:DBname</td>
</tr>
<tr>
<td>Configuration property</td>
<td>Setting</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>Consult your database documentation for specific instructions regarding how to construct the JDBC URL. The value that you enter must exactly match the value that is defined in your Campaign web server.</td>
</tr>
<tr>
<td>Deliver</td>
<td>partitions</td>
</tr>
<tr>
<td></td>
<td>The HCL Unica user that references the data source that contains the system tables login credentials. You create this user when you configure access to the local Deliver system tables.</td>
</tr>
<tr>
<td>Deliver</td>
<td>partitions</td>
</tr>
<tr>
<td></td>
<td>The Platform data source that contains login credentials to the database that contains the system tables. You create this data source when you create a user to access the Deliver system tables.</td>
</tr>
</tbody>
</table>

**Testing upload to HCL Unica hosted services**

To test the ability to upload data to HCL Unica hosted services from your local environment, run the `rlu` script in check mode.

In the `bin` directory under your Deliver installation, run the `rlu` script in either of the following ways.

- `rlu -c`
- `rlu --check`
Testing download from HCL Unica hosted services

To test the ability to download information from HCL Unica hosted services, run the rct script in check mode.

In the bin directory under your Deliver installation, run the rct script as follows.

rct check

Testing the connection to the hosted messaging interface

Unica hosts the messaging interface from its data centers in the US and Europe. Test the connection to the hosted mailing interface by attempting to access an Deliver feature.

Log in to HCL Unica and select Deliver mailings from the Campaign menu.

If the connection to the Deliver user interface is established correctly, the Deliver mailings page opens and displays a list of mailings and related mailing characteristics.

If the connection to the user interface is not established properly, you see an error.
Chapter 6. Configurations for Unica Deliver

The Unica Platform provides various configuration properties to modify the behavior and appearance of Deliver. Some configuration properties are set during installation. You can change configuration properties at any time.

After you update the Campaign or Deliver configurations, you must restart the Response and Contact Tracker (RCT) and the web application server that hosts Campaign.

<table>
<thead>
<tr>
<th>Characteristic or feature</th>
<th>Configuration property (including path)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable or disable Deliver in the Campaign partition.</td>
<td>Campaign</td>
</tr>
<tr>
<td>Characteristics of email recipient lists.</td>
<td>Campaign</td>
</tr>
<tr>
<td>URLs required to connect to HCL Unica hosted services.</td>
<td>Deliver</td>
</tr>
<tr>
<td>Database and account access credentials for connecting to HCL Unica hosted services.</td>
<td>Deliver</td>
</tr>
<tr>
<td>Characteristic or feature</td>
<td>Configuration property (including path)</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Database access and schema settings for the Deliver system tables.</td>
<td>Deliver</td>
</tr>
<tr>
<td>Location of a script that runs in response to the actions or status of the Recipient List Uploader. (optional)</td>
<td>Deliver</td>
</tr>
<tr>
<td>Settings related to data download, processed by the Response and Contact Tracker (RCT).</td>
<td>Deliver</td>
</tr>
<tr>
<td>Support for presenting lists of personalized data in Deliver based on dimension tables in Campaign.</td>
<td>Campaign</td>
</tr>
<tr>
<td>Support for tracking mailing execution history.</td>
<td>Deliver</td>
</tr>
<tr>
<td></td>
<td>See the enableExecutionHistoryDataTracking parameter.</td>
</tr>
<tr>
<td>Characteristic or feature</td>
<td>Configuration property (including path)</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>responseContactTracker (on page 72)</td>
<td></td>
</tr>
</tbody>
</table>

For more information about working with configuration properties, see the Unica Platform Administrator's Guide.

### Configuring access to additional mailing execution history

You can request that Unica provide additional data for mailing execution history. Access to additional mailing execution history data is available by request from Unica and by updating the Deliver configuration. Data for mailing execution history is recorded your local Deliver system tables in the UACE_ExecHistory table to describe completed mailing runs.

To download additional mailing run data, you must update the configuration property enableExecutionHistoryDataTracking. By default, enableExecutionHistoryDataTracking is not exposed in the Deliver configuration properties.

You can display this configuration property in your local Deliver installation by running the switch_config_visibility.bat script, which is in the Deliver\tools directory. The following types of records are available in additional mailing execution history.

- Message subject line
- From address
- User that updated the mailing
- Document description
- Mailing save date

1. Request access to additional mailing execution history data. To request access, contact your Unica Deliver Services team via HCL technical support.

2. Update the Deliver configuration. Configure the following configuration property.
Affinium|deliver|partitions|partition1|responseContactTracker|
enableExecutionHistoryDataTracking

Set `enableExecutionHistoryDataTracking` to **True**.

3. Verify that the table mapping for UACEExecHistory is correct.

You can query the Deliver system tables to retrieve mailing run information from the UACEExecHistory table.

For more information about the Deliver system tables, see the Unica Deliver System Tables and Data Dictionary.

**Support for Campaign offer integration**

Unica Deliver supports adding offers that are configured in Campaign to personalized email created in Deliver.

The offers are based on offer templates that are configured in Unica Campaign. To support integrating Campaign offers into personalized email, you must update the `contactAndResponseHistTracking` property in the Campaign configuration and complete other configurations in Campaign.

For more information about how to configure support for offer integration, see topics for Deliver offer integration in the Unica Campaign Administrator’s Guide.

**Configuring support for dimension tables**

To support certain features that are provided by advanced scripts for email, the `oltDimTableSupport` configuration property must be set to **True**.

Deliver provides advanced scripts to create email messages that display lists of personalized information. These lists require associating dimension tables created in Campaign with an Output List Table (OLT) that defines the email recipient list. Output List Tables are created in the Deliver schema.
The `oltDimTableSupport` configuration property controls support for creating dimension tables in the Deliver schema. When the value for this property is set to `True`, an OLT can use information provided in a dimension table.

Complete the following steps to update the `oltDimTableSupport` property.

For more information about how marketers use advanced scripts to create data tables, see the Unica Deliver User's Guide.

1. Go to **Settings > Configuration > Campaign > Partitions > partition[n] > Deliver**
2. Click **Edit Settings** and set the value of the `oltDimTableSupport` property to `True`.

### Configuring access to local Deliver system tables

Deliver components must be able to access the Deliver system tables in the Campaign schema. You must create and configure a system user that can access the system tables automatically. The system user that was configured during the installation of Campaign already has the necessary access to the Campaign schema.

**Note:** If your installation contains multiple partitions, you must complete this task for each partition. You cannot share system users across partitions.

If you want to use a different system user to access the Deliver system tables, you must create a new system user in the Platform and create new platform data source with access to the Campaign schema.

1. In the Deliver configuration, specify a system user that accesses the database that hosts the Campaign schema.

   You can create a new user or specify an existing user. The system user that you set up for Campaign already has access to the Campaign schema.

   **Use the configuration property** `Deliver > partitions > partition [n] < dataSources > systemTables > asmUserForDBCredentials`

   **By default the specified user is** `asm_admin`. 
2. In the Deliver configuration, specify the data source that is configured to contain the user name and password that is required to access the database that hosts the Campaign schema.

You can use the data source that was created to access the Campaign schema when you installed Campaign.

Use the configuration property `Deliver > partitions > partition [n] < dataSources > systemTables > amDataSourceForDBCredentials`.

Configuring Journey - Deliver integration

In Journey - Deliver integration, to make the email template visible in Unica Journey, run `switch_config_visibility` utility and restart the web server. Perform the following steps.

1. Open command prompt and navigate to `<Campaign_Home>/Deliver/tools`.
2. Execute the following commands of `switch_config_visibility`.

   - `./switch_config_visibility.sh -p "Affinium|Deliver|serverComponentsAndLocations|hostedServices|ftpProtocol" -v true`
   - `./switch_config_visibility.sh -p "Affinium|Deliver|serverComponentsAndLocations|hostedServices|uiHTTPProtocol" -v true`
   - `./switch_config_visibility.sh -p "Affinium|Deliver|serverComponentsAndLocations|hostedServices|dataHTTPProtocol" -v true`
   - `./switch_config_visibility.sh -p "Affinium|Deliver|serverComponentsAndLocations|hostedServices|ftpPort" -v true`
   - `./switch_config_visibility.sh -p "Affinium|Deliver|serverComponentsAndLocations|hostedServices|uiHTTPPort" -v true`
   - `./switch_config_visibility.sh -p "Affinium|Deliver|serverComponentsAndLocations|hostedServices|dataHTTPPort" -v true`
   - `./switch_config_visibility.sh -p "Affinium|Deliver|serverComponentsAndLocations|hostedServices|mailingUIHTTPPath" -v true`
   - `./switch_config_visibility.sh -p "Affinium|Deliver|serverComponentsAndLocations|hostedServices|dataHTTPPath" -v true`
3. Restart the web server.

Configuration properties for Deliver

You access Deliver configuration properties from the Settings menu in the Platform. Properties for configuring Deliver are contained in the Campaign and Deliver configuration categories.

To access the configuration properties, navigate to **Settings > Configurations**. The Configurations page lists all of the available configuration properties for your HCL Unica installation.

**Campaign | partitions | partition[n] | Deliver**

Define properties in this category to define characteristics of recipient lists and specify the location of resources that upload the lists to HCL Unica.

**DeliverPluginJarFile**

**Description**

Complete path to the location of the file that operates as the Recipient List Uploader (RLU). This plug-in to Campaign uploads OLT data and associated metadata to the remote services hosted by Unica. The location that you specify must be the full local directory path in the file system for the computer that hosts the Campaign web application server.

The Unica installer populates this setting automatically for the default partition when you run the installer. For other partitions, you must configure...
this property manually. Because there is only one RLU for each Deliver installation, all partitions must specify the same location for the RLU.

Do not change this setting unless Unica instructs you to do so.

**Default value**

No default value defined.

**Valid Values**

Full local directory path to where you installed the Campaign web server.

**defaultSeedInterval**

**Description**

The number of messages between seed messages if defaultSeedType is Distribute list.

**Default value**

1000

**defaultSeedType**

**Description**

The default method that Deliver uses to insert seed addresses into a recipient list.

**Default value**

Distribute IDS

**Valid Values**

- Distribute IDS - Distribute IDs evenly, based on the size of the recipient list and the number of seed addresses available, inserts seed addresses at equal intervals throughout the entire recipient list.
- Distribute list - Insert seed address for every defaultSeedInterval IDs in main list. Inserts the entire list of available seed addresses at
specified intervals throughout the recipient list. You must specify the interval between insertion points.

oltTableNamePrefix

**Description**

Used in the generated schema for the output list table. You must define this parameter.

**Default value**

OLT

**Valid Values**

The prefix can contain no more than 8 alphanumeric or underscore characters, and must start with a letter.

oltDimTableSupport

**Description**

This configuration parameter controls the ability to add dimension tables to output list tables (OLT) created in the Deliver schema. Dimension tables are required to use advanced scripting for email to create data tables in email messages.

You must set this property to **True** (by default it is **True**) so that marketers can create dimension tables when they use the Deliver process to define a recipient list. For more information about creating data tables and working with advanced scripts for email, see the Unica Deliver User Guide.

You must set this property to **False**, if you are using dimension table fields to output in olt and want to use these dimension fields in communication as a personalisation field.

**Default value**

**True**
Valid Values

True | False

Campaign | partitions | partition[n] | server | internal

Properties in this category specify integration settings and the internalID limits for the selected Campaign partition. If your Campaign installation has multiple partitions, set these properties for each partition that you want to affect.

internalIdLowerLimit

Configuration category

Campaign|partitions|partition[n]|server|internal

Description

The internalIdUpperLimit and internalIdLowerLimit properties constrain the Campaign internal IDs to be within the specified range. Note that the values are inclusive: that is, Campaign may use both the lower and upper limit.

Default value

0 (zero)

internalIdUpperLimit

Configuration category

Campaign|partitions|partition[n]|server|internal

Description

The internalIdUpperLimit and internalIdLowerLimit properties constrain the Campaign internal IDs to be within the specified range. The values are inclusive: that is, Campaign may use both the lower and upper limit. If Unica Collaborate is installed, set the value to 2147483647.

