

Superset Reports Installation and Configuration Guide



Contents

Chapter 1. Overview.....	3
Chapter 2. Superset Reports Installation and Configuration Guide.....	4
Plan your Superset reports installation.....	4
Prerequisites.....	4
Prerequisites for Unica Deliver Reports.....	5
Preinstallation Checks for Superset.....	7
Install and configure Superset reports.....	7
Install Superset reporting components.....	8
Database Compatibility for Superset Reports.....	8
Configuring a user with the Reports System role.....	9
Loading templates for the Reports SQL Generator.....	9
Creating and populating reporting tables for Unica Campaign and Unica Deliver.....	9
Creating views or materialized views for Unica Campaign and Unica Deliver only:	9
Setting up data synchronization.....	10
Unica Campaign Configuration and Reporting Schema.....	10
Unica Deliver Configuration and Reporting Schema.....	16
Unica Interact RTP Configuration and Reporting Schema.....	18
Unica Journey Configuration and Reporting Schema.....	19
Unica Plan Configuration and Reporting Schema.....	20
For Unica Deliver only: Stored procedures for delta processing.....	21
For Unica Deliver only: Creating stored procedures, staging tables, and indexes.....	22
For Unica Deliver only: How to schedule and run stored procedures.....	24
Sample configuration of stored procedures for Oracle.....	26
Sample configuration of stored procedures for Microsoft™ SQL Server.....	29
Sample configuration of stored procedures for DB2.....	31
Sample configuration of stored procedures for MariaDB.....	33
Sample configuration of stored procedures for OneDB.....	35
Known limitations.....	38
Known issues.....	39

Chapter 1. Overview

HCL Unica 12.1.9 introduces support to Apache Superset (an Open source BI tool). This tool has competitive dashboarding capabilities.

You can perform the following activities:

- Create quick charts and group them together in dashboards.
- Refresh and download reports to share them with the teams.

HCL Unica supports Superset reports for the following Unica products:

- Unica Campaign
- Unica Deliver
- Unica Interact RTP
- Unica Journey
- Unica Plan

Since this is the first release, Superset supports the majority of Superset reports for products. It increases the flexibility of creating dashboards with ease for users.

Chapter 2. Superset Reports Installation and Configuration Guide

This guide provides information on how to install and configure Superset reports.

Plan your Superset reports installation

When you plan your Superset reports installation for HCL Unica products, you must ensure that you correctly set up your system and configure your environment. Review the prerequisites carefully.

Prerequisites

Before you install or upgrade any HCL Unica product, you must ensure that your computer complies with all the prerequisite software and hardware.

System requirements

For information about system requirements, see the Recommended Software Environments and Minimum System Requirements guide.

Network domain requirements

The HCL Unica products that are installed as a suite must be installed on the same network domain to comply with the browser restrictions that are designed to limit the security risks that can occur with cross-site scripting

Knowledge requirements

To install HCL Unica products, you must have a thorough knowledge of the environment in which the products are installed. This knowledge includes knowledge about operating systems, databases, and web application servers.

Access permissions

Verify that you have the following network permissions to complete the installation tasks:

- Administration access for all required databases.
- Read and write access to the relevant directory and sub-directories for the operating system. account that you use to run the web application server and HCL Unica components.
- Write permission for all files that you must edit.
- Write permission for all directories where you must save a file, such as the installation directory and backup directory if you are upgrading.
- Appropriate read, write, and execute permissions to run the installer.

Verify that you possess the administrative password for your web application server. The following additional permissions are required for UNIX:

- The user account that installs and Unica Platform must be a member of the same group as the Unica Campaign users. This user account must have a valid home directory, and have write permissions for that directory.
- All installer files for HCL products must have full permissions, for example, `rwxr-xr-x`.

The HCL Unica installer installs a JRE in the top-level directory for the HCL Unica installation. Individual HCL Unica application installers do not install a JRE. Instead, they point to the location of the JRE that is installed by the HCL Unica installer. You can reset the environment variable after all installations are complete.

For more information about the supported JRE, see the Recommended Software Environments and Minimum System Requirements guide.

Unica Platform requirement

You must install or upgrade Platform before you install or upgrade any HCL Unica products. For each group of products that work together, you must install or upgrade Platform only once. Each product installer checks whether the required products are installed. If your product or version is not registered with Platform, a message prompts you to install or upgrade Platform before you proceed with your installation. Unica Platform must be deployed and running before you can set any properties on the Settings > Configuration page.

Supported locales

Currently, only English language is supported.

Settings for HCL OneDB

When setting or configuring the OneDB environment, set the variable `DELIMITIDENT` to `Y` to avoid syntax errors in queries and procedures.

Prerequisites for Unica Deliver Reports

If you use Unica Deliver reports, you must comply with specific prerequisites in addition to the system requirements for installing reports.

For better performance, you might need 40% of the data size as temporary table space. Work with your database administrator to regularly fine-tune your database. For best results, you can store Unica Deliver system tables in a separate table space that is mounted on a separate, unshared disk.

Settings for DB2®

If you use DB2®, you must use version 9.7.8 or higher.



Important: You must set the following value before you begin applying the Unica Deliver reports:

```
db2set DB2_COMPATIBILITY_VECTOR=ORA
```

Sample sizes for DB2®

For an DB2® Unica Campaign database of approximately 600 GB with most of the data population in the reporting tables (UCC_*), you might use the following settings:

- Tablespace page size: 16K
- Temp tablespace: 250 GB
- db2 update db cfg using auto_reval DEFERRED_FORCE;
- db2 update db cfg using decflt_rounding ROUND_HALF_UP;
- db2 update db config using LOGFILSIZ 102400;
- db2 update db config using logprimary 13;
- db2 update db config using LOGSECOND 25;
- db2stop force
- db2start



Important: The size of your transaction log can affect reports processing. Review your database environment requirements, especially the size of the transaction log, with your database administrator.

Settings for Oracle

If you use Oracle, you must use version 11g or higher. Review your environment requirements with your database administrator.

Sample sizes for Oracle

For an Oracle Unica Campaign database of approximately 650 GB with most of the data population in the reporting tables (UCC_*), you might use the following settings:

- Temp tablespace: 250 GB
- Size for REDO logs: 2 GB
- Number of REDO logs: 4

Settings for Microsoft™ SQL Server

If you use Microsoft™ SQL, you must use SQL Server 2008 or higher. Review your environment requirements with your database administrator.

Sample sizes for Microsoft™ SQL Server

For Microsoft™ SQL Server database of approximately 520 GB with most of the data population in the reporting tables (UCC_*), you might use the following settings:

- Temp tablespace: 250 GB

Settings for Internet Explorer

If you use Internet Explorer, make sure that the browser security settings allow automatic prompting for file downloads. Complete the following steps to verify that the browser allows automatic prompting for file downloads:

1. Open Internet Explorer and go to **Tools > Internet Options**.
2. In the **Security** tab, click **Custom Level**.
3. Scroll down to the **Downloads** section.
4. Make sure that the **Automatic prompting for file downloads** option is set to **Enable**.

Preinstallation Checks for Superset

Before you begin the installation of Superset, confirm that the following files exist for the mentioned products in the mentioned folder paths:

Products	Folder Paths and Files
Unica Campaign	<code><CAMPAIGN_HOME_DIRECTORY>\Report\superset</code> <ul style="list-style-type: none"> • <code>sp_campaign_update_dashboards_tags.sql</code> • <code>campaignsupersetreports.zip</code>
Unica Deliver	<code><CAMPAIGN_HOME_DIRECTORY>\Report\superset</code> <ul style="list-style-type: none"> • <code>sp_deliver_update_dashboards_tags.sql</code> • <code>deliversupersetreports.zip</code>

Install and configure Superset reports

For information related to installation of Superset, see *Cloud Native Unica Installation and Configuration Guide*.



