HCLSoftware

Unica Campaign V12.1.3 Installation Guide

Contents

Chapter 1. Installation overview

An installation of HCL Unica products is complete when you install, configure, and deploy the HCL Unica products . The Installation Guide provides detailed information about installing, configuring, and deploying the products.

Use the Installation Roadmap section to obtain a broad understanding about using the Installation Guide.

From version 12.0 - Unica Campaign installer installs Unica Optimize. It gets installed by default with the installation of Unica Campaign. Although Unica Optimize gets installed by default with Unica Campaign, you require to install license to run Optimize sessions. Contact the Technical Support team to get Optimize licenses.

In version 12.1, Unica Campaign listener supports Optimize Session execution along with Campaign flowchart execution or both (Optimize session execution and flowchart execution both). While installing Campaign Analytical Server (Listener), users can choose the type of listener to install.

- Run Optimize session only
- Run Campaign flowcharts
- Run Optimize Session and Campaign flowcharts

Installation roadmap

Use the installation roadmap to quickly find the information that you need for installing Unica Campaign and Unica Optimize.

You can use Table 1: Unica Campaign installation roadmap on page 5 to scan the tasks that must be completed for installing Unica Campaign and Unica Optimize. The **Information** column in the following table provides links to the topics that describe the tasks for installing Unica Campaign and Unica Optimize.

Table 1. Unica Campaign installation roadmap

Chapter	Information
Unica Campaign web application clustering on page 163	If you are using web application clustering, review this appendix before starting your installation.
Installing a clustered listener environment on page 173	If you are using Unica Campaign listener clustering, review this appendix before starting your installation.
Installation Overview on page 5	This topic provides the following information: • How the installers work on page 8 • Modes of installation on page 8

Table 1. Unica Campaign installation roadmap

(continued)

Chapter	Information
Planning the Unica Campaign and Unica Optimize installation on page 10	This topic provides the following information: • Prerequisites on page 10 • Unica Campaign installation worksheet on page 13 • Installation order for Unica products on page 15 • Planning failover configuration of Unica Campaign on page 16
Preparing data sources for Unica Campaign on page 18	This topic provides the following information: • Creating database or schema for Unica Campaign system tables on page 18 • Creating ODBC or native connections on page 19 • Configuring the web application server for your JDBC driver on page 20 • Creating JDBC connections in web application server on page 23 • Preparing DB2 on z/OS user datasource for Unica Campaign on page 37 • Preparing Amazon Redshift user data sources for Unica Campaign on page 38 • Preparing Apache Hadoop Hive user data sources for Unica Campaign on page 40
Installing Unica Campaign and Unica Optimize on page 73	This topic provides the following information: • Installing Unica Campaign and Unica Optimize by using the GUI mode on page 74 • Installing Unica Campaign and Unica Optimize using the console mode on page 83 • Installing Unica Campaign and Unica Optimize silently on page 84
Configuring Unica Campaign and Unica Optimize before deployment on page 87	This topic provides the following information: • Creating and populating Unica Campaign and Unica Optimize system tables manually on page 87 • Registering Unica Campaign manually on page 91 • Setting data source variables in the Unica Campaign startup script (UNIX only) on page 92

Table 1. Unica Campaign installation roadmap

(continued)

Chapter	Information
Deploying the Unica Campaign web application on page 97	This topic provides the following information: • Setting the web application session timeout on page 102 • Deploying Unica Campaign on WebSphere Application Server on page 102 • Deploying Unica Campaign on WebLogic on page 105 • Deploying Unica Campaign on JBoss Application Server on page 106 • Deploying Unica Campaign on Tomcat Application Server on page 107 • Starting the Unica Campaign server on page 109
Configuring Unica Campaign and Unica Optimize after deployment on page 111	This topic provides the following information: • Verifying that the Unica Campaign listener is running on page 112 • Setting up the Unica Campaign system user on page 113 • Adding data source properties on the Configuration page on page 113 • Unica Campaign configuration properties on page 115 • Mapping user tables in Unica Campaign on page 116 • Verifying the Unica Campaign installation on page 119 • Verifying the Unica Optimize installation on page 119 • Setting properties for integration with Unica products on page 120
Configuring multiple partitions for Unica Campaign on page 123	This topic provides the following information: • How multiple partitions work on page 123 • Setting up multiple partitions on page 123 • Assigning roles, permissions, and groups for partitions on page 128
Configuring multiple partitions for Unica Optimize on page 129	This topic provides the following information: • How multiple partitions work on page 123 • Setting up multiple partitions for Unica Optimize on page 129
Configuring multiple partitions in Unica Deliver on page 131.	This topic provides the following information: • Partitions for Unica Deliver: Overview on page 131 • Roadmap for configuring multiple partitions in Unica Deliver on page 132 • Creating a new partition for Unica Deliver on page 132

Table 1. Unica Campaign installation roadmap

(continued)

Chapter	Information
	This topic provides the following information:
	This topic provides information about how to uninstall Unica Campaign and Unica Optimize.

How the installers work

You must use the suite installer and the product installer when you install or upgrade any Unica product. For example, for installing Unica Plan, you must use the Unica suite installer and the Unica Plan installer.

Make sure that you use the following guidelines before you use the Unica suite installer and the product installer:

- The Unica installer and the product installer must be in the same directory on the computer where you want to install the product. When multiple versions of a product installer are present in the directory with the Unica installer, the Unica installer always shows the latest version of the product on the Unica Products screen in the installation wizard.
- If you are planning to install a patch immediately after you install an Unica product, make sure that the patch installer is in the same directory as that of the suite and product installers.
- The default top-level directory for Unica installations is `/HCL/Unica` for UNIX™ or `C:\HCL\Unica` for Windows™. However, you can change the directory during installation.

Modes of installation

The Unica suite installer can run in one of the following modes: GUI mode, X Window System mode, console mode, or silent mode (also called the unattended mode). Select a mode that suits your requirements when you install Unica Campaign.

For upgrades, you use the installer to perform many of the same tasks that you perform during the initial installation.

GUI X Window System mode

Use the GUI mode for Windows™ or the X Window System mode for UNIX™ to install Unica Campaign by using the graphical user interface.