Default value
deliverInstalled

**Configuration category**

Campaign|partitions|partition[n]|server|internal

**Description**

Indicates that Deliver is installed. When you select Yes, Deliver features are available in the Campaign interface.

The Unica installer sets this property to Yes for the default partition in your Deliver installation. For additional partitions where you installed Deliver, you must configure this property manually.

**Default value**

No

**Valid Values**

Yes | No

Legacy_campaigns

**Configuration category**

Campaign|partitions|partition[n]|server|internal

**Description**

For this partition, enables access to campaigns created before Unica Plan and Campaign were integrated. Applies only if MO_UC_integration is set to Yes. Legacy campaigns also include campaigns created in Campaign 7.x and linked to Plan 7.x projects. For more information, see the Unica Unica Plan and Campaign Integration Guide.

**Default value**

No
Valid Values

Yes | No

Campaign | partitions | partition[n] | Deliver | contactAndResponseHistTracking

Use the properties in this category to configure Deliver offer integration with Unica Campaign for the current partition.

etlEnabled

Description

Campaign uses its own ETL process to extract, transform, and load offer response data from the Deliver tracking tables into the Campaign contact and response history tables.

The ETL process coordinates information across the necessary tables, including `UA_UsrResponseType` (Campaign response types) and `UA_RespTypeMapping` (mapping of response types between Campaign and Deliver).

Setting the value to Yes ensures that information about Deliver offer contact and response history is coordinated between Campaign and Deliver. For example, email response data will be included in Campaign reports.

Note: You must also set Campaign | partitions | partition[n] | server | internal | DeliverInstalled to Yes for this partition or the ETL process will not run.

Tip: If you want to monitor the progress of the ETL, enable Campaign | monitoring | monitorEnabledForDeliver.

Default value

No
Valid values

| Yes | No |

runOnceADay

Description

Indicate whether the ETL process should run only once a day.

If the value is Yes: You must specify a `startTime`; the ETL job then runs until all of the records are processed; and the `sleepIntervalInMinutes` is ignored.

If the value is No: The ETL job starts as soon as the Campaign web server starts. The ETL job stops after all of the records are processed, then waits for the time specified by `sleepIntervalInMinutes`.

Default value

No

Valid values

| Yes | No |

batchSize

Description

The ETL process uses this parameter to fetch records that have been downloaded by the RCT into the local Deliver system tables. Because large values can impact performance, the list of available values is restricted to the valid values shown below. If you anticipate large volumes of records, adjust the `batchSize` together with the `sleepIntervalInMinutes` to process records at regular intervals.

Default value

100

Valid values
sleepIntervalInMinutes

Description
Specify the interval in minutes between ETL jobs. This option determines the wait time after a job finishes. The ETL process waits for this duration before starting the next job. Multiple jobs can run synchronously and there may be multiple ETL jobs per partition.

If runOnceADay is Yes, you cannot set a sleep interval.

Default value
60

Valid values
Positive integers

startTime

Description
Specify a time to start the ETL job. You must use the English locale format to specify the start time.

Default value
12:00:00 AM

Valid values
Any valid time in the format hh:mm:ss AM/PM

notificationScript

Description
An optional executable or script file that is run after each ETL job is done. For example, you might want to be notified of the success or failure of each ETL
job, for monitoring purposes. The notification script runs every time the ETL job for a given partition finishes running.

The parameters passed in to this script are fixed and cannot be changed. The following parameters can be used by the script:

- `etlStart`: The start time of ETL in number of milliseconds.
- `etlEnd`: The end time of ETL in number of milliseconds.
- `totalCHRecords`: Total number of contact records processed.
- `totalRHRecords`: Total number of response history records processed.
- `executionStatus`: Execution status of the ETL with value 1 (failed) or 0 (succeeded).

**Default value**

No default value defined.

**Valid values**

Any valid path that the Campaign server can access with Read and Execute permissions. For example: `D:\myscripts\scriptname.exe`

---

**Deliver | serverComponentsAndLocations | hostedServices**

Define properties to specify the URLs for connecting to HCL Unica hosted services. Deliver uses separate connections for uploading recipient lists, metadata that describes recipient lists, and for general communication sent to the hosted environment.

You must change the default values if you are connecting to HCL Unica hosted services through the data center that is established by Unica in the Europe. Consult Unica to determine the data center to which you are connected.

**uiHostName**

**Description**
The address that Deliver uses for all communication to HCL Unica hosted services, except uploading recipient lists and related metadata.

**Default value**

em.unicadeliver.com

If you are connecting to the Europe data center, change this value to em-eu.unicadeliver.com.

**dataHostName**

**Description**

The address that Deliver uses for uploading metadata that is related to recipient lists to HCL Unica hosted services.

**Default value**

em.unicadeliver.com

If you are connecting to the Europe data center, change this value to em-eu.unicadeliver.com.

**ftpHostName**

**Description**

The address that Deliver uses for uploading recipient list data (except list metadata) to HCL Unica hosted services.

**Default value**

ftp-em.unicadeliver.com

If you are connecting to Europe datacenter, change this value to ftp-eu.unicadeliver.com.
Deliver | partitions | partition[n] | hostedAccountInfo
Define properties in this category to define user credentials for the database that contains account information that is required to access HCL Unica hosted services. Values that you specify here must be defined as user settings in the Platform.

amUserForAcctCredentials

**Description**

Use this property to specify the Platform user that contains a Platform data source that specifies the account access credentials that are required to access HCL Unica hosted services.

**Default value**

asm_admin

**Valid Values**

Any Platform user.

amDataSourceForAcctCredentials

**Description**

Use this property to specify the Platform data source that defines login credentials for HCL Unica hosted services.

**Default value**

UNICA_HOSTED_SERVICES

**Valid Values**

A data source that is associated with the user you specify in amUserForAcctCredentials
Deliver | partitions | partition[n] | dataSources | systemTables

This category contains configuration properties that define the schema, connection settings, and login credentials for the database that contains the Deliver system tables in your network environment.

**type**

**Description**

Type of database that hosts the Deliver system tables.

**Default value**

No default value defined. You must define this property.

**Valid Values**

- SQLSERVER
- ORACLE
- DB2

**schemaName**

**Description**

Name of the database schema for the Deliver system tables. This is the same as the schema name for the Campaign system tables.

You must include this schema name when referencing system tables in scripts.

**Default value**

dbo

**jdbcBatchSize**

**Description**
The number of execution requests JDBC runs on the database at a time.

**Default value**
10

**Valid Values**
An integer greater than 0.

**jdbcClassName**

**Description**
JDBC driver for system tables as defined in your Campaign web server.

**Default value**
No default value defined. You must define this property.

**jdbcURI**

**Description**
JDBC connection URI for system tables as defined in your Campaign web server.

**Default value**
No default value defined. You must define this property.

**asmUserForDBCredentials**

**Description**
Use this property to specify an HCL Unica user that will be allowed to access the Deliver system tables.

**Default value**
No default value defined. You must define this property.

**Valid Values**
Any user defined in the Platform. This should typically be the name of the system user for Campaign.

**amDataSourceForDBCredentials**

**Description**

Use this property to specify the data source that defines login credentials for the database that contains the Deliver system tables. This can be the same as the data source for the Campaign system tables.

**Default value**

UA_SYSTEM_TABLES

**Valid Values**

A Platform data source associated with the HCL Unica user you specify in *asmUserForDBCredentials*.

The data source specifies a database user and credentials used to access the Deliver system tables. If the default schema for the database user is not the schema that contains the system tables you must specify the system table schema in the JDBC connection used to access the system tables.

**poolAcquireIncrement**

**Description**

When the database connection pool runs out of connections, the number of new connections Deliver creates for the system tables. Deliver creates new connections up to the number specified in *poolMaxSize*.

**Default value**

1

**Valid Values**

An integer greater than 0.
poolIdleTestPeriod

Description
The number of seconds Deliver waits between testing idle connections to the Deliver system tables for activity.

Default value
100

Valid Values
An integer greater than 0.

poolMaxSize

Description
The maximum number of connections Deliver makes to the system tables. A value of zero (0) indicates there is no maximum.

Default value
100

Valid Values
An integer greater than or equal to 0.

poolMinSize

Description
The minimum number of connections Deliver makes to the system tables.

Default value
10

Valid Values
An integer greater than or equal to 0.
poolMaxStatements

**Description**

The maximum number of statements that Deliver stores in the PrepareStatement cache per connection to the system tables. Setting poolMaxStatements to zero (0) disables statement caching.

**Default value**

0

**Valid Values**

An integer equal to or greater than 0.

timeout

**Description**

The number of seconds Deliver maintains an idle database connection before dropping the connection.

If poolIdleTestPeriod is greater than 0, Deliver tests all idle, pooled, but unchecked-out connections, every timeout number of seconds.

If poolIdleTestPeriod is greater than timeout, the idle connections are dropped.

**Default value**

100

**Valid Values**

An integer equal to or greater than 0.

Deliver | partitions | partition[n] | recipientListUploader

This configuration category contains an optional property for the location of a user-defined script that runs in response to the actions or status of the Recipient List Uploader.
**pathToTriggerScript**

**Description**

You can create a script that triggers an action in response to the upload of a recipient list to HCL Unica hosted services. For example, you can create a script to send an email alert to the list designer when the list upload completes successfully.

If you define a value for this property, Deliver passes status information about the Recipient List Uploader to the specified location. Deliver takes no action if you leave this property blank.

**Default value**

No default value defined.

**Valid Values**

Any valid network path.

**Deliver | partitions | partition[n] | responseContactTracker**

Properties in this category specify behavior for the Response and Contact Tracker (RCT). The RCT retrieves and processes data for email contacts, email delivery, and recipient responses, such as link clicks and opens.

**pauseCustomerPremisesTracking**

**Description**

Deliver stores contact and response data in a queue in HCL Unica hosted services. This property allows you to instruct the RCT to temporarily stop retrieving data from HCL Unica hosted services. When you resume tracking, the RCT downloads the accumulated data.

**Default value**

False
Valid Values
True | False

**waitTimeToCheckForDataAvailability**

**Description**
The RCT periodically checks for new data regarding email contacts or recipient responses. This property allows you to specify how often, in seconds, the RCT checks for new data in HCL Unica hosted services. The default value is 300 seconds, or every 5 minutes.

**Default value**
300

**Valid Values**
Any integer greater than 1.

**perfLogInterval**

**Description**
This property allows you to specify how often the RCT logs performance statistics to a log file. The value you enter determines the number of batches between log entries.

**Default value**
10

**Valid Values**
An integer greater than 0.

**enableSeparatePartialResponseDataTracking**

**Description**

...
This property determines if Deliver forwards partial email response data to the tracking tables in your local Deliver installation.

Deliver requires the Mailing Instance ID and Message Sequence Number to properly attribute email responses. When you enable separate partial response data tracking, Deliver places the incomplete responses in separate local tracking tables where you can review them or perform additional processing.

**Default value**

True

**Valid Values**

True | False

**enableExecutionHistoryDataTracking**

**Description**

This property controls whether you can download additional mailing execution history data from HCL Unica.

By default, this property is set to **False**, to prevent download of additional data. When you set this property to **True**, you can download data about mailing runs that is not ordinarily entered to the Deliver system tables. You can use this supplementary information to help automate mailing and database management.

This property is hidden by default. You can display this configuration property in your local Deliver installation by running the `switch_config_visibility.bat` script, located in the Deliver\tools directory.

Access to mailing execution history data is available by request from Unica. To request access to additional mailing execution history data, contact Unica Deliver Services team via HCL technical support.

**Default value**
False

Valid Values

True | False
Chapter 7. Configurations for implementing Push notifications

Introduction

Unica supports push notifications by using the Kumulos SDK, provided in the form of a framework to ease the integration into your iOS, Android or cross platform app. This guide provides an overview of setting up your app with in both your Apple Developer and Firebase Cloud Messaging accounts before detailing the specific integration steps for the appropriate SDK for your project.