Note:

- Before creating a Superset dashboard, confirm if you have completed the Superset configuration.
- Before enabling Superset reports, ensure that you have disabled Unica Insights Reports.

To enable Superset reports within Unica, you must configure the following properties in the **Settings > Configuration** page and expand **Reports** and **Integrations**, and select **Superset** (Affinium | Reports | Integrations | Superset).

Enabled

Set to `True` to enable the Unica and Superset integrations. By default, this is set to `False`.

Superset URL

Superset instance URL.

Superset secret holder

Platform username against which the datasource created with name specified in **Superset datasource** property. The datasource contains the Superset Admin user login credentials, for example, `asm_admin`.

Superset Datasource

Datasource name created under the user specified in **Superset secret holder** property. For example, `SUPERSET_DS`.

Once the superset configuration in Unica is completed, you can create Superset dashboard through Platform **Create dashboard** page.

You will see the **Embed Superset dashboard** option while creating new dashboard only if following criteria is met:

- The `ReportsSystem` role is assigned to the logged in user.
- Under **Superset** node in Platform configuration, the **Enabled** flag is set to `True`.

After you create the embedded Superset dashboard, you can perform the all operations that can be performed in a standalone Superset environment.



Note:

- User without `ReportsSystem` role, cannot create or edit the embedded Superset dashboard within Unica.
- On click of Edit dashboard, all the list of charts available in Superset will be listed.
- Deleting the Platform dashboard, will also delete its corresponding dashboard created in the Superset.
- On the embedded Superset dashboards, edit chart and create chart are disabled using custom CSS.
- If the charts added into embedded Superset dashboard uses `userid` in the SQL query then only user specific data is shown in the dashboard

Install Superset reporting components

To install Superset reports for your HCL Unica products, you must install the Superset reporting components.

Reporting components include the following items:

- HCL Unica integration components
- Reporting schemas

Database Compatibility for Superset Reports

The database compatibility of Superset Reports for the various products of Unica ae as follows:

Unica Product	Database
Unica Campaign	IBM DB2, Maria DB. Microsoft SQL Server, and Oracle,

Unica Product	Database
Unica Deliver	IBM DB2, Maria DB, Microsoft SQL Server, and Oracle,
Unica Journey	MariaDB
Unica Plan	MariaDB
Unica Interact RTP	MariaDB



Note: All supported databases refers to the databases supported by the product. For more information related to supported databases, see the *Recommended Software Environments and Minimum System Requirements* guide.

Configuring a user with the Reports System role

You must configure a user with the Reports System role. This role is used to configure reporting properties and to generate the SQL script that is used to create the reporting schemas.

A user with the Reports System role can access the Configuration and Report SQL Generator pages. You must configure a user with access to the HCL Unica Settings > Configuration and Settings > Report SQL Generator pages. Then, you can log in as this user to configure the reporting properties and generate the SQL script that is used to create the reporting schemas.

To configure a user with the Reports System role, complete the following steps.

1. Create a user.



Note: You can also use the platform_admin user.

2. Go to **User Roles and Permissions > Report > PartitionN** and assign the Reports System role to that user.
3. Verify that the user has access to the **Settings > Configuration** and **Settings > Report SQL Generator** pages.
4. Grant the roles ReportsSystem (Unica Platform Report), ReportsUser (Unica Platform Report) to user.

Loading templates for the Reports SQL Generator

For more information, see *Unica Insights Reports Installation and Configuration Guide*

Creating and populating reporting tables for Unica Campaign and Unica Deliver

For more information, see *Unica Insights Reports Installation and Configuration Guide*

Creating views or materialized views for Unica Campaign and Unica Deliver only:

For more information, see *Unica Insights Reports Installation and Configuration Guide*

Setting up data synchronization

For more information, see *Unica Insights Reports Installation and Configuration Guide*

Unica Campaign Configuration and Reporting Schema

Datasource Configuration

1. Login to the Superset environment.
2. Navigate to **Settings > Database Connections**.
3. Edit **CampaignDS**.
4. Provide password for the database user.
5. Test the connection.

Verify the Setup

1. Navigate to Dashboards.
2. Filter **Campaign_Analytics**, **Campaign_Performance**, **Campaign_OS_offer**, **Campaign_Dashboard**, **Campaign_OS_Campaign**, **CA_Offer_Performance_Over_Time**, and **Campaign_Offer_Performance_Over_Time**.
3. Run each dashboard. The Dashboard mut run without any error.

Reports

Table 1. Campaign Analytics

Report Name	Description
Campaign summary	The Campaign summary report provides an overview of all campaigns that have been created. It lists the campaign code, creation date, start and end dates, last run date, initiative and objective of each campaign. User can filter report by the campaign name.
Offer Campaign listings	The Offer campaign listings report lists campaigns grouped by offers. It lists the campaign code, initiative, start and end dates and last run date. User can filter report by the campaign name.

Table 2. Campaign Performance

Report Name	Description
Campaign financial summary by offer (Actual)	The Campaign financial summary by offer (Actual) report provides financial data for offers within campaigns. It includes data such as contact costs, gross revenue, net profit, and ROI. User can filter report by the campaign name.

Table 2. Campaign Performance (continued)

Report Name	Description
Campaign detailed offer response breakout	The Campaign detailed offer response breakout report provides campaign performance data for different response types. It lists all offers associated with a campaign and the number of responses for each response type. User can filter report by the offer name.
Campaign performance comparison	The Campaign performance comparison report compares the financial performance of campaigns. It includes data such as response transactions and response rate, number of unique responders and responder rate. It also includes optional lift over control group information. User can filter report by the campaign name.
Campaign performance comparison by initiative	The Campaign performance comparison by initiative report compares the financial performance of campaigns grouped by the corresponding initiatives. It includes data such as response transactions and response rate, number of unique responders and responder rate. It also includes optional lift over control group information. User can filter report by the campaign name.
Campaign performance comparison (with revenue)	The Campaign performance comparison report compares the financial performance of campaigns. It includes data such as response transactions and response rate, number of unique responders and responder rate. It also includes optional lift over control group information. User can filter report by the campaign name.
Campaign performance summary by cell	The Campaign performance summary by cell report provides performance data for campaigns with cells grouped by the corresponding campaigns. It includes data such as number of offers given, number of response transactions, response rate, number of unique responders and responder rate. It also includes optional lift over control group information. User can filter report by the campaign name.
Campaign performance summary by cell and initiative	The Campaign performance summary by cell and initiative report provides performance data for campaigns with cells grouped by the corresponding campaigns and initiatives. It includes data such as number of offers given, number of response transactions, response rate, number of unique

Table 2. Campaign Performance (continued)

Report Name	Description
	responders and responder rate. It also includes optional lift over control group information. User can filter report by the campaign name.
Campaign performance summary by cell and offer	The Campaign performance summary by cell and offer report provides a summary of cell and offer performance from the cell perspective. It includes data such as number of offers given, number of response transactions, response rate, number of unique responders and responder rate. It also includes optional lift over control group information. User can filter report by the campaign name.
Campaign performance summary by cell and offer (with revenue)	The Campaign performance summary by cell and offer (with revenue) report provides a summary of cell and offer performance from the cell perspective. It includes data such as number of offers given, number of response transactions, response rate, number of unique responders, responder rate and actual revenue. It also includes optional lift over control group information. User can filter report by the campaign name.
Campaign performance summary by cell (with revenue)	The Campaign performance summary by cell (with revenue) report provides performance data for campaigns with cells grouped by the corresponding campaigns. It includes data such as number of offers given, number of response transactions, response rate, number of unique responders, responder rate and actual revenue. It also includes optional lift over control group information. User can filter report by the campaign name.
Campaign performance summary by offer	The Campaign performance summary by offer report provides a summary of campaign and offer performance with offers grouped by the corresponding campaigns. It includes data such as number of offers given, number of response transactions, response rate, number of unique responders, and responder rate. It also includes optional lift over control group information. User can filter report by the campaign name.
Campaign performance summary by offer (with revenue)	The Campaign performance summary by offer (with revenue) report provides a summary of campaign and offer performance with offers grouped by the corresponding