UNIX™ X Window System mode

Use the X Window System mode for UNIX™ to install Unica Campaign by using the graphical user interface.

Console mode

Use the console mode to install Unica Campaign by using the command line window.

Note: To display the Installer screens correctly in console mode, configure your terminal software to support UTF-8 character encoding. Other character encoding, such as ANSI, will not render the text correctly, and some information will not be readable.

Silent mode

Use the silent or unattended mode to install Unica Campaign multiple times. The silent mode uses response files for installation and does not require user input during the installation process.

Note: Silent mode is not supported for upgrade installations in clustered web application or clustered listener environments.

Chapter 2. Planning the Unica Campaign and Unica Optimize installation

When you plan your Unica Campaign and Unica Optimize installation, you must ensure that you have set up your system correctly, and that you have configured your environment to deal with any failures.

IBM SPSS Modeler Advantage Enterprise Marketing Management Edition components

IBM SPSS Modeler Advantage Enterprise Marketing Management Edition has the following components:

- IBM® Analytical Decision Management including Modeler Advantage
- IBM® SPSS® Modeler Server Professional (without Modeler Client)
- IBM® SPSS® Collaboration and Deployment Services (C&DS)

Prerequisites

Before you install or upgrade any Unica product, you must ensure that your computer complies with all of 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 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.

JVM requirements

Unica applications within a suite must be deployed on a dedicated Java™ virtual machine (JVM). Unica products customize the JVM that is used by the web application server.

Knowledge requirements

To install 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.

Internet browser settings

Make sure that your internet browser complies with the following settings:

- The browser must not cache web pages.
- The browser must not block pop-up windows.

Access permissions

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

- Administration access for all necessary databases

  📝 **Note:** Administrator must have `CREATE`, `SELECT`, `INSERT`, `UPDATE`, `DELETE`, and `DROP` rights for both tables and views.

- 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 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 have the administrative password for your web application server.

For UNIX™, all installer files for products must have full permissions, for example, rwxr-xr-x.

The following additional permissions are necessary for UNIX™:

- The user account that installs Unica Plan 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 Unica products must have full permissions, for example, rwxr-xr-x.

Points to consider before you install Unica Plan

For Unica Plan installation you are required to consider the following points.

JAVA_HOME environment variable

If a JAVA_HOME environment variable is defined on the computer where you install an Unica product, verify that the variable points to a supported version of JRE. For information about system requirements, see the *Unica Recommended Software Environments and Minimum System Requirements* guide.

If the JAVA_HOME environment variable points to an incorrect JRE, you must clear the JAVA_HOME variable before you run the Unica installers.

You can clear the JAVA_HOME environment variable by using one of the following methods:

- Windows™: In a command window, enter set JAVA_HOME= (leave empty) and press **Enter**.
- UNIX™: In the terminal, enter export JAVA_HOME= (leave empty) and press **Enter**.

You can clear the JAVA_HOME environment variable by running the following command in the terminal:

export JAVA_HOME= (leave empty)

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

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

Unica Platform requirements

You must install or upgrade Unica Platform before you install or upgrade any Unica products. For each group of products that work together, you must install or upgrade Unica Platform only once. Each product installer checks whether the required products are installed. If your product or version is not registered with Unica Platform, a message prompts you to install or upgrade Unica 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.

> **Note:** For installations on UNIX, you may require to set the `Djava.awt.headless` property to true in your web application server. The setting is required only when you are unable to view Unica Optimize reports. See the Unica Campaign Installation Guide for details. You do not require to prepare any additional data sources for Unica Optimize because Unica Optimize uses the Unica Campaign system tables data source.

> **Note:** For versions 12.0.0 and higher, ensure that you do not select the database type Informix as it is not functional. From version 12.1.0.3 and higher, users can use OneDB database as system tables and user tables. See the *Unica V12.1.0.3 Installation Guide for OneDB* for more details.

IBM SPSS Modeler Advantage Enterprise Marketing Management Edition replaces PredictiveInsight

If you previously used PredictiveInsight, you can no longer use the legacy Model and Score process boxes in Unica Campaign.

After you install Unica Campaign version 9.1.0 or later, the legacy Model and Score process boxes are unconfigured in existing flowcharts. You must manually re-create the predictive models either by using IBM SPSS Modeler Advantage Enterprise Marketing Management Edition or by using the SPSS Model process box.

> **Note:** Legacy Model and Score process boxes are inactive and cannot be run. However, they can be viewed for configuration details.

For more information, see the *Unica Campaign and IBM SPSS Modeler Advantage Enterprise Marketing Management Edition Integration Guide*.

Unica Campaign installation worksheet

Use the Unica Campaign installation worksheet to gather information about the Unica Campaign database and about other Unica products that are required for the installation of Unica Campaign.

Use the following table to gather information about the empty database that was created for the Unica Campaign system tables. The empty database that you set up for Unica Campaign can have any name. If you use UA_SYSTEM_TABLES as the ODBC name for that database, the Unica Campaign system tables will be mapped automatically.

Table 2. Information about the database

Field	Notes®
Database type	
Database name	
Database account user name	
Database account password	
JNDI name	UnicaPlatformDS
ODBC name	

If you are installing on UNIX™, obtain the information that is given in the following table:

Table 3. Information about the database for UNIX™

Database information	Notes®
If your database type is one of the following types, record the database installation directory: • DB2® • Oracle	
For all database types when Campaign is installed on the Solaris, Linux™, or AIX®® operating systems, record the location of the `lib` directory in the database installation.	

You can use the information in when you edit the `setenv.sh` file during the installation and configuration process.

Checklist for Unica Platform database

The installation wizards for each Unica product must be able to communicate with the Unica Platform system table database to register the product. Each time that you run the installer, you must enter the following database connection information for the Unica Platform system table database:

- JDBC connection URL
- Database host name
- Database port
- Database name or schema ID
- User name and password for the database account

Checklist for Unica Platform deployment on the web application server

Obtain the following information before deploying Unica Platform:

- Protocol: HTTP, or HTTPS if SSL is implemented in the web application server.
- Host: The name of the machine on which the Unica Platform will be deployed.
- Port: The port on which the web application server listens.
- Domain name: The company domain of each machine where HCL products are installed. For example, `example.com`. All HCL products must be installed in the same company domain, and you must enter the domain name in all lower case letters.