The integration is completed by completing the following steps:

1. Configure your app in your Apple Developer account
2. Configure your app in your Firebase Cloud Messaging account
3. Configure your Unica app for APNS and FCM by providing the relevant credentials
4. Select the appropriate SDK for your selected development platform

Configure your Apple developer account

Register your bundle identifier and capabilities

In order to publish your app to the App Store you must define a bundle identifier and configure its capabilities to allow push notifications and an app group.

From your Apple Developer account select "Certificates, Identifiers & Profiles" then select "Identifiers" from the left menu. Click the + icon to register your new identifier.
When prompted to register a new identifier select the App IDs option and click continue, on the second step select the type App and again click continue.

On the final step you must configure your Bundle ID and Capabilities, your bundle ID will usually follow the standard of com.[organisation name].[app name] for example "com.kumulos.myPushApp", select the 'Explicit' radio button and enter your fully qualified bundle ID.

In the Capabilities section ensure that the "App Groups" and "Push Notifications" checkboxes are checked, then press continue.
The final form step will repeat your details for confirmation, if everything is correct then click "Register"

Create an App group

Again in the Identifiers list click the + icon to register a new identifier, select the App Groups radio button and click 'Continue', your App Group identifier must follow the convention group.{your.bundle.identifier}.kumulos, following our example bundle ID this would be "group.com.kumulos.myPushApp.kumulos". Click continue, and on the final screen if all details are correct click "Register"
Link the App Group to your App ID

From the Identifiers list click on your AppID for the appropriate bundle identifier, you can also filter the list to only App ID types using the filter at the top right.

From the "Edit your AppID Configuration" screen click the "Configure" button next to the "App Groups" capability, in the popup window check the box next to the group created in the previous step with the matching bundle ID and then click "Continue", the text next to "App Groups" in the configuration screen should now read "Enabled App Groups (1)", click Save.

Create APNS certificates

In order to send push notifications to iOS devices with Unica, you'll need to create certificates in the Apple Developer Member Center in order to register the credentials with your Unica App.

At the end of this step you will have a passphrase secured .p12 file for adding to the Unica App.

First we will create a certificate signing request, using Keychain Access. Access the Keychain Access menu from your toolbar and select Certificate Assistant, Request a Certificate From a Certificate Authority.

In the dialog window enter your email in the 'User Email Address' field, your name should already appear in the 'Common Name' field. Select the 'Saved to disk' radio button and click 'Continue', save the file to your disk for use later.
Next, access your Apple Developer account and select ‘Certificates, Identifiers & Profiles’ on the Certificates screen click the blue + button at the top, on the ‘Create a New Certificate’ screen scroll to ‘Services’ and select ‘Apple Push Notification service SSL (Sandbox & Production)’, click Continue.
On the next screen select the App ID created during the getting started process and click Continue.

When you are prompted to upload a Certificate Signing Request, click Choose File and find then select the `certSigningRequest` file created earlier, click Continue.
On the final step you will be presented with a screen confirming all the details selected so far, click the Download button.

This will save a .cer file to your local disk, double click on the file to add it to your Keychain Access.

You can filter the results of your keychain by using the text filter at the top right. Once you have found the item matching the Kind 'certificate' with the name 'Apple Push Services [your bundle ID]' right click the item and select 'Export'.
In the new window make sure the selected File Format is 'Personal Information Exchange (.p12)' and choose a location to save the file, when you click 'Save' you should be prompted for a password to protect the exported items, enter one and verify.

You will require the passphrase and .p12 file to configure your Unica App backend.

Configure Firebase Cloud Messaging

In order to send messages to your Android users with Unica you'll need to create a Firebase app and configure it for Firebase Cloud Messaging.

In this step we will generate several artifacts used to configure both the Unica App and your Android app project, these are:

- Google Services Json file
- Google Services Account Json file
- Server key

First sign into your Firebase Console and select 'Add project' and enter your Project name in the wizard, you can opt in or out to Google Analytics for your project, it will not affect the integration.

Finally the firebase console will create your app, once the setup process has finished click Continue.

Next we will add an Android app to the Firebase project, from the project overview screen click the configuration cog next to 'Project Overview' and click 'Project settings'
In the 'General' tab in the 'Your Apps' panel, click the icon for Android, this will open the wizard to create the app.
Enter your app's Package name in the top window, and click 'Register app', this will automatically download the Google Services Json which will be used later during the SDK integration with your android project.

Select the cloud messaging tab and copy the Server key for reference later, this will be used to configure the Unica App.
Next, click the 'Service Accounts' tab and click 'Generate new private key' this will download a JSON file describing your account credentials, this will also be used to configure your Unica app.

You will require both the Service Accounts JSON file and the Server key to configure your Unica App backend.

**Configure your Unica app**

Complete the following steps.

1. Under **Settings** menu, select **Messaging settings**. You can see "Mobile Push Notification Settings" option on following page, otherwise contact the Unica support to activate this option.
2. Click **Display the list of mobile app keys**. The following page displays.

3. Click **Add a mobile application key**. The following page displays.
4. Fill the form with appropriate values.
   • App name: Name of the app
   • Description: Description of the app
   • App key type – It must be "production".
   • App Operating System: Android or iOS. It depends on the users you are targeting.
   • Provider Account: This is fetched from Unica backend according to the accounts created.
   • Default time zone: Can be chosen according to the location.
   • Google FCM File: Provide a Google service account json file for mobile app.

If you choose iOS in ‘App Operating System’ then instead of ‘Google FCM File’ you can see the following.

   • Certificate File: Specify p12 file for iOS app.
   • Certificate Password: Provide the password for p12 certificate.
   • Security Policies: Choose as per the policies to be applied on app.

5. Click **Save** to create the Mobile app. You can see the app on the listing page. The "File Upload Success" provides status of FCM/P12 file upload, which you have provided in the form.

**Integrate the SDK**

Select the appropriate SDK for your selected development platform

**Native**
• iOS Swift (on page 89)
• Android (on page 95)

Cross Platform

• React Native (on page 115)

iOS Swift

Introduction

The Kumulos SDK is an open source project hosted on Github and can be found at https://github.com/Kumulos/KumulosSdkSwift.

This guide assumes you have completed the steps from the Configurations for implementing Push notifications (on page 76) and have configured your Apple Identifier, Capabilities and Provisioning Profile from that guide and covers the following integration steps:

1. Integrate the SDK and configuring your project for APNS capability
2. Initializing the SDK components within your project and register for push notifications
3. Register the Kumulos InstallationID with your backend to provide a link between the device and your users represented in your CRM backend for later targeting of notifications.
4. Send a test push notification from your Unica app and receive on the device.
5. Optional hooks for advanced behavior

Integration

Both Carthage and CocoaPods integration instructions are available in the GitHub repo.
Get started with Cocoapods

Get started with Carthage

Simply follow the instructions for your preferred dependency manager to add the KumulosSDK framework to the project.

Add a notification service extension

In order to support all the features of the Apple Push Notifications Service your app must have a notification service extension in order to allow some limited processing of the notification on receipt, before the operating system presents it to the user.

This is a second project which must be added to your workspace as demonstrated in the video below.

If using CocoaPods, add the following to your Podfile and run `pod install`.

```ruby
target 'KumulosNotificationServiceExtension' do
  pod 'KumulosSdkSwiftExtension', '~> 8.4.0'
end
```

Then replace the contents of `NotificationService.swift` with the following lines:

```swift
import UserNotifications
import KumulosSDKExtension

class NotificationService: UINotificationServiceExtension {
  override func didReceive(_ request: UNNotificationRequest, withContentHandler: @escaping (UNNotificationContent) -> Void) {
    KumulosNotificationService.didReceive(request, withContentHandler)
  }
}
```

The Kumulos SDK helper functions will automatically add picture attachments and buttons to the notification content.
Configure your app capabilities and entitlements

In your app project settings use the "+ capability" button to add the App Groups, Background Modes and Push Notifications capabilities.

In your Notification Extension use the "+ capability" button to add the App Groups capability.

In both projects the App Groups capability should be configured to share the same group, this must exactly match the group defined in your Identifiers capabilities earlier.

`group.{your.bundle.identifier}.kumulos`

In your app project the Background Modes should have the "Remote Notifications" mode checked.

Test your configuration

At this point you can test deploy your app onto a device to ensure that your entitlements and capabilities are configured correctly.

Initialization

To configure the SDK for use you need to initialize it with your app’s credentials, this should be done early in your application startup.
The Kumulos SDK automatically supports badges, buttons and image content, although please note that due to iOS limitations badges are not set when app is in the foreground.

**Registering for Push notifications**

iOS requires explicit user permission to receive notifications, when you consider it appropriate you can trigger the prompt to allow notifications by calling:

```swift
Kumulos.pushRequestDeviceToken()```

This helper will prompt the OS to request the user to accept push notifications with the badge, alert, and sound options.

When the user accepts, the Kumulos SDK will automatically handle registration of the push token with the Kumulos backend.
Registering your CRM

When initialized for the first time, the Kumulos SDK will create a unique identifier for the app installation that initialized the SDK, this identifier can be used later to target push notifications to a specific device.

In order to retrieve this installation ID, simply access the class variable:

```javascript
let installId = Kumulos.installId;
```

Once you have the installation ID, you can send it to your app's CRM backend to be used later for push targeting.

**Optionally associate your app user with Kumulos for targeting**

If your app makes use of an identifier to uniquely tell which user is signed into a device (for example a primary key integer or UUID, or an email address), you can send this identifier to Kumulos for later push targeting via the same key.

```javascript
Kumulos.associateUserWithInstall(userIdentifier: "unique-user-identifier")
```

**Sending a test**

**Advanced features**
Handling notification opened events

When a user interacts with your push message, either by tapping the notification itself, or an included action button, the `pushOpenedHandlerBlock` will be called, in this block you can provide further behavior to handle custom actions.

```swift
let builder = KSConfigBuilder(apiKey: "your-api-key", secretKey: "you
.setPushOpenedHandler(pushOpenedHandlerBlock: { (notification : K
    // Inspect notification data and do work.
    if let action = notification.actionIdentifier {
        print("User pressed an action button.")
        print(action)
        print(notification.data)
    } else {
        print("Just an open event.")
    }
})