Table 2. Campaign Performance (continued)

Report Name	Description
	campaigns. It includes data such as number of offers given, number of response transactions, response rate, number of unique responders, responder rate and actual revenue. It also includes optional lift over control group information. User can filter report by the campaign name.
Offer performance comparison	The Offer performance comparison report compares the performance of offers. It includes data such as number of offers given, number of response transactions, response rate, number of unique responders, and responder rate. It also includes optional lift over control group information. User can filter report by the offer name.
Offer performance metrics	Offer performance metrics report compares the performance of offers based on various response attributions, such as Best Match, Fractional Match and Multiple Match. It also includes optional lift over control group information and percentage difference between various attribution rates. User can filter report by the offer name.
Offer performance summary by campaign	The Offer performance summary by campaign report provides a summary of offer and campaign performance with campaigns grouped by the corresponding offers. It includes data such as number of offers given, number of response transactions, response rate, number of unique responders and responder rate. It also includes optional lift over control group information. User can filter report by the campaign name.

Table 3. Campaign Offer Performance Over Time

Report Name	Description
Campaign offer performance by month	The Campaign offer performance by month report shows campaign performance for a specified month with performance data grouped by the corresponding offers. It lists the number of offers given, number of response transactions and response rate for the specified month. User can filter report by the offer name, campaign name and year/month.

Table 3. Campaign Offer Performance Over Time (continued)

Report Name	Description
Offer performance by day	The Offer performance by day report shows offer performance for a specified date or date range. It lists the number of offers given, number of response transactions and the response rate during the specified date or date range. User can filter report by the offer name and actual date.

Table 4. Campaign Out-of-the-Box Campaign

Report Name
Campaign flowchart status summary
Campaign performance summary by cell
Campaign detailed offer response breakout
Campaign offer performance by month
Campaign performance summary by cell and offer
Campaign financial summary by offer (Actual)
Campaign performance summary by cell and offer (with revenue)
Campaign performance summary by cell (with revenue)
Campaign performance summary by offer

Table 5. Campaign Out-of-the-Box Offer

Report Name	Description
Revision history	JAVA-based report.
Offer performance summary by Campaign	
Offer performance by day	

Table 6. Campaign Dashboard

Report Name	Description
Campaign return on investment comparison	Campaign Return on Investment Comparison report gives you return on investment for respective campaigns and you can filter report by the name of the campaign.

Table 6. Campaign Dashboard (continued)

Report Name	Description
Campaign response rate comparison	Campaign Response Rate Comparison In this report we will get the response rate for respective campaigns and filter report on the basis of the campaign name.
Campaign revenue comparison by offer	The campaign revenue comparison by offer report gives you the gross revenue associated with each offer. You can filter it by the name of the offer.
Offer responses for last 7 days	The Offer Responses for last 7 days report will provide you with the response transactions and the offer given information for the response date of the offer. You can filter these reports according to the offer name, if you wish to. User can filter report by the offer name.
Offer response rate comparison	With the Offer Response Rate Comparison report, you can view information about offers with response rates that you can filter by offer name.
Offer response breakout	With the Offer Response Breakout report, you will get the commit response rate, consider response rate, explore response rate, usage response rate, and fulfill response rate of offers. Reports can also be filtered by offer name. User can filter report by the offer name.
My recent campaigns	JAVA-based reports.
My recent sessions	JAVA-based reports.
My custom bookmarks	JAVA-based reports.
Campaign monitor portlet	JAVA-based reports.

Customize your Reports

You can customize or create new Unica Campaign reports using out of the box datasets. These datasets consist of database SQLs. We recommend that you do not modify existing datasets in case you want to create a new field in the same. Instead, you can make a copy of it to modify SQL of datasets.

For more information, see <https://superset.apache.org/docs/intro> documentation.

Database Configurations for Superset

If you want to configure Superset on On Premises version of Unica, you must run a few SQL functions to activate Superset for use on On Premises setup.

1. Access the home directory of Campaign.
2. Navigate to `Report\superset\ddl`.
3. Depending on the database of your setup, select one of the following folders:
 - If your setup has DB2 database, open the `db2` folder.
 - If your setup has MariaDB database, open the `mariadb` folder.
 - If your setup as Oracle database, open the `oracle` folder.
 - If your setup has Microsoft SQL Server database, open the `sql` folder.
4. Open the `superset_report_functions.sql` file.
5. Manually run all the functions in that file.

These functions are applicable for both Unica Campaign and Unica Deliver.

Unica Deliver Configuration and Reporting Schema

Datasource Configuration

1. Login to the Superset environment.
2. Navigate **Settings > Database Connections**.
3. Edit **CampaignDS**.
4. Provide password for the database user.
5. Test the connection.

Verifying Setup

1. Navigate to Charts.
2. Navigate to Dashboards .
3. Filter **Deliver_Analytics**.
4. Filter **Deliver_Dashboard**.
5. Run each dashboard. The Dashboard mut run without any error.

Dashboards

Table 7. Deliver Analytics

Report Name	Description
Message Overview	This report provides a summary of the overall effectiveness of your email marketing campaign. User can able to apply filter on the basis of time range,Based on campaign Name and based on mailing Instances.
Detailed Link Report	This report provides a summary of link click data at the mailing level. User can able to apply filter on the basis of time range. Based on campaign Name and based on mailing Instances.

Table 7. Deliver Analytics (continued)

Report Name	Description
Detailed Link Report LP	This report provides a summary of Landing Page Link click data at the mailing level. User can able to apply filter on the basis of time range. Based on campaign Name and based on mailing Instances.
Detailed Link by Cell Report	This report provides a summary of link click data for each cell used by a mailing. User can able to apply filter on the basis of time range,Based on campaign Name. Based on CellCode and based on mailing Instances.
Detailed Link by Cell Report LP	This report provides a summary of LP Link click data for each cell used by a mailing. User can able to apply filter on the basis of time range. Based on campaign Name,Based on CellCode and based on mailing Instances
A/B Testing Performance Report	This report provides a summary of detailed mailing and response data that are generated by a single A/B test. User can able to apply filter on the basis of time range. Based on campaign Name and based on mailing Instances.
Detailed Bounce Report	This report provides a summary of the breakdown of the deliver failures by category. User can able to apply filter on the basis of time range. Based on campaign Name and based on mailing Instances.
Deliver reports processing overview	This report monitors the reports data processing and to determine whether data is available for use in Deliver reports. User can able filter reports data based on RunID.
SMS Message Summary Report	This reports provides a summary of the overall effectiveness of your SMS marketing campaign. User can able to apply filter on the basis of time range. Based on campaign Name and based on mailing Instances.
WhatsApp Message Summary Report	This reports provides a summary of the overall effectiveness of your WhatsApp marketing campaign. User can able to apply filter on the basis of time range. Based on campaign Name and based on mailing Instances.
Mobile Push Message Summary Report	This report provides a summary of the overall effectiveness of your Push marketing campaign. User can able to apply filter on the basis of time range. Based on campaign Name and based on mailing Instances.