  If there is a mismatch in domain name entries, you might encounter problems when you attempt to use Unica Platform features or navigate among products. You can change the domain name after the products are deployed by logging in and changing values of the relevant configuration properties in the product navigation categories on the **Settings > Configuration** page.

Checklist for enabling Unica Platform utilities

If you plan to use the Unica Platform utilities, obtain the following JDBC connection information before you start to install the Unica Platform.

- Path to the JRE. The default value is the path to the 1.8 version of the JRE that the installer places under your installation directory.

  You can accept this default or specify a different path. If you specify a different path, you must point to the 1.8 version of the Sun JRE.

- JDBC driver class. The installer automatically provides this, based on the database type that you specify in the installer.
- JDBC connection URL. The installer provides the basic syntax, including the host name, database name, and port. You can customize the URL by adding additional parameters.
- JDBC driver classpath on your system.

Information about the web component

Obtain the following information for all Unica products that have a web component, which you deploy on a web application server:

- The name of the systems on which the web application servers are installed. You can have one or several web application servers, depending on the Unica environment that you are setting up.
- The port on which the application server listens. If you plan to implement SSL, obtain the SSL port.
- The network domain for your deployment system. For example, mycompany.com.

Installation order for Unica products

When you install or upgrade multiple Unica products, you must install them in a specific order.

The following table provides information about the order that you must follow when you install or upgrade multiple Unica products.

Table 4. Installation or upgrade order for Unica products

For this product or combination:	Install or upgrade in this order:
Unica Campaign (with or without Unica Deliver) and Unica Optimize	1. Unica Platform 2. Unica Campaign and Unica Optimize ✏️ **Note:** Unica Deliver is installed automatically when you install Unica Campaign. However, Unica Deliver is not configured or enabled during the Unica Campaign installation process.
Unica Interact	1. Unica Platform 2. Unica Campaign 3. Unica Interact Design Time environment 4. Unica Interact Run Time environment 5. Unica Interact Extreme Scale Server If you want to install or upgrade the Interact design time environment only, then install or upgrade the Interact design time environment in the following order: 1. Unica Platform 2. Unica Campaign 3. Unica Interact Design Time environment

Table 4. Installation or upgrade order for Unica products

(continued)

For this product or combination:	Install or upgrade in this order:
	If you want to install or upgrade the Interact runtime environment only, then install or upgrade the Interact runtime environment in the following order: 1. Unica Platform 2. Unica Interact Run Time environment If you want to install the Unica Interact Extreme Scale Server only, then install the Unica Interact Extreme Scale Server in the following order: 1. Unica Platform 2. Unica Interact Run Time environment 3. Unica Interact Extreme Scale Server
Unica Plan	1. Unica Platform 2. Unica Plan **Note:** If you are integrating Unica Plan with Unica Campaign, you must also install Unica Campaign. The order of installation for those two products does not matter.
Unica Interact Advanced Patterns	1. Unica Platform 2. Unica Campaign 3. Unica Interact 4. Unica Interact Advanced Patterns
IBM SPSS Modeler Advantage Enterprise Marketing Management Edition	1. IBM SPSS Modeler Advantage Enterprise Marketing Management Edition

Planning failover configuration of Unica Campaign

When you install Unica Campaign, you can configure your environment to address any failures of the web application server (front-end) or listener server (back-end).

Definition of terms

Required component	Installation option	Definition
Unica Campaign web application server (front end)	J2EE Application	The web application that provides the Unica Campaign user interface.
Unica Campaign listener server, sometimes called analytic server (back end)	Unica Campaign Server	The startup scripts and components that support the design and execution of flowcharts.

Unica Campaign web application server clustering

You can deploy the Unica Campaign web application server in a clustered environment to support load balancing.

Before you begin the installation, be sure to read Unica Campaign web application clustering on page 163.

Unica Campaign listener server clustering

Listener (analytic) server clustering, also referred to as back-end clustering, involves the use of multiple listeners for the purpose of high availability and load balancing. Listener clustering is important because flowchart runs happen at the back end.

Before you begin the installation, be sure to read Installing a clustered listener environment on page 173.

Chapter 3. Preparing data sources for Unica Campaign

The Unica Campaign runtime environment uses data sources to store user and interaction data.

About this task

📝 **Note:** Information about using web application clustering in Unica Campaign is provided in Unica Campaign web application clustering on page 163.

Complete the following steps to prepare data sources for Unica Campaign.

1. Create a database or a database schema for the Unica Campaign system tables. The empty database that you set up for Unica Campaign can have any name.

   📝 **Note:** Unica Campaign requires customer tables, which should already exist.

2. Create a database user account.

   The database user account must have the following rights: CREATE, DELETE, DROP, INSERT, SELECT, and UPDATE.

3. Create ODBC or native connections. If you use UA_SYSTEM_TABLES as the ODBC name for that database, the Unica Campaign system tables will be mapped automatically.
4. Configure the web application server for your JDBC driver.
5. Create JDBC data sources in the web application server.

Prerequisites of Oracle 12c or Oracle 19.x database before Unica Campaign installation

You must set the following configuration while using Oracle Database version 12.2.x and 19c.

On Oracle database server, add the following parameters in "sqlnet.ora" file located under `ORACLE_HOME/network/admin/`

`SQLNET.ALLOWED_LOGON_VERSION_SERVER`

Contact the Database Administrator to provide appropriate value to this parameter. See the following link for more details:
https://docs.oracle.com/database/121/NETRF/sqlnet.htm#NETRF2016

Oracle 19c is also supported with ODBC connectivity, User would not be required to edit the sqlnet.ora parameters as mentioned above while connecting with ODBC.

If user does not wants to use this parameter, they can use Oracle ODBC connections instead of Oracle Native connections.

Creating database or schema for Unica Campaign system tables

The Unica Campaign system tables hold application data. The empty database that you set up for the system tables can have any name. To ensure that the system tables will be mapped automatically, the ODBC name for that database must be

UA_SYSTEM_TABLES. If you have multiple partitions, create data sources for each partition where you want to enable Unica Campaign.