Kumulos.initialize(config: builder.build())
```

Handling background data pushes

When you send a push with the `content-available` flag set on the notification this will allow your app to be woken up to process the push notification in the background, triggering any required behavior in your app without launching into the foreground.

If you set a title & message then the notification will be silent and nothing will be shown to the user in the notification center, however you can also provide a title and message in order to trigger behavior and then notify the user.

The `content-available` flag will trigger the `application:didReceiveRemoteNotification:fetchCompletionHandler:` delegate, from here you can inspect the notification payload and perform any actions required.
Android

Introduction

The Kumulos SDK is an open source project hosted on Github and can be found at https://github.com/Kumulos/KumulosSdkAndroid.

This guide assumes you have completed the steps from the Configurations for implementing Push notifications (on page 76) and have configured your Firebase Console and Unica App with the relevant credentials for Cloud Messaging. It will cover the following steps:

1. Integrate the SDK and configuring your project.
2. Initialize the SDK components within your project and register for push notifications.
3. Register the Kumulos InstallationID with your backend to provide a link between the device and your users represented in your CRM backend for later targeting of notifications.
4. Send a test push notification from your Unica app and receiving on the device.
5. Optional hooks for advanced behavior.
Configure your Apple developer account

Register your bundle identifier and capabilities

In order to publish your app to the App Store you must define a bundle identifier and configure its capabilities to allow push notifications and an app group.

From your Apple Developer account select "Certificates, Identifiers & Profiles" then select "Identifiers" from the left menu. Click the + icon to register your new identifier.

When prompted to register a new identifier select the App IDs option and click continue, on the second step select the type App and again click continue.

On the final step you must configure your Bundle ID and Capabilities, your bundle ID will usually follow the standard of com.[organisation name].[app name] for example "com.kumulos.myPushApp", select the 'Explicit' radio button and enter your fully qualified bundle ID.
In the Capabilities section ensure that the "App Groups" and "Push Notifications" checkboxes are checked, then press continue.

The final form step will repeat your details for confirmation, if everything is correct then click "Register"

**Create an App group**

Again in the Identifiers list click the + icon to register a new identifier, select the App Groups radio button and click 'Continue', your App Group identifier must follow the convention group.{your.bundle.identifier}.kumulos, following our example bundle ID this would be "group.com.kumulos.myPushApp.kumulos". Click continue, and on the final screen if all details are correct click "Register"
Link the App Group to your App ID

From the Identifiers list click on your AppID for the appropriate bundle identifier, you can also filter the list to only App ID types using the filter at the top right.

From the "Edit your AppID Configuration" screen click the "Configure" button next to the "App Groups" capability, in the popup window check the box next to the group created in the previous step with the matching bundle ID and then click "Continue", the text next to "App Groups" in the configuration screen should now read "Enabled App Groups (1)", click Save.

Create APNS certificates

In order to send push notifications to iOS devices with Unica, you'll need to create certificates in the Apple Developer Member Center in order to register the credentials with your Unica App.

At the end of this step you will have a passphrase secured .p12 file for adding to the Unica App.

First we will create a certificate signing request, using Keychain Access. Access the Keychain Access menu from your toolbar and select Certificate Assistant, Request a Certificate From a Certificate Authority.
In the dialog window enter your email in the 'User Email Address' field, your name should already appear in the 'Common Name' field. Select the 'Saved to disk' radio button and click 'Continue', save the file to your disk for use later.

Next, access your Apple Developer account and select 'Certificates, Identifiers & Profiles' on the Certificates screen click the blue + button at the top, on the 'Create a New Certificate' screen scroll to 'Services' and select 'Apple Push Notification service SSL (Sandbox & Production)', click Continue.
On the next screen select the App ID created during the getting started process and click Continue.

When you are prompted to upload a Certificate Signing Request, click Choose File and find then select the `certSigningRequest` file created earlier, click Continue.
On the final step you will be presented with a screen confirming all the details selected so far, click the Download button.

This will save a `.cer` file to your local disk, double click on the file to add it to your Keychain Access.

You can filter the results of your keychain by using the text filter at the top right. Once you have found the item matching the Kind ‘certificate’ with the name ‘Apple Push Services [your bundle ID]’ right click the item and select ‘Export’.
In the new window make sure the selected File Format is ‘Personal Information Exchange (.p12)’ and choose a location to save the file, when you click ‘Save’ you should be prompted for a password to protect the exported items, enter one and verify.

You will require the passphrase and .p12 file to configure your Unica App backend.

Integration

Add the Firebase components to your app as shown below.

In the root build.gradle, ensure the Google repository is enabled and the google-services plugin is on the classpath:

```groovy
buildscript {
    // ...
    dependencies {
        // ...
        classpath 'com.google.gms:google-services:4.2.0' // google-se
    }
}

allprojects {
    // ...
    repositories {
        google() // Google's Maven repository
        // ...
    }
}
```

The Kumulos libraries are distributed through JCenter. To install the libraries, edit your app's build.gradle file to add the following:

- Exclude conflicting meta data files from the build
- Declare source & target compile options
- Add the Kumulos library dependencies
- Add the Firebase core SDK
- Apply the google-services plugin

A sample `build.gradle` can be seen below.

```java
android {
  // Exclude duplicate files from the build
  packagingOptions {
    exclude 'META-INF/NOTICE'
    exclude 'META-INF/ASL.0'
    exclude 'META-INF/LICENSE'
  }

  compileOptions {
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
  }

  apply plugin: 'com.android.application'

  dependencies {
    // Kumulos debug & release libraries
    debugImplementation 'com.kumulos.android:kumulos-android-debug:11'
    releaseImplementation 'com.kumulos.android:kumulos-android-release:11'
    implementation 'com.google.firebase:firebase-core:16.0.7'
  }

  // ADD THIS AT THE BOTTOM
  apply plugin: 'com.google.gms.google-services'
}
```

The `debugImplementation` will have logging enabled with a tag matching `com.kumulos.*`. When running in debug mode, the log messages should be visible in LogCat.

Run a Gradle sync to install the Kumulos libraries and build your project.

By default, the Firebase SDK will send analytics data to Google. To disable this, simply add

```xml
<meta-data android:name="firebase_analytics_collection_deactivated"
    android:value="true" />
```

to your app's `AndroidManifest.xml`
Download the `google-services.json` file from your Firebase app 'General' settings, and add it to your `app/` folder.

Now you can add the Kumulos `FirebaseMessagingService` and `PushBroadcastReceiver` to your `AndroidManifest.xml`.

```xml
<manifest ...
<application ...
<service ...
</service>
<receiver ...
</receiver>
</manifest>
```

**Initialization and registering for push notifications**

To initialize the SDK, we recommend subclassing the `Application` class and initializing Kumulos in its `onCreate` method.
If you want to unregister the installation, you can use `Kumulos.pushUnregister(context)`.

Registering with your CRM

**Installation ID**

When initialized for the first time, the Kumulos SDK will create a unique identifier for the app installation that initialized the SDK, this identifier can be used later to target push notifications to a specific device.

In order to retrieve this installation ID, simply access the class variable:
Once you have the installation ID, you can send it to your app's CRM backend to be used later for push targeting.

### Optionally associate your app user with Kumulos for targeting

If your app makes use of an identifier to uniquely tell which user is signed into a device (for example a primary key integer or UUID, or an email address), you can send this identifier to Kumulos for later push targeting via the same key.

```java
Kumulos.associateUserWithInstall(context, "unique-user-identifier");
```

### Advanced features

#### Handling Push action buttons

Push messages allow you to hand-off to native application screens via deep-linking push action buttons. When tapped, these buttons pass control to the defined push action handler.

If you want to handle deep-links as part of a Push message you can create a class that implements the `PushActionHandlerInterface` and assign it during SDK initialization.

```java
Kumulos.setPushActionHandler(new MyPushActionHandler());
```

A stub implementation of the handler could be as follows:
Default Push Behavior

By default, the Kumulos PushBroadcastReceiver will show a notification on the notification area of the device when a content push is received.

Tapping this notification will open the main launcher activity of your application and track the push open conversion for you.

Your main activity will receive the push contents in its options bundle under the PushMessage.EXTRA_KEY.

Changing the Push Icon

To change the icon shown in the status bar on Android, you can configure Kumulos with a drawable at initialization time:

```java
public class MyPushActionHandler implements PushActionHandlerInterface{
    public void handle(Context context, PushMessage pushMessage, String actionId, String note) {
        // actionId is the button id you set when creating the notification.
        // Note, that when action button is clicked your app's activity is launched.
    }
}
```

Ensure that you comply with the status bar icon guidelines so the icon renders correctly on all devices. For help preparing assets, we suggest checking out the Android Asset Studio.
Customizing Push Behavior

To customize the behavior of the SDK when a push is received or its notification is tapped, we suggest subclassing the `PushBroadcastReceiver` and overriding its base methods depending on what you want to customize.

Example extension class:

```java
package com.example;

import com.kumulos.android.PushBroadcastReceiver;

public class MyPushReceiver extends PushBroadcastReceiver {
}
```

Make sure to change the `AndroidManifest.xml` receiver:

```xml
<receiver android:name="com.example.MyPushReceiver" android:exported="true">
    <intent-filter>
        <action android:name="com.kumulos.push.RECEIVED" />
        <action android:name="com.kumulos.push.OPENED" />
        <action android:name="com.kumulos.push.DISMISSED" />
        <action android:name="com.kumulos.push.BUTTON_CLICKED" />
    </intent-filter>
</receiver>
```

Changing the Launched Activity

To change which activity will be launched when the user taps a notification, you can override the `PushBroadcastReceiver#getPushOpenActivityIntent(Context,`
The `PushMessage` model will not be added to the `Intent` by default, it is up to you to add it as an extra if desired:

```java
Intent launchIntent = new Intent(context, MyActivity.class);
launchIntent.putExtra(PushMessage.EXTRA_KEY, pushMessage);
```

You can return `null` to track the push conversion and do nothing when the notification is tapped.

If the `Intent` returned does not describe an `Activity`, it will be ignored.
Customizing the notification

To customize the notification shown to the user for content pushes, you can override `PushBroadcastReceiver#buildNotification(Context, PushMessage)`.

```java
package com.example;

import android.app.Notification;
import android.content.Context;

import com.kumulos.android.PushBroadcastReceiver;
import com.kumulos.android.PushMessage;

public class MyPushReceiver extends PushBroadcastReceiver {

    @Override
    protected Notification buildNotification(Context context, PushMessage pushMessage) {
        // TODO customize the notification
        return super.buildNotification(context, pushMessage);
    }
}
```

If you want to handle the open/dismissal with the broadcast receiver, be sure to set up the content intents of the notification as follows:
This will ensure that the notification conversion is tracked in Kumulos.

If you want to do something else, you can manually track push open conversion using `Kumulos#pushTrackOpen(Context, int)` and track dismissed event using `Kumulos#pushTrackDismissed(Context, int)`. Additionally, you would have to add deep link extras for in-app message deep links to keep working.

**Launching a Service for Background Data Pushes**

To launch a service when a background data push is received, you can override `PushBroadcastReceiver#getBackgroundPushServiceIntent`. 
This would easily allow you to handle data processing in the background by launching an `IntentService` for example.

The `PushMessage` model will not be added to the `Intent` by default, it is up to you to add it as an extra if desired:

```java
Intent serviceIntent = new Intent(context, MyIntentService.class);
serviceIntent.putExtra(PushMessage.EXTRA_KEY, pushMessage);
```

Return `null` if you want to do nothing with the data push.

If the `Intent` returned does not describe a `Service`, it will be ignored.

### URL Pushes

Push notifications sent to open a URL will, by default, track the push open if the user taps the notification, and then open the default web browser.
Overriding All Behaviors

If you want to completely replace the logic for handling pushes, you can override PushBroadcastReceiver#onPushReceived(Context, PushMessage).

Bear in mind you will be responsible for all aspects of the push process such as showing a notification to the user, tracking an open conversion using Kumulos#pushTrackOpen(Context, int) and dismissed events using Kumulos#pushTrackDismissed(Context, int), or launching any activities or services.

In addition, you may need to implement behaviors for:

- Delivery tracking: pushTrackDelivered(context, pushMessage)

Using Your Own FirebaseMessagingService with Kumulos

If you already consume FCM push notifications with your own FirebaseMessagingService but you want to also enjoy the benefits of the Kumulos push service, you can use the SDK's helper methods in your own implementation. For example:
If you are using ProGuard to optimize your Java code, then you need to ensure that your proguard.cfg file includes the Kumulos SDK and required components, something like:
Proguard is also very sensitive to UTF-8 with BOM encoding, whereas Android tooling will only accept UTF-8. Therefore, an easy way to ensure you don't inadvertently have UTF-8 byte-order-marks in your proguard.cfg is to use vim via the terminal as follows:

```
$ vim proguard.cfg
:set nobomb
:wq!
```

### React Native

**Introduction**

The Kumulos SDK is an open source project hosted on Github and can be found at [https://github.com/Kumulos/KumulosSdkReactNative](https://github.com/Kumulos/KumulosSdkReactNative).
This guide assumes you have completed the steps from the introduction and have configured your project via the Apple Developer Account and the Firebase Console, it will cover the following steps:

1. Integrate the SDK and configuring your projects for APNS/FCM capability.
2. Initializing the SDK components within your project and registering for push notifications.
3. Registering the Kumulos InstallationID with your backend to provide a link between the device and your users represented in your CRM backend for later targeting of notifications.
4. Sending a test push notification from your Unica app and receiving it on the device.
5. Optional hooks for advanced behavior.

Integration

The Kumulos React Native module requires native features so should be installed in an ejected project.

To install and link the project, run the following commands:

```bash
npm install kumulos-react-native --save
don install --project-directory=ios
```

linking steps are required for each platform.

Android linking steps

To complete the linking process for Android, you need to ensure your project uses the following versions for tools and libraries:

- Gradle plugin v3.1.3 or greater
- Build tools v23.0.3 or greater
- Support library v27.+
Place the `google-services.json` created during the introduction in your project’s `android/app` directory. In addition, you must add the following to your `android/app/build.gradle` file:

```gradle
android {
    // ...
    packagingOptions {
        exclude 'META-INF/NOTICE'
        exclude 'META-INF/ASL2.0'
        exclude 'META-INF/LICENSE'
    }

    dependencies {
        // Kumulos debug & release libraries
        classpath 'com.google.gms:google-services:4.2.0'
    }

    // ADD THIS AT THE BOTTOM
    apply plugin: 'com.google.gms.google-services'
}
```

**iOS Project Configuration**

The notification will expand upon swiping the notification on devices supporting 3D Touch. In order to enable this functionality you need open your iOS project in Xcode and add a Notification Service Extension to your application.

**Add a Notification Service extension**

In order to support all the features of the Apple Push Notifications Service your app must have a notification service extension in order to allow some limited processing of the notification on receipt, before the operating system presents it to the user.
This is a second build target added to your existing xcode project by going to your project info screen and clicking the '+' button at the footer.

From the popup window, select the

Notification Service Extension template for your new project and click 'Next'.

On the final window add an appropriate name for your extension and click 'Finish'.
Add the following to the Podfile generated by React Native and run `pod install`.