Table 8. Deliver Dashboard

Report Name	Description
Email Communication Report	This Report Provides the data related to Email Channel Mailing according to their mailing instances in the bargraph chart format.
SMS Communication Report	This Report Provides the data related to SMS Channel Mailing according to their Message instances in the bargraph chart format.
WhatsApp Communication Report	This Report Provides the data related to WhatsApp Channel Mailing according to their Message instances in the bargraph chart format.
Push Communication Report	This Report Provides the data related to Push Channel Mailing according to their Message instances in the bargraph chart format.
Communication Delivery by Channel Type	This Report Provides Delivery Data for all The channel Types in The bargraph chart format
Sent Time Analytics Heatmap (Engagement)	This Reports give details about Response count With respect to time in Heatmap Chart format.
Recent email bounce responses	This Report Provides you the recent email bounce responses details in table chart format.
Recent email campaigns sent	This Report Provides you the recent Campaign Sent details in table chart format.

Unica Interact RTP Configuration and Reporting Schema

Datasource configuration

1. Login to Superset environment.
2. Navigate **Settings > Database Connections**.
3. Edit RTPDS.
4. Provide password for the database user.
5. Test Connection.

Verifying the Setup

1. Navigate to Dashboards.
2. Filter RTP_Dashboard.
3. Run the dashboard. The Dashboard must run without any error.

Dashboards

Report Name	Description
Domain Performance	This dataset provides a performance summary of offers contacted, accepted and rejected by audience count for each domain.

Unica Journey Configuration and Reporting Schema

Datasource Configuration

1. Login to the Superset environment.
2. Navigate **Settings > Database Connections**.
3. Edit **JourneyDS**.
4. Provide password for the database user.
5. Test the connection.

Verifying the Setup

1. Navigate to Dashboards.
2. Filter **Journey_Analytics** or **Journey_Dashboard**.
3. Run each dashboard they should run without any error.

Journey Datasets

Report Name	Description
Audience Comparison Entry Sources	This dataset provides a comparison of Entry sources with respect to audiences.
Audience Comparison Across Channel	This dataset provides a comparison of Channels (touchpoints) with respect to audiences.
Journey Channel Performance	This dataset provides a comparison of Journeys with respect to various channels.
Milestone Funnel	This dataset shows a Journeys progress throughout defined milestones.
Accepted/Rejected Audiences Across Journeys	This dataset provides a detail about accepted and rejected audience data for Journeys.
Journey Overview	This dataset provides details about Active, Completed, Paused, Draft Journey Counts.

Customize your Reports

You can customize or create new Unica Journey reports using out of the box datasets. These datasets consist of database SQLs. We recommend that you do not modify existing datasets in case you want to create a new field in the same. Instead, you can make a copy of it to modify SQL of datasets.

For more information, see <https://superset.apache.org/docs/intro> documentation.

Unica Plan Configuration and Reporting Schema

Configuration Steps

1. Access the Superset environment .
2. Navigate to **Settings > Database Connections**.
3. Edit **PlanDS**.
4. Provide the password for the database user.
5. Test the connection.

Verification Steps

1. Navigate to Charts.
2. Filter **Plan_Analytics**.
3. Navigate to Dashboards.
4. Filter **Plan_Analytics**.
5. Run each dashboard. The Dashboard must run without any error.

Unica Plan Datasets

Report Name	Description
Project Budget	This dataset shows system projects with their forecast, allocated and actual budges.
My Tasks	This dataset shows all users' tasks with various statuses with their start and completed timelines.
My Approvals	This dataset shows all users' tasks with various statuses with their start and completed timelines.

Customize your Reports

You can customize or create new Unica Plan reports using out of the box datasets. These datasets consist of database SQLs. We recommend that you do not modify existing datasets in case you want to create a new field in the same. Instead, you can make a copy and modify SQL of datasets.

For more information, see <https://superset.apache.org/docs/intro> documentation.

For Unica Deliver only: Stored procedures for delta processing

Unica Deliver reports require staging tables that are associated with the Unica Deliver system tables. The system tables are part of the Unica Campaign schema. You must periodically run stored procedures to process message response data for use in Unica Deliver reports.

For more information about schema changes, see *Unica Deliver System Tables and Data Dictionary*.

The initial setup for the Unica Deliver stored procedures relies on the following database scripts:

- `acer_indexes_dbname.sql`
- `acer_tables_dbname.sql`
- `acer_scripts_dbname.sql`



Note: If you observe any issues while executing the mentioned sql files, use the script terminator based on the database client. If your database client shows errors for `acer_scripts_dbname.sql`, create the procedures one after the other.

In case of DB2 database, execute the following script from the DB2 command line:

```
db2 -td! -svtf acer_scripts_db2.sql
```

Alternatively, you can also use any SQL client tools, set the delimiter to "!", and run the earlier mentioned script.

The database scripts are in the `Campaign\reports\Deliver-ddl` directory for the Oracle, IBM® DB2®, and Microsoft™ SQL Server databases.

The scripts set up indexes, tables, views, and stored procedures. The stored procedures refresh message data to populate the staging tables. The batch procedures must be run regularly to populate the staging tables. Running the stored procedures is referred to as delta processing.

The initial runs of the Unica Deliver stored procedures can take a long time to complete, depending on the amount of data that is contained in your tables. Subsequent delta processing also can take a long time to complete. You can significantly reduce the processing time by limiting the number of mailing instances (containers) that are processed by the stored procedures.

By default, data is processed for the past 90 days. However, you can change the default value before or after you run the SQL scripts for Unica Deliver.

Example for Oracle

The following examples for an Oracle database illustrate the changes that you can make to the `acer_tables` script to limit processing to the previous 30 days:



Note: The changes include modifying the `UARE_MAILING_MASTER` view.

Definition of the current view

```
CREATE VIEW UARE_MAILING_MASTER AS
(
(SELECT UCC_CONTAINER.CAMPAIGNID,UCC_CONTAINER.CONTAINERID,
substr(UCC_CONTAINERATTR.STRINGVALUE,1,100) AS CAMPAIGN_NAME,
UCC_CONTAINER.CONTAINERNAME AS MAILING_INST,
UCC_CONTAINER.CREATED AS MAILING_CREATED,
UCC_CONTAINER.CONTAINERTYPEID CONTAINERTYPEID,
UCC_CONTAINER.CONTCHANNELTYPEID CONTCHANNELTYPEID
FROM
UCC_CONTAINER,UCC_CONTAINERATTR
WHERE
UCC_CONTAINERATTR.CONTAINERID=UCC_CONTAINER.CONTAINERID AND
UCC_CONTAINERATTR.ATTRIBUTENAME='CampaignName' AND
UCC_CONTAINER.CREATED >= sysdate - 91
)
```

Definition of the modified view

```
CREATE VIEW UARE_MAILING_MASTER AS
(
SELECT UCC_CONTAINER.CAMPAIGNID, UCC_CONTAINER.CONTAINERID,
substr(UCC_CONTAINERATTR.STRINGVALUE,1,100) AS CAMPAIGN_NAME,
UCC_CONTAINER.CONTAINERNAME AS MAILING_INST, UCC_CONTAINER.CREATED AS
MAILING_CREATED FROM UCC_CONTAINER,UCC_CONTAINERATTR WHERE
UCC_CONTAINERATTR.CONTAINERID=UCC_CONTAINER.CONTAINERID AND
UCC_CONTAINERATTR.ATTRIBUTENAME='CampaignName'
AND
UCC_CONTAINER.CREATED >= sysdate - 30
)
```

To view all available report data, modify the `UARE_MAILING_MASTER` view to remove the date filter from the view. Then, refresh all Oracle or DB2® materialized views. For example, in the sample view creation that is shown above, remove the following line:

```
UCC_CONTAINER.CREATED >= sysdate - 30
```

For Unica Deliver only: Creating stored procedures, staging tables, and indexes

After you install or upgrade reporting templates, you must run specific SQL scripts before you generate Unica Deliver reports. The SQL scripts create stored procedures and staging tables.