Complete the following steps to create a database or database schema for Unica Campaign system tables:

1. Create a system user account for later in the installation process.

   ✏️ **Note:** The system user account must have the following rights: CREATE, DELETE, DROP, INSERT, SELECT, and UPDATE.

2. Print the Unica Campaign installation worksheet on page 13. Obtain the information about your database or schema and database account, and then note this information in the checklist. Continue to fill in the checklist as you perform the rest of the steps in this section, so that you can use this information later in the installation process.

   ✏️ **Note:** The Unica Deliver system tables are created in the Unica Campaign schema during the installation. However, Unica Deliver is not enabled.

Creating ODBC or native connections

Create ODBC or native connections so that the Unica Campaign server can access the Unica Campaign database. Create ODBC or native connections on the computer where you installed the Unica Campaign server.

About this task

The computer on which the Unica Campaign server is installed must have an ODBC or native connection to the following databases:

- The database or schema that contains the Unica Campaign system tables
- The database or schema that contains the customer tables (sometimes called user tables).

Use the following guidelines to create these ODBC or native connections:

- **For databases on UNIX™/SUSE**: Create native connections for DB2® and Oracle databases or ODBC connections for other databases, such as Teradata, Netezza®. The procedure for creating a native data source differs depending on the data source type and the version of UNIX™. See the data source and operating system documentation for specific ODBC driver installation and configuration instructions.
- **For databases on Windows™**: Create a new ODBC connection in the Control Panel **Administrative Tools > Data Sources (ODBC)** section.
- **For DB2® databases on z/OS**®: The database user account must have SYSADM privileges. The DBADM privilege is not sufficient because it does not provide access to all of the required system objects. Without SYSADM privileges, flowcharts might not run successfully.
- **For Oracle ODBC**: For configuring your environment in ODBC, you will need to do following changes in environment:
  1. Install Unix ODBC driver version 2.3.7 or higher.
  2. Create System and User Datasource in odbc.ini file

a. Edit odbc.ini file

b. Add entry for system and User Datasource

c. Make sure DSN entry, Servername in ODBC.ini file should be exactly matching to the DSN entry in tnsnames.ora file

e.g.

[ORCL]

Driver = /u01/oracle19c/product/db_home/lib/libsqora.so.19.1

Servername = orcl

Port = 1521

> **Note:** For more information, see

- **For MariaDB**: Set the following parameters in MariaDB configuration (.cnf or my.ini) file for Platform and Campaign.
  - **Lower_case_table_names**: The value is 1, so that table names are considered as case insensitive.
  - **Set wait_timeout**= `<Time in seconds that the server waits for a connection to become active before closing it. The session value is initialized when a thread starts up from either the global value, if the connection is non-interactive, or from the interactive_timeout value, if the connection is interactive.>`. For example: set this to 25,92,000 (seconds), in case the setup can be inactive for 30 days.
  - **max_connections**= `<The maximum number of simultaneous client connections>`.
  - **innodb_strict_mode** = 0

> **Note:** Unica Campaign supports certain versions of DB2® on z/OS® for customer tables only, not system tables. For more information, see the *Recommended Software Environments and Minimum System Requirements*.

Be sure to record the ODBC name in the .

Configuring the web application server for your JDBC driver

The web application server where you plan to deploy Campaign must contain the correct JAR file to support JDBC connections. This enables your web application to connect to the system tables. The location of the JAR file must be included in the class path of the web application server.

If you are using WebSphere®, Weblogic, Tomcat and if you allow the installer to create the data source when you install the product, you can skip the step in this procedure about setting the classpath. To enable automatic data source creation during installation, on the Datasource Creation panel, select the **Create Campaign Datasource** check box and supply information about your application server. If you do not want to allow the installer to create the data source, then perform this entire procedure.

If you are using WebLogic application server you must perform this entire procedure.

1. Consult the *Recommended Software Environments and Minimum System Requirements* document to determine the correct JAR file for your database type.
2. Obtain the latest vendor-provided Type 4 JDBC driver that is supported by Unica.
   ◦ If the driver does not exist on the machine where you plan to deploy Unica Campaign, copy the driver to any location on that machine. Unpack the driver in a path that does not include spaces.
   ◦ If you obtain the driver from a machine where the data source client is installed, verify that the version is supported.
3. Include the full path and file name of the driver in the classpath of the web application server where you plan to deploy Unica Campaign.

   ◦ For all supported versions of WebLogic, set the classpath in the setDomainEnv script in the `WebLogic_domain_directory/bin` directory where environment variables are configured.

   The driver must be the first entry in the CLASSPATH list of values before any existing values. For example:

   UNIX™

   ```
   CLASSPATH="/home/oracle/product/<version>/jdbc/lib/ojdbc8.jar:
     ${PRE_CLASSPATH}${CLASSPATHSEP}${WEBLOGIC_CLASSPATH}
     ${CLASSPATHSEP}${POST_CLASSPATH}${CLASSPATHSEP}${WLP_POST_CLASSPATH}"
     export CLASSPATH
   ```

   Windows™

   ```
   set CLASSPATH=c:\oracle\jdbc\lib\ojdbc8.jar;%PRE_CLASSPATH%;
   %WEBLOGIC_CLASSPATH%;%POST_CLASSPATH%;%WLP_POST_CLASSPATH%
   ```

   ◦ For all supported versions of WebSphere®, set the classpath in the Administration console when you set up the JDBC providers for the Unica products.
   ◦ Please make a note of the database driver class path in the Campaign installation worksheet, as you must enter the path when you run the installer.
4. Restart the web application server so your changes take effect. During startup, monitor the console log to confirm that the classpath contains the path to the database driver.

Datasource creation during installation is not supported for JBOSS application Server, so you must perform this entire procedure.