```ruby
  target 'KumulosNotificationServiceExtension' do
    pod 'KumulosSdKObjectiveCExtension', '4.2.2'
  end
```

The template for the project will have created a file named `NotificationService.m` automatically, replace its content with the following lines:
The Kumulos SDK helper functions will automatically add picture attachments and buttons to the notification content.

### Configure your app capabilities and entitlements

In your app project settings use the "+ capability" button to add the App Groups, Background Modes and Push Notifications capabilities.

In your Notification Extension use the "+ capability" button to add the App Groups capability.

In both projects the App Groups capability should be configured to share the same group, this must exactly match the group defined in your Identifiers capabilities earlier.

```
group.{your.bundle.identifier}.kumulos
```

In your app project the Background Modes should have the "Remote Notifications" mode checked.

### Test your configuration

At this point you can test deploy your app onto a device to ensure that your entitlements and capabilities are configured correctly. Make sure to correctly set up signing for the extension target.
Initialization

To configure the SDK for use you need to initialize it with your app's API credentials. This must be done early in your application startup so that you can start calling API methods and using features.

In your **App.js**:

```javascript
import Kumulos from 'kumulos-react-native';

Kumulos.initialize({
    apiKey: 'YOUR_API_KEY',
    secretKey: 'YOUR_SECRET_KEY'
});

// When you are ready to request the push token from the user, you wo
Kumulos.pushRequestToken();
```

In your **ios/AppDelegate.m** `application:didFinishLaunchingWithOptions:` method:

```objective-c
#import <KumulosReactNative/KumulosReactNative.h>
...

KSConfig* kumulosConfig = [KSConfig
    configWithAPIKey:@"YOUR_API_KEY"
    andSecretKey:@"YOUR_SECRET_KEY"];

[KumulosReactNative initializeWithConfig:kumulosConfig];
```

In your **android/app/src/main/java/.../MainApplication.java** `onCreate` method:
Registering with your CRM

When initialized for the first time, the Kumulos SDK will create a unique identifier for the app installation that initialized the SDK, this identifier can be used later to target push notifications to a specific device.

In order to retrieve this installation ID, simply access the class variable:

```
const id = await Kumulos.getInstallId();
```

Once you have the installation ID, you can send it to your app's CRM backend to be used later for push targeting.

User association

Optionally associate your app user with Kumulos for targeting

If your app makes use of an identifier to uniquely tell which user is signed into a device (for example a primary key integer or UUID, or an email address), you can send this identifier to Kumulos for later push targeting via the same key.

```
Kumulos.associateUserWithInstall('unique-user-identifier');
```
Advanced features

Handling open events or background data pushes

The following sample code shows how to use Kumulos to handle push notifications for deep-linking and other common messaging tasks.

```javascript
import Kumulos from 'kumulos-react-native';

Kumulos.initialize({
    apiKey: 'YOUR_API_KEY',
    secretKey: 'YOUR_SECRET_KEY',
    pushReceivedHandler: (notification) => {
        // Called when a push is received with your app in the foreground
    },
    pushOpenedHandler: (notification) => {
        // Called when a user taps on a push notification
    }
});
```

Handling notification action buttons

When a user interacts with your push message the `pushOpenedHandler` defined as above is called. If a button was tapped, the notification object will contain an `actionId` property:

```javascript
Kumulos.initialize({
```
Kumulos.initialize({
  ...,
  pushOpenedHandler: (notification) => {
    console.log(notification.actionId);
  },
});
Chapter 8. Utilities for Deliver

Deliver provides several scripts that you use to administer Deliver functions.

You can use the software utilities described in this section for a variety of startup and administration functions. In addition to software utilities used with Unica Platform, Unica Deliver uses utilities are specific to Deliver and you use them only to manage Deliver components.

For more information about other utilities available for your HCL Unica installation, see the Unica Platform Administrator's Guide.

The RLU script

Use the RLU script to check the status of the Recipient List Uploader (RLU).

Little Note: You cannot use this script to start or stop the RLU. Use this script to check connectivity between on premise and on demand components.

RLU script is located at `<Deliver Install Home>/bin` folder. The Deliver directory is a subdirectory in the Campaign directory.

In UNIX or Linux environments, run the script as `rlu.sh`.

In Windows, run the script from the command prompt as `rlu.bat`.

**Syntax**

`rlu -c | --check [-h]`

**Commands**

- `-c, --check`

Check that the RLU is correctly configured, and that is it connected to HCL Unica.
Options

-h, --help

Display syntax for the script

Example

In a Linux environment, determine whether the RLU is connected to HCL Unica hosted services.

rlu.sh --check

Depending on the status of your system, the output of this command might look like this sample.

```
Configuring Data Source [systemTables]...
Testing configuration for partition partition1
Testing connectivity for partition partition1
Testing user accessibility for partition partition1
Succeeded. List uploader config and connectivity test succeeded for partition partition1
```

Deliver Response and Contact Tracker (RCT) script

Use this script to run and check the status of the Response and Contact Tracker (RCT).

This script is located in the `bin` directory under your Deliver installation. The Deliver directory is a sub-directory in the Campaign directory.

In UNIX or Linux environments run the script as `rct.sh`.

In Windows, run the script from the command line as `rct.bat`.

Syntax

```
rct [ start | stop | check ]
```
**Commands**

*start*
Start the RCT

*stop*
Stop the RCT

**Options**

*check*
Check the status of the connection between the RCT and HCL Unica hosted services.

**Examples**

- To start the RCT on Windows.
  
  `rct.bat start`

- To stop the RCT on Windows.
  
  `rct.bat stop`

- In a Linux environment, to determine if the RCT is connected to HCL Unica hosted services.
  
  `rct.sh check`

Depending on the status of your system, the output of this command might look like this:

```
C:\<UNICA_HOME>\Campaign\Deliver\bin>rct check
Testing config and connectivity for partition partition1
Succeeded | Partition: partition1 - Hosted Services Account ID: asm_admin
```
The MKService_rct script

The MKService_rct script adds or removes the Response and Contact Tracker (RCT) as a service. Adding the RCT as a service restarts the RCT every time you restart the computer where you have installed the RCT. Removing the RCT as a service prevents the RCT from restarting automatically.

This script is located in the `bin` directory under your Deliver installation.

In UNIX or Linux environments run `MKService_rct.sh` with a user that has root permissions or permissions to create daemon processes.

In Windows, run the script from the command line as `MKService_rct.bat`.

Syntax