About this task

The `Campaign/reports/Deliver-ddl` directory is on the server. This directory contains the following database scripts for Oracle, IBM® DB2®, and Microsoft™ SQL Server:

- `acer_indexes_dbname.sql`
- `acer_tables_dbname.sql`
- `acer_scripts_dbname.sql`



Note: If you observe any issues when executing the mentioned sql files, use the script terminator based on the database client. If your database client shows errors for `acer_scripts_dbname.sql`, create the procedures one after the other.

Run the following scripts against the Unica Campaign database in the order listed.

1. `acer_indexes_dbname.sql`

Make sure that you allow sufficient time for the script to complete. The time depends on the volume of data that is stored in the Unica Deliver system tables.

2. `acer_tables_dbname.sql`

This script creates the delta processing staging tables in the Unica Deliver system schema.

3. `acer_scripts_dbname.sql`



Important: For DB2® databases, change the termination character from ; (semicolon) to ! (exclamation point).

This script creates the stored procedures that you must configure after you install reports for Unica Deliver.

4. Navigate to `Campaign\reports\tools` directory, under your Campaign installation and locate the following scripts.

- `uare_lookup_create_DB_type.sql`

- `uare_lookup_populate*.sql`: Run the script against your Unica Campaign system tables database for all languages.



Note: Superset supports four characters for locale. For example, in the `Unica_home\Campaign\reports\tools` directory, if there are two lookup populate SQL files for the french locale:

- `uare_lookup_populate_fr.sql`
- `uare_lookup_populate_fr_FR.sql`

you must execute `uare_lookup_populate_fr_FR.sql` and ignore the other file.

What to do next



Note: You must configure the stored procedures to run on a regular basis to populate the staging tables. You cannot see data in the reports until you run the stored procedures for Unica Deliver reports.

For more information about running and scheduling the stored procedures, see [For Unica Deliver only: How to schedule and run stored procedures on page 24](#).

For Unica Deliver only: How to schedule and run stored procedures

Unica Deliver reports use the data that is contained in staging tables, which are populated by stored procedures. The stored procedures perform a delta refresh operation. Run the stored procedures at least once per day. If you run the procedures more frequently, the delta refresh method prevents multiple concurrent runs.

The following table provides information about the stored procedures and the tasks that they complete:

Table 9. Stored procedures for Deliver

Stored procedure	Task
sp_runid	Creates a unique run identifier. The list of the run IDs is stored in the UARE_RUNS table.
sp_update_ucc_tables_stats	Updates statistics for the ucc_* tables. You can run this script before the sp_populate_* scripts.
sp_populate_mailing_contacts	Processes the mailing contact data that is received since the previous run of stored procedures.
sp_populate_mailing_responses	Processes the mailing response data that is received since the previous run of stored procedures.
sp_populate_sms_contacts	If the SMS feature is enabled: Processes the SMS contact data that is received since the previous run of stored procedures.
sp_populate_sms_responses	If the SMS feature is enabled: Processes the SMS response data that is received since the previous run of stored procedures.
sp_populate_WhatsApp_contacts	If the WhatsApp feature is enabled: Processes the WhatsApp contact data that is received since the previous run of stored procedures.
sp_populate_WhatsApp_responses	If the WhatsApp feature is enabled: Processes the WhatsApp response data that is received since the previous run of stored procedures.
sp_get_delta_mailing_contacts	Called internally by sp_populate_mailing_contacts procedure. Responsible for retrieving the mailing contacts that were sent since the previous run of the stored procedures.
sp_generate_mailing_contacts	Called internally by the sp_populate_mailing_contacts procedure. Responsible for retrieving the mailing and link level counts on contacted customers for the mailings that were run since the previous run of the stored procedures.
sp_get_delta_mailing_responses	Called internally by sp_populate_mailing_responses procedure. Responsible for retrieving the responses that were received since the previous run of the stored procedures.

Table 9. Stored procedures for Deliver**(continued)**

Stored procedure	Task
sp_generate_mailing_responses	Called internally by sp_populate_mailing_responses procedure. Responsible for retrieving mailing and link level responses since the previous run of the stored procedures.
sp_get_delta_sms_contacts	Called internally by sp_populate_sms_contacts procedure. Responsible for retrieving SMS since the previous run of the stored procedures.
sp_generate_sms_contacts	Called internally by sp_populate_sms_contacts procedure. Responsible for retrieving the mailing and link level counts on contacted customers since the previous run of the stored procedures.
sp_get_delta_sms_responses	Called internally by sp_populate_sms_responses procedure. Responsible for retrieving SMS responses since the previous run of the stored procedures.
sp_generate_sms_responses	Called internally by sp_populate_sms_responses procedure. Responsible for retrieving the mailing and link level SMS responses since the previous run of the stored procedures.
sp_get_delta_WhtsApp_contacts	Called internally by sp_populate_WhtsApp_contacts procedure. Responsible for retrieving WhatsApp messages since the previous run of the stored procedures.
sp_generate_WhtsApp_contacts	Called internally by sp_populate_WhtsApp_contacts procedure. Responsible for retrieving the mailing and link level counts on contacted customers since the previous run of the stored procedures.
sp_get_delta_WhtsApp_responses	Called internally by sp_populate_WhtsApp_responses procedure. Responsible for retrieving WhatsApp responses since the previous run of the stored procedures.
sp_generate_WhtsApp_responses	Called internally by sp_populate_WhtsApp_responses procedure. Responsible for retrieving the mailing and link level WhatsApp responses since the previous run of the stored procedures.
sp_populate_mobile_responses	Processes the mobile response data that was received since the previous run of stored procedures.
sp_get_delta_mobile_responses	Called internally by sp_populate_mobile_responses procedure. Responsible for retrieving the responses that were received since the previous run of the stored procedures.

Table 9. Stored procedures for Deliver

(continued)

Stored procedure	Task
sp_generate_mobile_responses	Called internally by sp_populate_mobile_responses procedure. Responsible for retrieving mobile responses since the previous run of the stored procedures.

Guidelines for running stored procedures

Use the following guidelines when you run the stored procedures:

- You must create the stored procedures for your database by using the scripts that are provided with the installation files.
- Consider the size of the tables and indexes in your installation. Larger tables require more time to update. Allow sufficient time to process the contact and response data. The initial runs are likely to require more time to complete than subsequent runs.
- Because the stored procedures can run for an extended amount of time, consider running the procedures at times of reduced system activity, such as overnight.
- You can reduce the amount of the time that is required to refresh the reports data by limiting the scope of the reports data processed.
- You must schedule the following procedures to run at least 10 minutes after scheduling sp_runid:
 - sp_populate_mailing_contacts
 - sp_populate_mailing_responses
 - sp_populate_sms_contacts
 - sp_populate_sms_responses
 - sp_populate_WhtsApp_contacts
 - sp_populate_WhtsApp_responses
 - sp_populate_mobile_responses

When the scripts have run successfully, they display a final return code of 0.

Sample configuration of stored procedures for Oracle

Use the following guidelines when you configure stored procedures for the Oracle database.

Guidelines for configuring stored procedures

- recommends using Oracle Automatic Memory Management (AMM). For more information, go to http://docs.oracle.com/cd/B28359_01/server.111/b28310/memory003.htm.
- Create stored procedures by using a database utility, such as SQL Plus.
- Schedule the sp_runid procedure to run at least 10 minutes before the other scripts.