1. Obtain the latest vendor-provided Type 4 JDBC driver for your system table database that is supported by Unica, as described in the *Recommended Software Environments and Minimum System Requirements* guide.
2. Use the following guidelines after you obtain the JDBC driver.
   ◦ If the driver does not exist on the server where you plan to deploy Campaign, obtain it and unpack it on the server. Unpack the drivers in a path that does not include spaces.
   ◦ If you obtain the driver from a server where the data source client is installed, verify that the version is the latest supported by Campaign.
3. Add and register the full path of the driver including the file name, to the module using below commands:

- For all supported versions of JBoss, add the JDBC driver as module. Use the following procedure to add the JDBC driver as a module.

  For example for SQL Server:

  ```
  module add --name=com.microsoft.sqlserver.jdbc --resources=<JDBC_Driver_Location>\sqljdbc4.jar --
  dependencies=javax.api,javax.transaction.api
  ```

- Register this SQL JDBC Driver using the following guidelines: For example:
  - ```
    /subsystem=datasources/jdbc-driver=sql:add(driver-module-
    name=com.microsoft.sqlserver.jdbc,driver-name=sql,driver-xa-datasource-class-
    name=com.microsoft.sqlserver.jdbc.SQLServerXADataSource)
    ```
  - ```
    /subsystem=datasources/jdbc-driver=sql:read-resource
    ```
  - ```
    /subsystem=ee/service=default-bindings:write-attribute(name=datasource, value=undefined)
    ```

4. Make a note of the database driver class path in the Campaign installation worksheet, as you must enter the path when you run the installer.
5. Restart the web application server so that your changes take effect. During startup, monitor the console log to confirm that the class path contains the path to the database driver.
6. Obtain the latest vendor-provided Type 4 JDBC driver for your system table database that is supported by Unica, as described in the *Recommended Software Environments and Minimum System Requirements* guide.
7. Use the following guidelines after you obtain the JDBC driver.
   - If the driver does not exist on the server where you plan to deploy Campaign, obtain it and unpack it on the server. Unpack the drivers in a path that does not include spaces.
   - If you obtain the driver from a server where the data source client is installed, verify that the version is the latest supported by Campaign.
8. Add and register the full path of the driver including the file name, to the module using below commands:
   - For all supported versions of JBoss, add the JDBC driver as module. Use the following procedure to add the JDBC driver as a module.

     For example for SQL Server:

     ```
     module add --name=com.microsoft.sqlserver.jdbc --resources=<JDBC_Driver_Location>\sqljdbc4.jar --
     dependencies=javax.api,javax.transaction.api
     ```

   - Register this SQL JDBC Driver using the following guidelines: For example:
     - ```
       /subsystem=datasources/jdbc-driver=sql:add(driver-module-
       name=com.microsoft.sqlserver.jdbc,driver-name=sql,driver-xa-datasource-class-
       name=com.microsoft.sqlserver.jdbc.SQLServerXADataSource)
       ```
     - ```
       /subsystem=datasources/jdbc-driver=sql:read-resource
       ```
     - ```
       /subsystem=ee/service=default-bindings:write-attribute(name=datasource, value=undefined)
       ```
9. Make a note of the database driver class path in the Campaign installation worksheet, as you must enter the path when you run the installer.
10. Restart the web application server so that your changes take effect.

If you are using Apache Tomcat, you must perform this entire procedure.

1. Obtain the latest vendor-provided Type 4 JDBC driver for your system table database that is supported by Unica, as described in the *Recommended Software Environments and Minimum System Requirements* guide.
2. Use the following guidelines after you obtain the JDBC driver.
    ◦ If the driver does not exist on the server where you plan to deploy Campaign, obtain it and unpack it on the server. Unpack the drivers in a path that does not include spaces.
    ◦ If you obtain the driver from a server where the data source client is installed, verify that the version is the latest supported by Campaign.
3. Copy the jar file manually to *$Tomcat_Installed_Location/lib/*, here $Tomcat_Installed_Location is the Tomcat root installation directory where you plan to deploy Unica Campaign.
4. Make a note of the database driver class path in the Campaign installation worksheet, as you must enter the path when you run the installer.
5. Restart the web application server so that your changes take effect. During startup, monitor the console log to confirm that the class path contains the path to the database driver.

Following parameters must be added in the connector in server.xml.

maxHttpHeaderSize="209715200"

maxPostSize="-1"

Details of the parameters are below:

1. maxHttpHeaderSize

   The maximum size of the request and response HTTP header, specified in bytes. If not specified, this attribute is set to 8192 (8 KB).

2. maxPostSize

   The maximum size in bytes of the POST which will be handled by the container FORM URL parameter parsing. The limit can be disabled by setting this attribute to a value less than zero. If not specified, this attribute is set to 2097152 (2 megabytes). Note that the FailedRequestFilter can be used to reject requests that exceed this limit.

Creating JDBC connections in web application server

Create JDBC connections on each of the web application servers where Unica Campaign is deployed. Unica Campaign uses the JDBC connections to access the required databases.

About this task

Use the following list to determine which JDBC connections you must create for the Unica Campaign web application. The steps include the suggested and required JNDI names.

> **Note:** JNDI for Unica Platform and Unica Campaign must be separate. You must not use the same JNDI (even if underlying database is same) for Unica Platform and Unica Campaign.

For more information about creating JDBC connections in your web application server, see the WebLogic or WebSphere® documentation.

1. Create a connection to the database holding the Unica Platform system tables. Use `UnicaPlatformDS` as the JNDI name.

   > **Important:** `UnicaPlatformDS` is the required JNDI name.

   If you are deploying Unica Campaign in the same JVM as your Unica Platform deployment, you have probably already set up this connection.

   > **Note:** If you are using WebLogic to configure a connection to an Oracle or DB2® database, you must also specify the database user name in the form `user=<DBUser>` in the Properties section on the Connection Pool tab.

2. Create a connection to the database holding the Unica Campaign system tables.

   You can allow the installer to create this data source (for one partition only) when you install Unica Campaign. To enable automatic data source creation during installation, on the Datasource Creation panel, select the **Create Campaign Datasource** check box and supply information about your application server.

   - The installer creates the data source using `campaignPartition1DS` as the JNDI name.
   - Note that, if you are using WebLogic, you must add the JDBC driver to your classpath manually even if you allow the installer to create the data source. The installer does this automatically for WebSphere®.
   - Automatic DataSource creation during installation is not supported for the JBOSS application server.

   If you do not want to allow the installer to create the data source, then you must create this data source.

   If there is one partition, the suggested JNDI name is `campaignPartition1DS`.

   If there are multiple partitions, a best practice is to use `campaignPartition1DS` for the first connection, `campaignPartition2DS` for the second, and so on.

   > **Note:** This practice is provided as an example. You can specify any JNDI names for Unica Campaign system table connections.

3. Record all JNDI names in the Unica Campaign installation worksheet on page 13

Information for creating JDBC connections

Use default values when you create JDBC connections if specific values are not provided. For more information, see the application server documentation.

> ✏️ **Note:** If you are not using the default port setting for your database, make sure that you change it to the correct value.

WebLogic

Use the following values if your application server is WebLogic:

SQLServer

- Database Driver: Microsoft™ MS SQL Server Driver (Type 4) Versions: 2012, 2012 SP1 and SP3, 2014, 2014 SP1, 2016 SP1

- Default port: 1433

- Driver class: `com.microsoft.sqlserver.jdbc.SQLServerDriver`

- Driver URL: `jdbc:sqlserver://<your_db_host>[\`
  `\<named_instance>]:<your_db_port>;databaseName=<your_db_name>`

- Properties: Add `user=<your_db_user_name>`

Oracle

- Driver: Other

- Default port: 1521

- Driver class: `oracle.jdbc.OracleDriver`

- Driver URL: `jdbc:oracle:thin:@<your_db_host>:<your_db_port>:<your_db_service_name>`

  Enter the driver URL by using the format that is shown. Unica applications do not allow the use of Oracle's RAC (Real Application Cluster) format for JDBC connections.

- Properties: Add `user=<your_db_user_name>`

DB2®

- Driver: Other

- Default port: 50000

- Driver class: `com.ibm.db2.jcc.DB2Driver`

- Driver URL: `jdbc:db2://<your_db_host>:<your_db_port>/<your_db_name>`

- Properties: Add `user=<your_db_user_name>`

MariaDB

- Database type: MYSQL
- Driver: mariadb-java-client-2.5.1.jar
- Default port: 3306
- Driver class: org.mariadb.jdbc.Driver
- Driver URL: jdbc:mariadb://<your_db_host>:<PORT>/<Your_DB_user_name>
- Properties: Add user=<your_db_user_name>
- Password=<your_db_user_password>

> ✏️ **Note:** From version 12.1.0.2, use:
>
> - MariaDB ODBC connector version 3.1.6 for Windows and Linux.
> - MariaDB ODBC connector version 3.1.9 for SUSE operating system.

Table 5. MariaDB ODBC connector versions

Microsoft Windows	**Maria DB ODBC Connector: 3.1.6** **Download link**: https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.6/mariadb-connector-odbc-3.1.6-win64.msi
RHEL	**Maria DB ODBC Connector: 3.1.6** **Download link**: https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.6/mariadb-connector-odbc-3.1.6-ga-rhel7-x86_64.tar.gz
SUSE	**Maria DB ODBC Connector: 3.1.9** **Download link** : https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.9/mariadb-connector-odbc-3.1.9-sles123-amd64.tar.gz

MariaDB ODBC driver level configurations

Example for Microsoft Windows: Set the following configuration in ODBC Driver Manager following the **Statements** parameter:

```
SET SQL_MODE='PIPES_AS_CONCAT,ANSI_QUOTES,IGNORE_SPACE,NO_KEY_OPTIONS';
```

Example for SUSE/Linux: Set the following configuration in `odbc.ini`:

```
[<CampaignSystem>]
Driver=/usr/local/mariaDBConnector/lib/libmaodbc.so
DATABASE=<DATABASE_NAME>
DESCRIPTION=MariaDB
SERVER=<DATABASE_SERVER>
UID=<USER_NAME>
PASSWORD=<PASSWORD>
PORT=3306
InitStmt=SET
SQL_MODE='PIPES_AS_CONCAT,ANSI_QUOTES,IGNORE_SPACE,NO_KEY_OPTIONS';
```

Attention: Use MariaDB database user name. Do not use `root` and `unica*03` as username and password for `UA_SYSTEM_TABLES` datasource.

MariaDB JAVA Client (JDBC Client) : Download the version 2.4.0 client using the following link https://mariadb.com/kb/en/library/about-mariadb-connector-j/

Note: For AIX, we do not support Maria DB as System and User Database.

WebSphere®

Use the following values if your application server is WebSphere®:

SQLServer

- Driver: N/A

- Default port: 1433

- Driver class: `com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource`

- Driver URL: jdbc:sqlserver://<DBhostName>:1433;databaseName=<DBName>

In the **Database Type** field, select **User-defined**.

After you create the JDBC Provider and data source, navigate to the **Custom Properties** for the data source, and add or modify properties as follows.

- `serverName=<your_SQL_server_name>`
- `portNumber =<SQL_Server_Port_Number>`
- `databaseName=<your_database_name>`

Add the following custom properties.

- Name: `webSphereDefaultIsolationLevel`

- Value: `1`

- Datatype: `Integer`

Oracle

- Driver: Oracle JDBC Driver

- Default port: 1521

- Driver class: `oracle.jdbc.OracleDriver`

- Driver URL: jdbc:oracle:thin:@*<your_db_host>:<your_db_port>:<your_db_service_name>*

  Enter the driver URL by using the format that is shown. Unica applications do not allow the use of Oracle's RAC (Real Application Cluster) format for JDBC connections.

DB2®

- Driver: JCC Driver Provider

- Default port: 50000

- Driver class: `com.ibm.db2.jcc.DB2Driver`

- Driver URL: jdbc:db2://*<your_db_host>:<your_db_port>/<your_db_name>*

To add the custom properties, complete the following steps.

1. Click the data source that you created. Go to the Custom Properties for the data source.
2. Select the **Custom properties** link.
3. Set the value for the `resultSetHoldability` property to `1`. If you do not see the `resultSetHoldability` property, create the `resultSetHoldability` property and set its value to `1`.
4. Set the value for the `webSphereDefaultIsolationLevel` property to `2`. If you do not see the `webSphereDefaultIsolationLevel` property, create the `webSphereDefaultIsolationLevel` property and set its value to `2`.

The following are the custom properties.

- Name: `webSphereDefaultIsolationLevel`

- Value: `2`

- Datatype: `Integer`

MariaDB

- Database type: User_define
- Implementation class name = org.mariadb.jdbc.MySQLDataSource
- Driver: mariadb-java-client-2.5.1.jar

  JNDI custom properties

- Component-managed authentication alias = Select the authentication details for user
- Mapping-configuration alias = WSLogin
- portNumber = 3306
- relaxAutoCommit=true
- databaseName = <DATABASE NAME>
- serverName = the host name of the MariaDB server
- user = the user name of the MariaDB server
- password = the password associated with the user name

Tomcat

Use the following values if your application server is Tomcat:

MariaDB

- Driver: mariadb-java-client-2.5.1.jar
- Default port: 3306
- Driver class: org.mariadb.jdbc.Driver
- Driver URL: jdbc:mariadb://<your_db_host>:<PORT>/<Your_DB_user_name>
- Properties: Add user=<your_db_user_name>
- Properties: Add password=<your_db_password>

JBoss

Specify the native library path of the database driver JAR file on your server.

Use the following values if your application server is JBoss:

SQL Server

- Database Driver: Microsoft MS SQL Server Driver (Type 4) Versions: 2012, 2012 SP1 and SP3, 2014, 2014 SP1, 2016 SP1
- Default port: 1433
- Driver class: com.microsoft.sqlserver.jdbc.SQLServerDriver
- Driver URL: jdbc:sqlserver://<your_db_host>:<your_db_port>;databaseName=<your_db_name>,validconnection-checker-class>
- name: org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker

  For example:

```
/subsystem=datasources/data-source=UnicaPlatformDS:add(jndiname=" java:/
UnicaPlatformDS",connection-url="jdbc:sqlserver://localhost:1433;databaseName=plat11",driver-
name=sql,username= sa,password=test1234,valid-connection-checker-
classname="org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker") **
```

Oracle

- Driver: Oracle JDBC Driver
- Default port: 1521
- Driver class: oracle.jdbc.OracleDriver
- Driver URL:jdbc:oracle:thin:@<your_db_host>:<your_db_port>:<your_db_service_name>

DB2

- Driver: JCC Driver Provider
- Default port: 50000
- Driver class: com.ibm.db2.jcc.DB2Driver
- Driver URL: jdbc:db2://<your_db_host>:<your_db_port>/<your_db_name>
- valid-connection-checker class-name= org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionChecker

MariaDB

- Database Driver: mariadb-java-client-2.5.1.jar
- Default port: `3306`
- Driver class: org.mariadb.jdbc.Driver
- Driver URL: jdbc:mariadb://<your_db_host>:<PORT>/<Your_DB_user_name>
- Properties: `Add user=<your_db_user_name>`
- Properties: `Add password=<your_db_password>`
- Driver module xa-datasource-class= org.mariadb.jdbc.MySQLDataSource

Preparing datasource for Unica Campaign using ODBC Connection

From V12.1 Fixpack 4 onwards, Unica Campaign supports Oracle for both System and User databases using ODBC connectivity. Existing implementation with native client connectivity will remain supported.

Oracle datasource connectivity with ODBC connection:

1. Configure sqlnet.ora file
    a. Go to <ORACLE_HOME>/network/admin
    b. Remove "SQLNET.ALLOWED_LOGON_VERSION_SERVER" parameter if it is present in sqlnet.ora file

       e.g.

#SQLNET.ALLOWED_LOGON_VERSION_SERVER=10

NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)

2. Restart Oracle Database

    a. Ask all users to logout from application

    b. Stop all EMM applications. Stop Server

    c. Go to <CAMPAIGN_HOME>/bin folder and stop the listener process started with **rc.unica_ac** or **CmpServer.bat**.

    d. Kill all unica_acsvr process if any exist after listener stops.

    e. Restart Oracle Database

    export ORACLE_SID=<SID>

    export ORACLE_HOME= <ORACLE_HOME>

    export PATH=$PATH:/<ORACLE_HOME>/bin

    cd <ORACLE_HOME>/bin

    Make sure you are logged in as root. Stop Oracle Listener service as a root user

    ./lsnrctl stop

    Change user to Oracle user and restart Oracle DB

    su oracle

    sqlplus "/ as sysdba"

    shutdown

    startup

    quit

    Now start Oracle Listener service as a root user

    ./lsnrctl start LISTENER

3. Reset DB user passwords.

    a. Reset DB user to same value as it was before.

    sqlplus "/ as sysdba"

    ALTER USER <username> IDENTIFIED BY <password>;

    ALTER USER <username> IDENTIFIED BY <password>;

    commit;

4. Check ODBC connectivity
    a. Go to <CAMPAIGN_HOME>/bin location
    b. Add the path of odbc.ini in setenv file

    ```
    export ODBCINI=<Path_to_odbc.ini>
    ```

    c. Start the Appserver.
    d. Now test the ODBC connectivity by running cxntest utility.

    e.g.

    . ./setenv.sh

    ./cxntest

    Connection Library? libodb4dDD.so

    Registered Data Sources:

    ORCL

    Data Source? ORCL

    User ID? <username>

    Password? <password>

    Executing Unica Campaign DB Certification Utility at 06 10 2020 17.20...

    Connecting to DB with "libraryName=libodb4dDD.so serverName=ORCL userID=UserName

    DB Connection successful!

    >

5. Import Oracle ODBC template
    a. Save the 'Oracle_ODBC_Template' from zip file at <CAMPAIGN_HOME>/conf location.
    b. Use configtool to Import this template (Make sure Appserver is up and running).
    c. Go to <PLATFORM_HOME>/tools/bin and execute following-

    ./configTool.sh -i -p "Affinium|Campaign|partitions|partition1|dataSources" -f "<CAMPAIGN_HOME>/conf / Oracle_ODBC_Template.xml"

    d. Make sure 'Oracle_ODBC_Template' is available under **Settings > Configurations > Campaign > partitions > partition1 > dataSources node**.
6. Creating System and User Datasources (Changes in Unica Application Configuration)

    a. Make sure to take the backup of System table DS i.e. UA_SYSTEM_TABLES and User table DS (if they already exist).

       Use the below command `configTool.bat -x -p "Affinium|Campaign|partitions|partition1|dataSources" -f "<Platform_Home>\tools\bin\DataSourceBackup.xml"`

    b. Delete System table DS (UA_SYSTEM_TABLES) and User table DS.

    c. Recreate both the datasources using Oracle_ODBC_Template.

    d. Keep all the properties identical to NATIVE datasource except the below mentioned.

    e. Make sure to set the properties as mentioned below:

DateFormat	DELIM_Y_M_D
DateOutputFormatString	%Y-%m-%d
DateTimeFormat	DT_DELIM_Y_M_D
DateTimeOutputFormatString	%Y-%m-%d %H:%M:%S

    f. Specify DSN exactly same as in 'odbc.ini' and 'tnsnames.ora' file.

> 📝 **Note:** DSN is case sensitive

    g. Specify following query in SQLOnConnect

       ALTER SESSION SET NLS_LANGUAGE='American' NLS_TERRITORY='America' NLS_TIMESTAMP_FORMAT='YYYY-MM-DD HH24:MI:SSXFF' NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS'

    h. Keep below mentioned properties as default

Type	Oracle ODBC
BindDateToTimestamp	TRUE

7. Changes in setenv file

   Add following NLS variables in setenv file

   export NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS'

   export NLS_TIMESTAMP_FORMAT='YYYY-MM-DD HH24:MI:SSXFF'

   For UNICODE environment below are the recommended settings:

   NLS_LANG=AMERICAN_AMERICA.AL32UTF8

   export NLS_LANG

   export LC_ALL=en_US.UTF-8

export LANG=en_US.UTF-8

8. Restart application server and Campaign Listener.

   a. Stop the Application Server (WebLogic or WebSphere) for Platform and Campaign.

   b. Clean the application server cache

   c. Start the Web application server to bring up all the applications.

   d. Start the listener process with "rc.unica_ac" or "CmpServer.bat".

9. Access application

   a. IMPORTANT! Be sure to clean the browser cache for every user.

   b. Access Campaign and make sure all other products are accessible.

Preparing MariaDB datasource for Unica Campaign