```
MKService_rct -install
MKService_rct -remove
```

Commands

- **-install**
  Add the RCT as a service

- **-remove**
  Remove the RCT service

Examples

- To add the RCT as a Windows service.
  ```
  MKService_rct.bat -install
  ```
- To remove the RCT service on UNIX or Linux.
  ```
  MKService_rct.sh -remove
  ```
configTool

The properties and values on the **Configuration** page are stored in the Platform system tables. You can use the `configTool` utility to import and export configuration settings to and from the system tables. For more details, see the Platform Administrator Guide.
Chapter 9. About troubleshooting Deliver

Unica Deliver provides various tools and techniques that you can use to investigate issues related to your Campaign and Deliver installations.

Log files for Deliver

HCL Unica provides several log files that you can review to monitor your Deliver installation and investigate issues.

**Deliver log file**

This log contains the following types of information regarding information downloaded from HCL Unica hosted services. Located in the logs directory under your Deliver installation.

- general mailing information
- mailing instance ID
- link click data
- data for bounced email

**Deliver temporary files**

This directory contains the data being uploaded. Located in the temp directory under your Deliver installation.

**Campaign log files**

You can review log files in the following locations for information related to mailing-related activity in Campaign.

- Campaign\partitions\<partitionN>\logs
  
  Various log files relating to flowchart runs, including log entries from any Deliver process contained in the flowchart.
• Campaign\logs

This directory contains campaignweb.log that contains information about upload activity performed by the Recipient List Uploader.

Using log4j with Deliver

Deliver uses the Apache log4j utility for logging configuration, debugging, and error information related to the Response and Contact Tracker (RCT) and the Recipient List Uploader (RLU).

For information about changing the system log settings, see:

• The comments in the log4j.xml file.
• The log4j documentation on the Apache web site: https://logging.apache.org/log4j/2.x/manual/index.html

Using log4j with the Recipient List Uploader

When you run the Recipient List Uploader (RLU) utility from the command line, it uses default logger settings.

To change those settings, you modify the deliver_rlu_log4j.xml file.

Modify deliver_rlu_log4j.xml as instructed by the comments in that file. You must not modify this file unless suggested by HCL support.

When the RLU is invoked automatically by a flowchart, it uses the Campaign web application's logging, which is configured in campaign_log4j.xml under your Campaign installation directory.

Using log4j with the Response and Contact Tracker

When you run the Response and Contact Tracker (RCT) utility, it uses default logger settings.

To change those settings, you modify the deliver_rct_log4j.xml file.

Modify deliver_rct_log4j.xml as instructed by the comments in that file.
Chapter 10. Management of user access to messaging features

Campaign and Deliver use roles and permissions provided by the Unica Platform to control user access to messaging features in Deliver and Campaign. You must have permissions in Unica Platform and Campaign to make the required changes. You must also be familiar with how to configure roles and permissions in the Platform and how to define security policies for Campaign.

To conduct email marketing campaigns, email marketers access Deliver mailing features in Unica Campaign.

To create personalized communications and hosted landing pages, marketers work with features and content in the Deliver Document Composer.

For general information about how to configure roles, permissions, and policies, see the sections of the Unica Platform Administrator's Guide that describe how to manage security in the Unica Platform and Unica Campaign.

Role and policy assignment for mailing access

To log in to the HCL Unica system, email marketers enter a system user name and password. The permissions that are granted to the system user determine how the marketer can access mailing features, personalized communications, and content in Deliver and Campaign.

Permissions are associated with roles that are defined in the Unica Platform. To control access to mailing features in Campaign, you can define roles within one or more security policies. All system users that access mailing features, communications, and content must be assigned an Deliver role within a Campaign security policy. Through the policy, you selectively apply permissions for mailing features in Campaign and communications and content in the Deliver Document Composer.

Users that access mailing features must also be assigned the Deliver user and admin roles. These roles are separate from the Deliver roles available in Campaign security policies.
Roles and permissions in Platform and Campaign

Roles in Platform and Campaign are a configurable collection of permissions. For each role in Platform and Campaign, you can specify permissions that control access to the application.

You can use the default roles or create new roles. The set of available permissions is defined by the system; you cannot create a new permission.

About role assignment

Generally, you should give users roles with permissions that reflect the functions that users perform in your organization when they use HCL Unica. You can assign roles to a group or to an individual user. The advantage of assigning roles by group is that you can assign a combination of roles to the group, and if you later want to change that combination, you can do it in one place rather than having to do it multiple times for multiple users. When you assign roles by group, you add and remove users from your groups to control user access.

How the system evaluates roles

If a user has multiple roles, the system evaluates permissions from all those roles together. The ability to perform a function on a particular object is then granted or denied based on the aggregated permissions from all roles. In the case of Campaign, the ability to perform a function on a particular object is granted or denied based on the security policy of the object.

How security policies work

Security policies are the "rule books" that govern security for folders and objects in Campaign. They are consulted each time a user performs an action in the application.

You can create your own security policies or use the default global security policy included with Campaign.
In Campaign, security policies are assigned to folders. When you create a top-level folder, you are required to apply a security policy to the folder. Any objects or subfolders within that folder inherit the folder's security policy.

Because the top-level folder determines the security policy of the objects in the folder, you cannot directly assign a security policy to objects. To change the security policy of an object, you must move the object into a folder with the desired security policy or into the top-level root folder.

You also cannot directly assign a security policy to a user. Unlike objects and folders, which are assigned to security policies as a whole, users are assigned to roles within security policies. To control what users can do, you assign users to roles within security policies. In this way, you control user access to objects within folders that use those security policies.

If a user is not explicitly assigned to at least one role in a security policy, that user cannot create folders and objects under a top-level folder that uses that policy, and that user has no access to objects under that folder or its sub folders.

The following diagram illustrates the relationship between security policies, folders, objects, roles, and users.

![Diagram showing the relationship between security policies, folders, objects, roles, and users.]

**Top-level Administrative roles**

Administrative roles in Unica Campaign are assigned for each partition. Users with these roles can perform the allowed actions on any objects within the partition, regardless of the security policy used in the folders that contain the objects.
Security policies and partitions

Security policies are created per partition. There is no sharing of security policies across partitions.

Each partition in Unica Campaign can have multiple security policies.

Security policy changes when folders and objects are moved or copied.

Objects and folders can be moved or copied across security policies, but the user performing the move or copy must have permissions to do so, in both the source and destination policies.

After an object or folder is moved or copied to a folder that is assigned to a different security policy from its source, the security policy of the lower-level objects or subfolders is automatically changed to the security policy of the new folder.

The global security policy

Campaign includes a default global security policy. You cannot delete this policy; it always applies. However, you can customize your security scheme as follows.

- Modify the roles and permissions in the global policy to suit the needs of your organization.
- Create custom policies and assign users only to your custom policies rather than the global policy.
- Use both custom policies and the global policy.

Any custom policy you create exists under the global policy. If you choose not to create your own security policies, the global security policy is applied by default to the folders and objects that users create in Campaign.

The global security policy contains six pre-defined roles. You cannot delete the pre-defined roles, but you can modify their permissions.

The pre-defined roles in the global security policy are:
- **Folder Owner** - All permissions enabled for the folders a user has created. All users have this role; you do not need to assign users to it.
- **Owner** - All permissions enabled for the objects a user has created. All users have this role; you do not need to assign users to it.
- **Admin** - All permissions enabled. The default user `asm_admin` has this role.
- **Execute** - All permissions enabled.
- **Design** - Read and write permissions on all objects. This role cannot schedule flowcharts or sessions.
- **Review** - Read-only permissions.

**Deliver roles in the Global Policy**

In addition to the pre-defined Campaign roles, the Global Policy includes several roles that are specific to Deliver.

The Global Policy includes the following Deliver roles.

- **Deliver_admin** - Able to access to all mailing features, all content, and all documents.
- **Deliver_execute** - Able to access to all mailing features, all content, and all documents.
- **Deliver_design** - Able to access to all content, all documents, and most mailing features. However, it is explicitly not granted permission to send production mailings.
- **Deliver_review** - Able only to view content and documents, and has limited permissions for working with mailings. It is explicitly not granted permission to add, edit, or delete mailings. It is allowed to view and send test and production mailings.

**Note:** Deliver does not support the Owner and Folder Owner roles that are created by default for Campaign.

**Messaging permissions in Campaign**

Campaign controls user access to mailing features by enabling or disabling specific permissions that are defined in roles that are assigned to a user or group. These roles are associated with one or more security policies. You can define multiple Campaign security
policies and assign multiple roles to each policy. Each combination of policy and roles can define a specific set of permissions.

For more information about how to manage security permissions, including sample security scenarios, see the Unica Campaign Administrator’s Guide.

Under Roles and Permissions for the Unica Platform, you assign user permissions for mailing features and content in the Campaign section, as follows.

1. Define user roles.
   - System defined user roles for Deliver are created by default under the Global Policy.
   - You can also define custom roles and add them to the Global Policy or to other policies that you define.
2. Define security policies and add user roles to the policies.
   - The Global Policy is defined by default. You can define additional policies for Campaign.
3. Define specific permissions for each role in each policy.
   - You can define additional policies and custom roles with various sets of permissions for more control over access to mailing features in Campaign and the Deliver Document Composer.

Changes to permissions, roles, and policies are applied when the user logs in to HCL Unica. After you assign or change mailing permissions for a user, the user must log out and then log back in for the changes to be observed.

**Making roles and permissions available**

Depending on your Unica Platform installation, the administrative controls that are required to define and apply roles and permissions might not be immediately visible. You can make the necessary controls visible by accessing the Deliver Document Composer or a mailing in Campaign.

Perform the following procedure if you do not see all of the following permissions under the Campaign Global Policy.
• Permissions for mailings in the Campaigns category
• Permissions for the Content Library in the Digital Assets category
• Permissions for Deliver documents in the Documents category

1. Log in to HCL Unica.
   
   If you have multiple users that are configured, log in as a user with the most limited permissions. For example, log in as a user with only View permissions.

2. Navigate to **Campaign > Deliver Documents** to access the Document Composer.
   
   Wait for the Document Composer to finish loading.

3. Navigate to **Settings > User Roles and Permissions > Campaign > partition [n] > Global Policy**
   
   When prompted, confirm that you want to leave the Document Composer by leaving the page.

4. Click **Add Roles and Assign Permissions**. The following Deliver roles are visible.
   
   • deliver_admin
   • deliver_execute
   • deliver_design
   • deliver_review

5. Click **Save and Edit Permissions**.
   
   Mailing permissions are visible in the Campaigns, Digital Assets, and Documents categories.
   
   For more information about the specific permissions that are available, see the following topics.

### How Campaign evaluates permissions

When a user performs a task or tries to access an object, Campaign performs the following steps.
1. Identifies all groups and roles to which this user belongs within the global security policy.

Users can belong to one, many, or no roles. Users belong to the Owner role if they own an object; they belong to the Folder Owner role if they own the folder in which an object resides.

Users belong to other roles only if they have been specifically assigned to that role (either directly or because they belong in a group assigned to that role).

2. Identifies whether the object being accessed is assigned to a custom-defined policy. If so, the system identifies all groups and roles to which the user belongs within this custom policy.

3. Aggregates the permissions for all roles to which the user belongs, based on results from steps 1 and 2. Using this composite role, the system evaluates the permissions for the action are evaluated as follows:
   a. If any roles have **Denied** permission for this action, then the permissions are aggregated as follows:
      i. Consider a Global Policy, 1 Custom Policy, and a permission DENIED for the Custom Policy role. Then, DENIAL of any permission for a Custom policy Role takes precedence over permissions assigned to the Global Policy Role.
      ii. Consider a Global Policy, 2 or more Custom Policies, a permission DENIED for one of the Custom policy roles, and the same permission GRANTED to the other Custom policy role. Then, GRANT of any permission of Custom policy takes precedence over DENIAL of permission of the Custom policy.
   b. If no roles have **Denied** permission for this action, then it checks to determine whether any roles have **Granted** permission for this action. If so, the user is allowed to perform the action.
   c. If neither a nor b is true, the user is denied the permission.

**Example for one custom policy**

Consider one custom policy under Global Policy : CustomPolicyA. CustomPolicyA has CustomPolicyARole, that has Add/Edit Campaign permission DENIED.
Consider UserA who has CustomPolicyARole assigned. DENIAL of Add/Edit Campaign permission for a CustomPolicyARole takes precedence over permissions assigned to the Global Policy Role. Hence, the Add/Edit Campaign objects are not visible to UserA.

Example for two custom policies

Consider two custom policies under Global Policy: CustomPolicyA and CustomPolicyB. Both CustomPolicyA and CustomPolicyB have CustomPolicyARole and CustomPolicyBRole respectively. CustomPolicyARole has Add/Edit Campaign permission GRANTED. CustomPolicyBRole has Add/Edit Campaign permission DENIED.

UserA has both CustomPolicyARole and CustomPolicyBRole assigned. GRANT of Add/Edit permission of CustomPolicyARole takes precedence over DENIAL of permission of the CustomPolicyBRole. Hence, the Add/Edit Campaign objects are visible to UserA.

Definitions of permission states

For each role, you can specify which permissions are granted, not granted, or denied. You set these permissions on the Settings > User roles and permissions page.

These states have the following meanings.

- **Granted** - indicated with a check mark ⧫. Explicitly grants permission to perform this particular function as long as none of the user’s other roles explicitly denies permission.

- **Denied** - indicated with an "X" ❌. Explicitly denies permission to perform this particular function, regardless of any other of the user’s roles which might grant permission.

- **Not granted** - indicated with a circle〇. Does not explicitly grant nor deny permission to perform a particular function. If this permission is not explicitly granted by any of a user’s roles, the user is not allowed to perform this function.
Permissions for mailings in Campaign

In Campaign, you create, configure, run, and monitor Deliver mailings with controls on Deliver mailing tabs. You manage each mailing on a separate tab.

The following permissions control user access to the Deliver mailing tabs. They are in the Campaigns category.

<table>
<thead>
<tr>
<th>Permission</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Mailings</td>
<td>Allows a user to view an Deliver mailing tab in a campaign. The user cannot edit or change the mailing.</td>
</tr>
<tr>
<td>Edit Mailings</td>
<td>Allows a user to configure or change an Deliver mailing tab in a campaign.</td>
</tr>
<tr>
<td>Delete Mailings</td>
<td>Allows a user to remove an Deliver mailing from a campaign.</td>
</tr>
<tr>
<td>Add Mailings</td>
<td>Allows a user to create a mailing in a campaign.</td>
</tr>
<tr>
<td>Send production mailing</td>
<td>Allows a user to initiate a production run of the mailing, enable a mailing for transactional email, or schedule a production mailing run.</td>
</tr>
<tr>
<td></td>
<td>Production mailings can include many messages. Email messages are sent to every individual identified as a production recipient on the recipient list that is associated with the mailing.</td>
</tr>
<tr>
<td>Execute test run</td>
<td>Allows a user to initiate a test run of the mailing.</td>
</tr>
<tr>
<td></td>
<td>Test mailings usually involve a few messages. During a test run, an email message is sent to every address identified as a test recipient on the recipient list that is associated with the mailing.</td>
</tr>
</tbody>
</table>

Permissions for the Digital Assets category

The Digital Assets permissions control user access to content elements in the Deliver Content Library, and to folders and subfolders in which they are stored.
The Content Library is a repository for content elements (also called digital assets) that are used in communications that users create in the Deliver Document Composer.

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Deliver digital assets</td>
<td>Allows a user to open content elements to view properties and preview the content that can be added to a personalized communication.</td>
</tr>
<tr>
<td>Create new digital assets in the Deliver content library</td>
<td>Allows a user to create a content element and add it to the Content Library.</td>
</tr>
<tr>
<td>Edit existing digital assets in the Deliver content library</td>
<td>Allows a user to open and edit existing content elements.</td>
</tr>
<tr>
<td>Delete digital assets from the Deliver content library</td>
<td>Allows a user to remove a content element from the Content Library.</td>
</tr>
<tr>
<td>Move digital assets from one folder to another</td>
<td>Allows a user to move content elements within the Content Library. Moving a content element requires assigning this permission to the source and destination folders.</td>
</tr>
</tbody>
</table>

Permissions for the Documents category

The permissions in the Documents category control user access to create, edit, and manage personalized communications in the Deliver Document Composer.

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Deliver documents</td>
<td>Allows a user to view a document that is used to create an email, Inbox push notification, or hosted landing page.</td>
</tr>
<tr>
<td>Create new Deliver documents</td>
<td>Allows a user to create a new personalized communication.</td>
</tr>
<tr>
<td>Permissions</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Edit existing Deliver documents</td>
<td>Allows a user to change an existing personalized communication.</td>
</tr>
<tr>
<td>Delete Deliver documents</td>
<td>Allows a user to remove a personalized communication.</td>
</tr>
<tr>
<td>Publish Deliver document, making content available on the public Internet</td>