Example for creating a run identifier

The following example illustrates how to create a job and generate a run identifier. The example also illustrates the job ID when the job completes.

The example shows how to get a job number every day at 21:00 hours without an end date. The jobs start on November 29, 2014.

```
declare
jobno number;

BEGIN
DBMS_JOB.submit (job =>:jobno,
what => 'sp_runid;',
next_date => to_date('29-Nov-2014 21:00','DD-MON-YYYY HH24:MI' ),
interval => 'sysdate+1');
commit;
END;
/
```

Example for processing email contact data

The following example shows how to schedule a batch job to process contact data. The job runs at 21:10 hours every day.

```
declare
jobno number;

BEGIN
DBMS_JOB.submit (job =>:jobno,
what => 'sp_populate_mailing_contacts;',
next_date => to_date('29-Nov-2014 21:10','DD-MON-YYYY HH24:MI' ),
interval => 'sysdate+1');
commit;
END;
/
```

Example for processing email response data

The following example shows how to schedule a batch job to process response data. The job runs at 21:10 hours every day.

```
declare
jobno number;

BEGIN
DBMS_JOB.submit (job =>:jobno,
what => 'sp_populate_mailing_responses;',
next_date => to_date('29-Nov-2014 21:10','DD-MON-YYYY HH24:MI' ),
interval => 'sysdate+1');
commit;
END;
/
```

Example for processing SMS contact data

! **Important:** The SMS feature is not a part of the default Reports offering, and you must buy a license for the feature separately. However, the delta placement takes place regardless of whether you bought the SMS feature.

The following example shows how to get a job number every day at 21:00 hours without an end date. The jobs start on November 29, 2014.

```
BEGIN
DBMS_JOB.submit (job =>:jobno,
what => 'sp_populate_SMS_contacts;',
next_date => to_date('29-Nov-2014 21:10','DD-MON-YYYY HH24:MI' ),
interval => 'sysdate+1');
commit;
END;
/
```

Example for processing SMS response data

The following example shows how to get a job number every day at 21:00 hours without an end date. The jobs start on November 29, 2014.

```
BEGIN
DBMS_JOB.submit (job =>:jobno,
what => 'sp_populate_SMS_responses;',
next_date => to_date('29-Nov-2014 21:10','DD-MON-YYYY HH24:MI' ),
interval => 'sysdate+1');
commit;
END;
/
```

Example for processing WhatsApp contact data

! **Important:** The WhatsApp feature is not a part of the default Reports offering, and you must buy a license for the feature separately. However, the delta placement takes place regardless of whether you bought the WhatsApp feature.

The following example shows how to get a job number every day at 21:00 hours without an end date. The jobs start on November 29, 2014.

```
BEGIN
DBMS_JOB.submit (job =>:jobno,
what => 'sp_populate_WhatsApp_Contacts;',
next_date => to_date('29-Nov-2014 21:10','DD-MON-YYYY HH24:MI' ),
interval => 'sysdate+1');
commit;
END;
/
```

Example for processing WhatsApp response data

The following example shows how to get a job number every day at 21:00 hours without an end date. The jobs start on November 29, 2014.

```
BEGIN
DBMS_JOB.submit (job =>:jobno,
what => 'sp_populate_WhtsApp_Responses;',
next_date => to_date('29-Nov-2014 21:10','DD-MON-YYYY HH24:MI' ),
interval => 'sysdate+1');
commit;
END;
/
```

Example for processing Mobile response data

The following example shows how to get a job number every day at 21:00 hours without an end date. The jobs start on November 29, 2014.

```
BEGIN
DBMS_JOB.submit (job =>:jobno,
what => 'sp_populate_mobile_Responses;',
next_date => to_date('29-Nov-2014 21:10','DD-MON-YYYY HH24:MI' ),
interval => 'sysdate+1');
commit;
END;
/
```

Sample configuration of stored procedures for Microsoft™ SQL Server

Use the following guidelines when you configure stored procedures for the Microsoft™ SQL Server database.

Guidelines for configuring stored procedures

- Use the SQL Server Agent to create new jobs for each stored procedure.
- Schedule the jobs to run at least daily. You must schedule sp_runid to run at least 10 minutes before the other scripts.
- For each job in the SQL Server Agent interface, you must specify the step type as Transact-SQL script (T-SQL) and select the Unica Campaign database.

Example for creating a run identifier

The following example shows how to create a run identifier.

```
DECLARE @return_value int
EXEC @return_value = [dbo].[SP_RUNID]
SELECT 'Return Value' = @return_value
GO
```

Example for processing email contact data

The following example shows how to process email contact data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
DECLARE @return_value int
EXEC @return_value = [dbo].[SP_POPULATE_MAILING_CONTACTS]
SELECT 'Return Value' = @return_value
GO
```

Example for processing email response data

The following example shows how to process email response data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
DECLARE @return_value int
EXEC @return_value = [dbo].[SP_POPULATE_MAILING_RESPONSES]
SELECT 'Return Value' = @return_value
GO
```

Example for processing SMS contact data

The following example shows how to process SMS contact data.

```
DECLARE @return_value int
EXEC @return_value = [dbo].[SP_POPULATE_SMS_CONTACTS]
SELECT 'Return Value' = @return_value
GO
```

Example for processing SMS response data

The following example shows how to process SMS response data.

```
DECLARE @return_value int
EXEC @return_value = [dbo].[SP_POPULATE_SMS_RESPONSES]
SELECT 'Return Value' = @return_value
GO
```

Example for processing WhatsApp contact data

The following example shows how to process WhatsApp contact data.

```
DECLARE @return_value int
EXEC @return_value = [dbo].[SP_POPULATE_WHTSAPP_CONTACTS]
SELECT 'Return Value' = @return_value
GO
```

Example for processing WhatsApp response data

The following example shows how to process WhatsApp response data.

```
DECLARE @return_value int
EXEC @return_value = [dbo].[SP_POPULATE_WHTSAPP_RESPONSES]
SELECT 'Return Value' = @return_value
GO
```

Example for processing Mobile response data

The following example shows how to process Mobile response data.

```

DECLARE @return_value int
EXEC @return_value = [dbo].[SP_POPULATE_MOBILE_RESPONSES]
SELECT 'Return Value' = @return_value
GO

```

Sample configuration of stored procedures for DB2

Use the following guidelines when you configure stored procedures for the DB2® database.

Guidelines for configuring stored procedures

- The database must be DB2® version 9.7.8 or higher.
- Create new jobs in DB2® Administrative Task Scheduler (ATS)
- Schedule the jobs to run at least daily. You must schedule sp_runid to run at least 10 minutes before the other scripts.

Example for creating a run identifier

The following example shows how to get a job number every day at 20:50 hours without an end date.

```

call SYSPROC.ADMIN_TASK_ADD('RunID_Job',null,null,
null,'50 20 * * *','USER1','SP_RUNID',null,null,null)

```

Example for processing email contact data

The following example shows how to schedule a batch job to process contact data. In this example, the job runs at 21:00 hours every day. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```

call SYSPROC.ADMIN_TASK_ADD('Email_Contact_Job',null,null,null,'00 21 * * *',
'USER1','SP_POPULATE_MAILING_CONTACTS',null,null,null)

```

Example for processing email response data

The following example shows how to schedule a batch job to process response data. In this example, the job runs at 21:00 hours every day. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```

call SYSPROC.ADMIN_TASK_ADD('Email_Response_Job',null,null,
null,'00 21 * * *','USER1','SP_POPULATE_MAILING_RESPONSES',null,
null,null)

```

Example for processing SMS contact data

The following example shows how to schedule a batch job to process contact data. In this example, the job runs at 21:00 hours every day. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```

call SYSPROC.ADMIN_TASK_ADD('SMS_Contact_Job',null,null,null,'00 21 * * *',
'USER1','SP_POPULATE_SMS_CONTACTS',null,null,null)

```

Example for processing SMS response data

The following example shows how to schedule a batch job to process response data. In this example, the job runs at 21:00 hours every day.