<td>Allows a user to publish a personalized communication. Publishing a communication makes the document and all added content available for use in an Deliver mailing.</td>
</tr>
<tr>
<td>Copy Deliver documents from one folder to another</td>
<td>Allows a user to copy a personalized communication between folders in the Content Library. Copying a communication requires assigning this permission to the source and destination folders.</td>
</tr>
<tr>
<td>Move Deliver documents from one folder to another</td>
<td>Allows a user to move a personalized communication from one folder to another folder in the Content Library. Moving a communication requires assigning this permission to the source and destination folders.</td>
</tr>
</tbody>
</table>

Permissions for the Email Administration category

The permissions in the Email Administration category of the Campaign Global Policy provide Deliver administrators with access to settings that control user access to various messaging domains and messaging features.

Administrators assign domain and feature access in the Policy Settings section of the Deliver Settings window. For example, the administrator can restrict the list of email domains that a user can select as a From: domain in an email communication that is created in the Communication Editor. The Policy Settings section does not display unless appropriate permission is explicitly granted to the administrator in the Campaign Global Policy.
Administrators can also control access to administrative interfaces for registering mobile apps with Deliver and for configuring locations for use with location triggered delivery. Links to the administrative pages appear in the Mobile Notification Settings section of the Deliver Settings page. Mobile messaging must be enabled for the hosted messaging account to display the Mobile Notification Settings section of the Deliver Settings page.

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure Domains</td>
<td>Controls access to the Policy Settings section of the Deliver Settings page. If the administrator's role is not granted permission to configure email domains, the administrator cannot see the Policy Settings section. This permission is also required to administer short link domains.</td>
</tr>
</tbody>
</table>

**Messaging permissions for Deliver**

Unica Deliver controls access to mailing features outside of the mailing tab in Campaign through the following pre-defined security roles.

- **Deliver_admin**
- **Deliver_user**

Users must have both roles to have access to Deliver mailing features.

**Assigning Deliver roles**

To provide a user with full access to Deliver mailing features, assign the pre-defined Deliver roles to the user.

1. In Unica Platform, navigate to **Settings > User roles & Permissions > Deliver > partition [n] > Deliver_admin**.
2. Click **Assign Users**.
3. Select the user from the list of available users. Click **Add** to assign the role to the user.
4. Repeat steps 1-3 for the **Deliver_user role**.
5. Save changes.

Controlling email domains and short link domains

Upon request, Unica configures one or more email domains for your hosted email account. Unica can also assign domains that marketers use to create shortened links in various types of messages. System administrators with appropriate permissions control the messaging domains that are available to marketers.

Depending on your business requirements, it might be desirable to restrict the list of messaging domains that are available to specific marketers. Deliver administrators restrict the list of available domains through security policies that are applied to folders in the Document Composer. The ability of marketers to create and edit email communications depends on the security policy that is applied to the folder that contains the communication.

Deliver administrators with appropriate permissions can control the list of email domains that Deliver users can use as the From: domain in email communications. Administrators can also control the list of short link domains that is presented to marketers when they configure communications that use shortened links. For example, you can specify which short link domains are available when marketers add a social sharing link to marketing messages.

Deliver administrators use the Policy Settings page to grant permissions to use specific messaging domains. Access to the Policy Settings page is controlled by the Email Administration permissions that are granted through the Campaign Global Policy. Only administrators with the appropriate permissions can restrict access to email domains through the Policy Settings page.

1. In the Settings menu, select Messaging Settings.
   
   If you have the appropriate administrative permissions, the Policy Settings section displays on the Deliver Settings page.

2. Click Display a list of policies and their settings.
A list of security policies that are configured for your Deliver installation displays.

3. Click a security policy that is associated with the system user whose messaging domain access you want to control.

The Domains section displays the email domains that are configured for your hosted messaging account.

The Short Link Domains section displays the short link domains that are configured for your hosted messaging account.

- In either section, click **Use all domains** to allow users that are associated with the policy to use any of the email domains that Unica configured for your hosted email account.

  This option is the default.

- Click **Use specific domains** to select specific domains.

  **Note:** If you select **Use specific domains**, you must update the domain permissions when you register a new email or short link domain for your hosted messaging account. The system does not assign permissions for the new domain automatically.

For the users associated with the security policy, only the selected email domains appear as an option for the **From:** address in email communications. For communications that require shortened links, marketers can choose only from the specific short link domains that you select.

After you save the new settings, the Document Composer updates the domain options that are available to marketers.

For more information about how Deliver marketers create and manage communications, see the Unica Deliver User's Guide.

**Maintenance of hosted email domains**

To send email messages you must register at least one email domain with Unica. To improve message deliverability, Unica works with you to establish and maintain the email
reputation of the domain with leading Internet Service Providers (ISPs) around the world. You can establish multiple email domains with Unica.

When you configure the header in an email communication, the system populates the From address with the email domain that you have registered with Unica. If you establish multiple email domains with Unica, the available domains display in a drop-down list. System administrators can control the email domains that email marketers can select or modify.

You can request that Unica add or delete email domains established for your hosted messaging account. After Unica completes the change, the system updates the list of available email domains. The change is reflected in the list of available email domains the next time you create or edit an email communication.

**Note:** Email domain changes for your account do not update email communications that you created before the change request. To change the email domain for a communication created previously, you must reopen the email communication and update the email domain selection.

For more information about how to register an email domain with Unica, see HCL Unica Domain Name Options for Email.

To request changes related to your email domains, contact Unica Deliver Services team via HCL technical support.

## Configuring default sender address and display names

For each email domain that you have registered with Unica, you can define a default email address and a default friendly name. The combination of email address or friendly name and the email domain appears as the From: address for the email messages that you send.

Administrators can configure the default sender and display names on the Domain Settings page. The domain settings are part of the Deliver Settings interface. Access to the Domain Settings page is controlled by the Email Administration permissions that are granted
through the Campaign Global Policy. Only administrators with the appropriate permissions can restrict access to email domains through the Policy Settings page.

1. Go to **Settings > Deliver Settings**. In the Domain Settings section, click **Display** the list of domain settings.

   The Domain Settings page lists the default display names and email addresses associated with email domains that are registered to your hosted email account. The list includes only the domains that your user permissions allow you to modify.

   The Default column indicates the combination of display name, address, and domain that appears as the default From address for new email communications.

2. Click **Edit**. The Edit Domain Settings window opens.

   The Domain Name column lists the available email domains. You can do the following for any of the domains.

   - In the From Display Name column, enter a friendly name to appear as the default for an email domain in the list.
   - In the From Address column, enter the local part of the email address to appear as the default for an email domain in the list.

3. Optionally, in the Default column, select one combination of display name address, and domain to appear as the default From address for new email communications.

   If you do not select a default, the system uses the first domain on the list to create the default From address for new email communications.

4. Save changes.

The new address settings apply to all new email communications that you create. The settings do not change address information for email communications that you created previously. To update previous email communications, you must reopen and modify each communication.
Controlling access to the list of sent messages

Deliver provides a list of messages that have been sent from your Deliver environment. Because the list includes links to messaging configurations, your security plans might require that you restrict access to the list.

The list of messages is presented on the Message Overview page. By default, all users in your Campaign and Deliver environment can see the list of sent messages. However, when you enable the access restriction, you can prevent specific users from seeing the menu option to open the page that contains the list.

Restricting access to the list of sent messages affects all partitions in your Campaign installation. If your Campaign installation includes multiple partitions, you must update user permissions separately in each partition to explicitly grant or deny permission to access the list.

Controlling who can access the list of sent messages requires a series of tasks to change user permissions and the system configuration.

<table>
<thead>
<tr>
<th>Task</th>
<th>More information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify users who can access the list of messages. At first, all users are granted access.</td>
<td>[Granting access to the list of sent messages](on page 150)</td>
</tr>
<tr>
<td>Identify users who are not allowed to access the list of messages.</td>
<td>[Denying access to the list of sent messages](on page 150)</td>
</tr>
<tr>
<td>Enable the access restriction.</td>
<td>[Enabling the restriction to the sent message list](on page 152)</td>
</tr>
</tbody>
</table>

When you complete these tasks, the Message Overview option on the Campaign menu is visible only to users with roles that explicitly grant permission to access the list of mailings.
Granting access to the list of sent messages

If you restrict access to the list of sent messages, you must specifically grant access to users who must access the list.

Users access the list of sent messages by clicking the **Message Overview** link on the **Campaign** menu. You can grant a user access to the list of all sent messages by assigning the user a top-level administrative role that is explicitly granted permission to see the **Message Overview** link.

The default top-level roles include **Admin**, **Execute**, **Design**, and **Review**. Permissions that you grant through the top-level roles apply to all objects in the partition.

1. Go to **Settings > User Roles and Permissions > Campaign > partition (n)**.
2. Click **Save and Edit Permissions**.
   A list of permissions for the partition opens. The available top-level roles are listed across the top to the page.
3. In the **Administration** section, explicitly grant the **View Mailing List Page** permission to every role.

When you enable access restrictions for the list of sent messages, users with roles that are explicitly granted the **View Mailing List Page** permission can see the **Message Overview** link on the **Campaign** menu.

Create a role to deny access to the list of sent messages.

Denying access to the list of sent messages

If you restrict access to the list of sent messages, you must specifically deny access to users who must not be allowed to access the list.
Users access the list of sent messages by clicking the **Message Overview** link on the **Campaign** menu. You can prevent a user from accessing the list of all sent messages by assigning the user a top-level administrative role that is explicitly denied permission to see the **Message Overview** link.

The default top-level roles include **Admin**, **Execute**, **Design**, and **Review**. Permissions that you grant through the top-level roles apply to all objects in the partition. You can create new top-level roles to supplement the default top-level roles. The new roles can grant or deny specific permissions.

1. Go to **Settings > User Roles and Permissions > Campaign > partition (n)**. The partition `<n>` page opens.
2. Click **Add Role**. Assign a name to the role and enter a brief description. Save the changes and return to the partition `<n>` page.
3. Configure the new role to deny access to the list of sent mailings.
   a. Click **Add Roles and Assign Permissions**. The **Properties for Administrative Roles** page opens. The new role displays in the list of roles.
   b. Click **Save and Edit Permissions**.
      A list of permissions for the partition displays as a matrix of selection icons that indicate the state of each permission for each role. The new role displays next to the other top-level roles across the top to the matrix.
   c. In the **Administration** section, explicitly deny the **View Mailing List Page** permission for the new role. Save changes.
4. Assign the new role to the users that you want to prevent from accessing the mailing list page.
   a. Go to **Settings > Users**. Select the user that you want to prevent from accessing the list of sent messages.
   b. Click **Edit Roles**. The new role that you created in the previous step (a role that is configured to deny access) appears in the list of **Available Roles**.
   c. Move the new role from **Available Roles** to **Roles**. Save changes.
When you enable access restrictions for the list of sent messages, a user that is assigned the new role cannot see the **Message Overview** link.

Update the configuration to enable access restrictions for the list of sent messages.

### Enabling the restriction to the sent message list

Users access the list of sent messages through the **Message Overview** option on the **Campaign** menu. If you restrict access to the list of sent messages, the Security Function ID property controls the display of this menu option and, therefore, controls access to the list of sent messages.

To restrict access to the list of sent messages, you must update the Security Function ID property in the Platform configuration. This property applies to all partitions in your Campaign installation.

When you populate the Security Function ID with the correct value, the **Message Overview** option is available to only those users with a role that explicitly grants the View Mailing List Page permission. Users with roles where the View Mailing List Page permission is either denied or not granted, cannot see the **Message Overview** option.

1. Go to **Settings > Configuration > Platform > Platform-wide navigation > Main navigation menu > Campaign > Deliver Mailings**. Click **Deliver Mailings** to display the configuration settings.

2. Click **Edit Settings**.

3. In the **Security Function ID** field, enter **7000**. Save changes.

   To see the results of the configuration change, log out of the system and log in again.

Only users with roles that explicitly grant the View Mailing List Page permission can see the **Message Overview** link to access the list of sent messages.
Permissions for Deliver reports

Your user permissions determine your ability to view Deliver reports.

For information about setting permissions to access standard Deliver reports, see the section in the Unica Insights Reports Installation and Configuration Guide for reporting and security.
Chapter 11. Technote (troubleshooting)

Problem (Abstract)

To use the Deliver components installed with Unica Campaign and send personalized marketing messages, you must connect the local Campaign installation to remote message resources hosted by HCL. This section describes how to configure such a connection when your corporate firewall rules prohibit direct communication with the hosted environment.

Resolving the problem

Typical communication with the hosted email resources environment

The following diagram illustrates the standard configuration for communication between the On Premises (OP) environment and On Demand (OD) environment.

The local Deliver OP environment requires external communication with Deliver OD environment using HTTPS and FTPes (explicit FTP) or FTPs (implicit FTP). This section uses the term FTPS to refer both explicit and implicit FTP.

The OP environment includes a web application server (either IBM WebSphere or Oracle WebLogic) on which you have deployed Campaign. Campaign hosts the Deliver components (RCT and RLU) that communicate with the hosted email resources in the OD environment.

The Response and Contact Tracker (RCT) downloads response data from the OD environment.

The Response List Uploader (RLU) uploads mailing lists and other required mailing data to the OD environment.
When the machine where Unica Deliver is installed cannot communicate directly with OD environment, Deliver supports communication with the hosted OD resources through a SOCKS proxy.

**Connecting to message services through a proxy**

The following diagram illustrates communication between the OP and OD environments when using a SOCKS proxy. Note that SOCKS proxy is configured on the local "On Premises" environment.
Verify the following points before you enable the proxy options.

- The proxy server is a SOCKS proxy.
- The proxy server can access the Deliver OD environment and allows the traffic to and from the ports configured in the HCL Data center used by your hosted email account.
- You have installed the SOCKS proxy in such a way that the Deliver OP environment can access the proxy.

Changes required for routing FTPS and HTTPS traffic through a SOCKS proxy

To use a SOCKS proxy to access email resources hosted by HCL, you must make changes to the web application where you have deployed Campaign and to the startup scripts for the Deliver RCT and RLU.
Making changes for FTPS

For FTPS traffic, apply the following configurations to the RLU and the web application server.

- `Dhcl.unica.deliver.ftps.proxy.host = <socksHost>`
- `Dhcl.unica.deliver.ftps.proxy.port = <socksPort>`
- `Dhcl.unica.deliver.ftps.proxy.match.hosts = <comma separated list of host names and IP addresses>`

socksHost is the host name or IP of the SOCKS proxy.

socksPort is the port on which SOCKS proxy is running.

`-Dhcl.unica.deliver.ftps.proxy.match.hosts` matches host names and IPs used when routing traffic through the SOCKS proxy.

The IP address specified for `-Dhcl.unica.deliver.ftps.proxy.match.hosts` is the IP address that FTP server in the hosted OD environment sends to the FTP client in the local OP environment as part of the FTPS protocol during data transfer.

Set `-Dhcl.unica.deliver.ftps.proxy.match.hosts` to one of the following values. (depends on the data center used by your hosted email account)

**US data center:** `-Dhcl.unica.deliver.ftps.proxy.match.hosts=ftp-em.unicadeliver.com`

**Demo environment:** `-Dhcl.unica.deliver.ftps.proxy.match.hosts=1SS.212.8.162`

**Europe data center:** `-Dhcl.unica.deliver.ftps.proxy.match.hosts=ftp-eu.unicadeliver.com`

Making changes for HTTPS

For HTTPS traffic, apply the following configurations to the RCT and the web application server.

- `Dhcl.unica.deliver.https.proxy.host= <socksHost>`
- `Dhcl.unica.deliver.https.proxy.port= <socksPort>`
-Dhcl.unica.deliver.https.proxy.type=SOCKS

socksHost is the host name or IP of the SOCKS proxy.
socksPort is the port on which SOCKS proxy is running.

**Authentication requirements when using a SOCKS proxy**
If your SOCKS proxy requires authentication, configure the following for the web application servers, RLU and RCT.

- -Dhcl.unica.deliver.proxy.auth.user = <username>
- -Dhcl.unica.deliver.proxy.auth.password = <password>

Where username and password are the credentials required to authenticate against the proxy.

To configure the RCT using a SOCKS proxy
Configure the RCT to work through a SOCKS proxy, follow the procedure for your operating system.

**For the RCT in Windows environment**
Add the following proxy arguments to common.bat, located in the \deliver\bin directory of your local Deliver installation.