```
call SYSPROC.ADMIN_TASK_ADD('SMS_Response_Job',null,null,
null,'00 21 * * *','USER1','SP_POPULATE_SMS_RESPONSES',null,
null,null)
```

Example for processing WhatsApp contact data

The following example shows how to schedule a batch job to process contact data. In this example, the job runs at 21:00 hours every day. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
call SYSPROC.ADMIN_TASK_ADD('WHTSAPP_Contact_Job',null,null,null,'00 21 * * *',
'USER1','SP_POPULATE_WHTSAPP_CONTACTS',null,null,null)
```

Example for processing WhatsApp response data

The following example shows how to schedule a batch job to process response data. In this example, the job runs at 21:00 hours every day.

```
call SYSPROC.ADMIN_TASK_ADD('WhtsApp_Response_Job',null,null,
null,'00 21 * * *','USER1','SP_POPULATE_WHTSAPP_RESPONSES',null,
null,null)
```

Example for processing Mobile response data

The following example shows how to schedule a batch job to process response data. In this example, the job runs at 21:00 hours every day.

```
call SYSPROC.ADMIN_TASK_ADD('Mobile_Response_Job',null,null,
null,'00 21 * * *','USER1','SP_POPULATE_MOBILE_RESPONSES',null,
null,null)
```

Granting permissions for stored procedures for DB2

Before you configure stored procedures for DB2®, you must grant permissions.

About this task

To grant permissions, complete the following steps.

1. Enable the registry by completing the following steps:
 - a. Set the **DB2_ATS_ENABLE** registry variable to one of the following values:
 - **YES**
 - **TRUE**
 - **1**
 - **ON**
 - b. Restart the DB2® database after you set the variable.
2. Create the `SYSTOOLSPACE` table space.

Users who belong to the SYSADM or SYSCTRL group can create this space. Use the following query to verify that the space exists:


```
SELECT TBSPACE FROM SYSCAT.TABLESPACES WHERE TBSPACE = 'SYSTOOLSPACE'
```

3. Grant permissions. In the following examples, substitute the values that are appropriate for your environment.
 - Deliver: Database that contains the Unica Deliver system tables
 - USER1: Owner of the Deliver database
 - DB2ADMIN: DB2@ administrative user
 - Administrator: Super user
4. Connect to DB2@ as an administrative user and run the following grant commands:
 - db2 GRANT DBADM ON DATABASE TO USER DB2ADMIN
 - db2 GRANT DBADM ON DATABASE TO USER USER1
 - db2 grant all on table SYSTOOLS.ADMINTASKS to USER1
 - db2 grant all on table SYSTOOLS.ADMINTASKS to DB2ADMIN
5. If the SYSPROC.ADMIN_TASK_ADD table exists, run the following grant commands:
 - db2 grant execute on procedure SYSPROC.ADMIN_TASK_ADD to USER1
 - db2 grant execute on procedure SYSPROC.ADMIN_TASK_ADD to DB2ADMIN

Sample configuration of stored procedures for MariaDB

Use the following guidelines when you configure stored procedures for the MariaDB database.

Guidelines for configuring stored procedures

Use the MariaDB events to create new jobs for each stored procedure.

- Schedule the jobs to run at least daily. You must schedule `sp_runid` to run at least 10 minutes before the other scripts.
- Create events for Unica Campaign database.

Example for creating a run identifier

The following example shows how to create a run identifier.

```
CREATE EVENT SP_RUNS
ON SCHEDULE EVERY 1 DAY
STARTS '2020-11-20 20:30:00'
DO
CALL SP_RUNID();
```

Example for processing email contact data

The following example shows how to process email contact data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
CREATE EVENT EMAIL_CONTACT
ON SCHEDULE EVERY 1 DAY
STARTS '2020-11-20 20:40:00'
DO
CALL SP_POPULATE_MAILING_CONTACTS();
```

Example for email response data

The following example shows how to process email response data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
CREATE EVENT EMAIL_RESPONSE
ON SCHEDULE EVERY 1 DAY
STARTS '2020-11-20 20:40:00'
DO
CALL SP_POPULATE_MAILING_RESPONSES();
```

Example for processing SMS contact data

The following example shows how to process SMS contact data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
CREATE EVENT SMS_CONTACT
ON SCHEDULE EVERY 1 DAY
STARTS '2020-11-20 20:40:00'
DO
CALL SP_POPULATE_SMS_CONTACTS();
```

Example for processing SMS response data

The following example shows how to process SMS response data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
CREATE EVENT SMS_RESPONSE
ON SCHEDULE EVERY 1 DAY
STARTS '2020-11-20 20:40:00'
DO
CALL SP_POPULATE_SMS_RESPONSES();
```

Example for processing WhatsApp contact data

The following example shows how to process WhatsApp contact data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
CREATE EVENT WHTSAPP_CONTACT
ON SCHEDULE EVERY 1 DAY
STARTS '2020-11-20 20:40:00'
DO
CALL SP_POPULATE_WHTSAPP_CONTACTS();
```

Example for processing WhatsApp response data

The following example shows how to process WhatsApp response data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
CREATE EVENT WHTSAPP_RESPONSE
ON SCHEDULE EVERY 1 DAY
STARTS '2020-11-20 20:40:00'
```

```
DO
CALL SP_POPULATE_WHTSAPP_RESPONSES();
```

Example for processing Mobile response data

The following example shows how to process Mobile response data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
CREATE EVENT MOBILE_RESPONSE
ON SCHEDULE EVERY 1 DAY
STARTS '2020-11-20 20:40:00'
DO
CALL SP_POPULATE_MOBILE_RESPONSES();
```

Sample configuration of stored procedures for OneDB

Use the following guidelines when you configure stored procedures for the OneDB database.

Guidelines for configuring stored procedures

Use the OneDB tasks to create new jobs for each stored procedure. These tasks are created using sysadmin database.

- Run following command before initializing the Deliver database.

```
Set environment the DB_LOCALE and GL_USEGLS = 1
```

- Schedule the jobs to run at least daily. You must schedule sp_runid to run at least 10 minutes before the other scripts.
- Create tasks in Sysadmin database.

Example for creating a run identifier

The following example shows how to create a task to generate a run identifier.

```
INSERT INTO ph_task
(
tk_name,tk_description,tk_type,tk_execute,tk_start_time,tk_frequency, tk_attributes
)
VALUES
(
'SP_RUNS',
'This task is to invoke procedure to generate runids for the deliver delta refresh runs.',
'TASK',
'EXECUTE PROCEDURE < Deliver_Database name>@<DB Server Instance>:SP_RUNID()',
'20:30:00','1 0:00:00', 0
);
```

Example for processing email contact data

The following example shows how to process email contact data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
INSERT INTO ph_task
(
tk_name,tk_description,tk_type,tk_execute,tk_start_time,tk_frequency, tk_attributes
)
```

```
VALUES
(
'MAILING_CONTACTS',
'This task is to invoke procedure to populate mailing contacts.',
'TASK',
'EXECUTE PROCEDURE < Deliver_Database_name>@<DBServer_Instance>:SP_POPULATE_MAILING_CONTACTS()',
'20:40:00','1 0:00:00', 0
);
```