```
set RCT PROXY ARGS=
-Dhcl.unica.deliver.https.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.https.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.https.proxy.type=SOCKS
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUIH USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUIH_PASSWORD>
```

```
set RCT_JAVA_ARGS=%BASE_VM_ARGS% %RCT_MEM_ARGS%
```
For the RCT in UNIX environments

Add the following proxy arguments to `common.sh` located in the `deliver/bin` directory of your local Deliver installation.

⚠️ **Note:** Do not make changes directly to `rlu.sh.rc.t.sh` or `setenv.sh` because they will be overridden.

```bash
RCT_PROXYARGS="
-Dhcl.unica.deliver.https.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.https.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.https.proxy.type=SOCKS
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUIH_USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUIH_PASSWORD>

RCT_JAVA_ARGS="${BASE_VM_ARGS} ${RCT_MEM_ARGS} ${RCT_EXTRA_VM_ARGS} ${RCT_PROXYARGS}""
```

To configure RLU using a SOCKS proxy

To configure the RLU to work through a SOCKS proxy, follow the procedure for your operating system.

For the RLU in Windows environment

Add the following proxy arguments to `common.bat` located in the `deliver/bin` directory of your local Deliver installation.

```bash
set RLU_PROXYARGS=

-Dhcl.unica.deliver.ftps.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.ftps.proxy.port=<PROXY_PORT>
```
For the RLU in UNIX environments

Add the following proxy arguments to common.sh, located in the `\deliver\bin` directory of your local Deliver installation.

⚠️ **Note:** Do not make changes directly to `rlu.sh` or `setenv.sh` because they will be overridden.

```
RLU_PROXY_ARGS=
-Dhcl.unica.deliver.ftps.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.ftps.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.ftps.proxy.match.hosts=<comma separated list of host names and IP addresses>
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUIH USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUIH_PASSWORD>

RLU_JAVA_ARGS="$(BASE_VM_ARGS) $(%RLU_MEM_ARGS% %RLU_EXTRA_VM_ARGS%) $(%RLU_PROXY_ARGS%)
```

Changes to the WebSphere configuration

Add the following to the WebSphere generic JVM arguments (see the screenshot):

- `Dhcl.unica.deliver.https.proxy.host=<PROXY_HOST>`
- `Dhcl.unica.deliver.https.proxy.port=<PROXY_PORT>`
-Dhcl.unica.deliver.https.proxy.type=SOCKS
-Dhcl.unica.deliver.ftps.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.ftps.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.ftps.proxy.match.hosts=<comma separated list of host names and IP addresses>
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUIH USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUTH_PASSWORD>

Changes to the Oracle WebLogic configuration

For WebLogic, modify the script.

In Windows environment

JAVA_OPTIONS=%(JAVA_OPTIONS)
-Dhcl.unica.deliver.https.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.https.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.https.proxy.type=SOCKS
-Dhcl.unica.deliver.ftps.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.ftps.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.ftps.proxy.match.hosts=<comma separated list of host names and IP addresses>
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUIH_USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUTH_PASSWORD>

In UNIX environments

JAVA_OPTIONS='(JAVA_OPTIONS)
-Dhcl.unica.deliver.https.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.https.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.https.proxy.type=SOCKS
-Dhcl.unica.deliver.ftps.proxy.host=<PROXY_HOST>
-Dhcl.unica.deliver.ftps.proxy.port=<PROXY_PORT>
-Dhcl.unica.deliver.ftps.proxy.match.hosts=<comma separated list of host names and IP addresses>
-Dhcl.unica.deliver.proxy.auth.user=<PROXY_AUIH_USER>
-Dhcl.unica.deliver.proxy.auth.password=<PROXY_AUTH_PASSWORD>'
Chapter 12. Deliver channel vendor account configuration

Deliver supports SMS, WhatsApp, and Push as delivery channels in addition to email. SMS is supported using different vendors, so that customer has a choice to select a SMS partner based on the geographical and cost aspects. Whenever a customer decides to use a specific vendor for SMS or WhatsApp messages, HCL works with the customer and required vendor to onboard him smoothly to Deliver. As part of this process, customer’s account is created with each vendor. This account enables Deliver to send messages on behalf of that customer and process responses from users.

Deliver supports the following channels.

- SMS using Karix vendor
- SMS using RML vendor (for both referral and reseller type of licenses)
- Whatsapp using RML vendor
- Push using Kumulos as vendor
- Twilio SMS

The following document explains these steps for each of these channels. These steps must be performed by or for each customer account as part of the onboarding process.

Karix SMS account configuration

Perform the following steps to configure the customer’s Karix SMS account for it to work with Deliver.

1. Log in to Karix Console (www.karix.solutions) and click the API Keys button on dashboard to create API key and configure with Deliver.
2. In the top right corner, under the **Open My Account** list, select **Edit My Info** option and note the Sender ID configured for your Karix account to configure in Deliver.

3. Provide the API Key created in Step 1 and the Sender ID noted in Step 2 for it to be configured in the account.

4. Set the callback URL in the Karix console.

   **Note:** Once the account is configured in Deliver, `<account>` in above URL must be replaced with the account name provided by Deliver Provisioning team.
RML SMS account configuration

The customer’s RML SMS account must have the following configurations for it to work with Deliver.

**RML account onboarding**

- You must work with RML team to create a SMS account for India or World Wide location based on the customer’s location.
- The senderid must to be provided by the customer configured by RML.
- SMS templates also require to be whitelisted based on target SMS sending location. For example, pre-approved templates required for USA and Canada.

RML has different login URLs based on geography, which they provide as part of account creation email from their side. For example, URLs for India and Worldwide datacenters

- India account: [https://ems.rmlconnect.net/](https://ems.rmlconnect.net/)
• Worldwide account: https://client.rmlconnect.net/login

Perform the following steps.

1. Login to RML Console as per the above URLs and navigate to Utilities > DLR Push URL.

   **Note:** You must ask RML to add this menu while provisioning account.

2. Set the callback URL in RML console.
   - For US datacenter: https://smsin-us.unicadeliver.com/deliversmsib/sms?partition=<account>&provider=RML
   - For EU datacenter: https://smsin-eu.unicadeliver.com/deliversmsib/sms?partition=<account>&provider=RML

   **Note:** Once the account is configured in Deliver, <account> in above URL must be replaced with the account name provided by Deliver Provisioning team.

### RML WhatsApp account configuration

The customers who are provisioned with the WhatsApp feature (provided by RML) must perform the following steps for configuration.

1. Create a verified Facebook Business Manager account setup for WhatsApp. The RML team will guide customers to configure Facebook Business Manager account as required.
2. Create a WhatsApp account with RML. RML will create this account after the Facebook Business Manager account verification is complete.
3. Create Message templates approved by WhatsApp. The RML team will work with customers to prepare message templates in required format and getting it approved from customer.
4. Once RML provides approved templates to the customers, they require to upload them to the Deliver Message Editor using New > WhatsApp Content menu item. All the details
must be exactly provided as per approved template, since WhatsApp does not allow any changes to template post approval.

5. Configure callback URL in RML WhatsApp account. For this, need to provide below URL to RML so that they can configure it as callback URL for WhatsApp message deliver reports.

https://smsin-us.unicadeliver.com/deliversmsib/wa/<account>

Twilio SMS account configuration

Customer's Twilio SMS account requires the following configurations for it to work with Deliver-

1. Log in to Twilio Console (https://www.twilio.com/console) and create a project for SMS. Copy ACCOUNT SID and AUTH TOKEN for this project.
2. Create a messaging service ID for this account at https://www.twilio.com/console/sms/services and copy its messaging service ID for further usage.
3. Work with Twilio onboarding team to upgrade account to ensure sending SMS to any mobile number. Trial accounts need whitelisting of numbers before SMS can be sent to that number.
Index

c
  configTool
  128
  configTool utility
  128

U
  utilities
  configTool
  128