Example for email response data

The following example shows how to process email response data. Schedule the job to run atleast 10 minutes after the job that generates the run identifier.

```
INSERT INTO ph_task
(
tk_name,tk_description,tk_type,tk_execute,tk_start_time,tk_frequency, tk_attributes
)
VALUES
(
'MAILING_RESPONSES',
'This task is to invoke procedure to populate mailing responses.',
'TASK',
'EXECUTE PROCEDURE < Deliver_Database_name>@<DBServer_Instance>:SP_POPULATE_MAILING_RESPONSES()',
'20:40:00','1 0:00:00', 0
);
```

Example for processing SMS contact data

The following example shows how to process SMS contact data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
INSERT INTO ph_task
(
tk_name,tk_description,tk_type,tk_execute,tk_start_time,tk_frequency, tk_attributes
)
VALUES
(
'SMS_CONTACTS',
'This task is to invoke procedure to populate SMS contacts.',
'TASK',
'EXECUTE PROCEDURE < Deliver_Database_name>@<DBServer_Instance>:SP_POPULATE_SMS_CONTACTS()',
'20:40:00','1 0:00:00', 0
);
```

Example for processing SMS response data

The following example shows how to process SMS response data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```

INSERT INTO ph_task
(
tk_name,tk_description,tk_type,tk_execute,tk_start_time,tk_frequency, tk_attributes
)
VALUES
(
'SMS_RESPONSES',
'This task is to invoke procedure to populate SMS responses.',
'TASK',
'EXECUTE PROCEDURE <Deliver_Database_name>@<DBServer_Instance>:SP_POPULATE_SMS_RESPONSES()',
'20:40:00','1 0:00:00', 0
);

```

To check scheduled tasks creation in sysadmin database, run the following query.

```
SELECT * from ph_task;
```

To check run status, check in sysadmin database and run the following query.

```
select * from ph_run;
```

You can also check in each channel process data in Deliver database using following query.

```
Select * from uare_delta_refresh_log order by runid desc;
```

Example for processing WhatsApp contact data

The following example shows how to process WhatsApp contact data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```

INSERT INTO ph_task
(
tk_name,tk_description,tk_type,tk_execute,tk_start_time,tk_frequency, tk_attributes
)
VALUES
(
'WHTSAPP_CONTACTS',
'This task is to invoke procedure to populate WhatsApp contacts.',
'TASK',
'EXECUTE PROCEDURE < Deliver_Database_name>@<DBServer_Instance>:SP_POPULATE_WHTSAPP_CONTACTS()',
'20:40:00','1 0:00:00', 0
);

```

Example for processing WhatsApp response data

The following example shows how to process WhatsApp response data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```

INSERT INTO ph_task
(
tk_name,tk_description,tk_type,tk_execute,tk_start_time,tk_frequency, tk_attributes
)
VALUES

```

```
(
'WHTSAPP_RESPONSES',
'This task is to invoke procedure to populate WhatsApp responses.',
'TASK',
'EXECUTE PROCEDURE <Deliver_Database_name>@<DBServer_Instance>:SP_POPULATE_WHTSAPP_RESPONSES()',
'20:40:00','1 0:00:00', 0
);
```

To check scheduled tasks creation in sysadmin database, run the following query.

```
SELECT * from ph_task;
```

To check run status, check in sysadmin database and run the following query.

```
select * from ph_run;
```

You can also check in each channel process data in Deliver database using following query.

```
Select * from uare_delta_refresh_log order by runid desc;
```

Example for processing Mobile response data

The following example shows how to process Mobile response data. Schedule the job to run at least 10 minutes after the job that generates the run identifier.

```
INSERT INTO ph_task
(
tk_name,tk_description,tk_type,tk_execute,tk_start_time,tk_frequency, tk_attributes
)
VALUES
(
'MOBILE_RESPONSES',
'This task is to invoke procedure to populate Mobile responses.',
'TASK',
'EXECUTE PROCEDURE <Deliver_Database_name>@<DBServer_Instance>:SP_POPULATE_MOBILE_RESPONSES()',
'20:40:00','1 0:00:00', 0
);
```

To check scheduled tasks creation in sysadmin database, run the following query.

```
SELECT * from ph_task;
```

To check run status, check in sysadmin database and run the following query.

```
select * from ph_run;
```

You can also check in each channel process data in Deliver database using following query.

```
Select * from uare_delta_refresh_log order by runid desc;
```

Known limitations

The following is the list of Known Limitations in 12.1.9 of Superset Reports

Issue ID	Issue Description
HMA-375391	If there is a lot of data in reports and if you open the plan analytics report, you may not see the updated data. Manually refresh the report to view the latest data.

Known issues

The following is the list of Known Issues in 12.1.9 of Superset Reports

Issue ID	Issue Description
HMA-375850	Sometimes reports are opening with errors.
HMA-375827	For DB2 database, there is a data mismatch and offer given values axis is not available in the Dashboard report Offer responses for the last 7 days.
HMA-375493	For Unica Deliver on Oracle, incorrect Unique Click Rate appears in Message Overview reports.
HMA-374720	The discard button in embeded Superset dashboard will not work. You have to reload the Platform dashboard again.
HMA-375384	The columns Best Offer Lift Over This and Lift Over Worst Offer are appearing for the following reports: <ul style="list-style-type: none"> • Campaign performance comparison • Campaign performance comparison (with revenue) • Campaign performance summary by cell and offer • Campaign performance summary by cell and offer (with revenue) • Campaign performance summary by cell (with revenue) • Campaign performance summary by offer • Offer performance summary by campaign • Campaign performance summary by cell
HMA-375378	The Filter and Share option in reports is visible even if it is not applicable. You can ignore it.
HMA-375374	When you download a report, the option Export as Excel exists. If you select the option, the system displays a JSON file. Currently, we do not support exporting to Excel format. Do not use the option.
HMA-375292	In case of tooltips, if you hover the mouse over the long URL, the URL gets overlapped on the Alias column.
HMA-375102	Add one more filter for the Campaign details in the Campaign detailed offer response breakout report.

Issue ID	Issue Description
HMA-375087 / HMA-375086	In Message Overview Reports, incorrect values are seen for % Unique View Rate, % Unique Click Rate, and Open (%) in the downloaded CSV and Excel Files.
HMA-375078	In the Offer Campaign Listings report, the Initiative column is blank. There must at least be a hyphen or NA instead of being blank.
HMA-374996	The Superset report sequence is not displaying as expected in Campaign reports page, particularly with Unica Insights reports.
HMA-374855	What if offer financial summary report not available.
HMA-374645	An unexpected error appears in the Campaign Performance Summary by Offer report.
HMA-374615	No Data returned in Offer Campaign Listings table.
HMA-374288	4K big number chart not visible for A/B Testing Performance Report on Unica page for Superset Report. Workaround: You can open the Chart using Superset Interface, load the data, and then go to Unica and reload it. It should work.
HMA-374286	Pie chart not visible on Unica page and getting error while opening Detailed Bounce Report in Unica. Workaround: You can open the Chart using Superset Interface, load the data, and then go to Unica and reload it. It should work.
HMA-373903	Titles missing in Campaign Performance Summary By Offer Report as compared to Campaign report.
HMA-373637	If Superset is implemented on a DB2 environment, you will see an error when opening a few reports.
HMA-371498	The filters for campaign code and run date (ranging from 1 day to monthly, quarterly, and yearly) are not added in all performance reports (Email, SMS, Whatsapp, and Push).
HMA-365608	On Messaging Overview reports, mailing names does not display in graph for every alternate mailings.
HMA-356140	The sql script <code>uare_lookup_create_Oracle.sql</code> located in <code>\$CAMPAIGN_HOME/Campaign/reports/tools</code> contains an empty line after the open bracket of the <code>CREATE TABLE</code> statement, causing an error when run with Oracle sqlplus.