```
Before you install Unica products, ensure that you run this query on MariaDB. SET sql_mode =
 'NO_ZERO_DATE'
```

Complete the following steps to use a MariaDB database as a data source for Unica Campaign.

Unica Campaign supports MariaDB for both System and User databases. For information on supported versions, see the Recommended Software Environments and Minimum System Requirements document. For AIX, we do not support MariaDB as a System and User Database.

Database Client / Connector Installation
Windows:

1. Configure the following settings in my.ini for MariaDB database server is installed on Windows:

   ```
   [mysqld]
   sql-mode="NO_BACKSLASH_ESCAPES"
   innodb-page-size=32768
   character-set-server=utf8
   ```

2. Install the MariaDB Client and JDBC server, where the Unica Campaign analytical server (listener) is installed.

   ◦ MariaDB Connector/ODBC 3.1 Series Download link - https://downloads.mariadb.org/connector-odbc/
   ◦ https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.0/
   ◦ Sample ODBC Connector on Windows: mariadb-connector-odbc-3.1.0-win64.msi

3. Create DSN for Unica Campaign system database in Windows ODBC Manager (SystemDSN).

   ◦ Open ODBC Administrator GUI
   ◦ Click 'Add' button
   ◦ From the list select 'MariaDB ODBC Driver 3.1. Driver'
   ◦ Click 'Finish' button. This will open a new windows 'Create a new Data Source to MariaDB'
   ◦ Mention DSN name in 'name'
   ◦ text box and click 'Next'
   ◦ Mention appropriate 'Server Name', 'Port' (mostly 3306)
   ◦ Mention 'User name' and 'Password'. Note this has to be as same as DSN name for MariaDB
   ◦ Click 'Test DSN' button to check db connection

◦ On Success select the appropriate database from the list and click on 'Next' button

◦ In 'Statement(s):' section add the below line:

▪ **set SQL_MODE='PIPES_AS_CONCAT,ANSI_QUOTES,IGNORE_SPACE,NO_KEY_OPTIONS'**

The sql_mode system variable can be used to override the default behavior of the server in several contexts, such as:

▪ How SQL statements are parsed

▪ How SQL statements are executed

▪ How SQL statements handle errors

▪ How SQL statements interpret data types

Note: This is needed for MariaDB server 10.4 onwards to resolve some basic issues.

◦ Click on Next

◦ Click on 'Next'

Note: If you want to create DSN using SSL Settings then please fill up all required fields and click on 'Next'

◦ Click on 'Finish'

4. Import MariaODBCTemplate.xml in Unica Campaign data sources if not already imported by installer.

Linux/SUSE

1. Add the following lines to server.cnf (`/etc/my.cnf.d/server.cnf`) in section `mysqld` when MariaDB database is installed on Linux

```
[mysqld]
lower_case_table_names = 1
innodb-page-size=32768
character-set-server=utf8
sql_mode='NO_BACKSLASH_ESCAPES'
innodb_strict_mode=0
wait_timeout = 2592000
```

2. To install MariaDB Connector 3.0.2 for SUSE, complete the following substeps.

a. Go to https://downloads.mariadb.org/mariadb/repositories/#distro=SLES%26distro_release=sles12-amd64--sles12%26mirror=tuna%26version=10.4

b. Run the following commands.

▪ `sudo rpm --import https://yum.mariadb.org/RPM-GPG-KEY-MariaDB`

▪ `sudo zypper addrepo --gpgcheck --refresh https://yum.mariadb.org/10.4/sles/12/x86_64 mariadb`

▪ `sudo zypper addrepo --gpgcheck --refresh https://yum.mariadb.org/10.4/sles/12/x86_64 mariadb`

▪ `sudo zypper install MariaDB-connector`

3. To install MariaDB Connector/ODBC 3.1.0 for RHEL, complete the following substeps.

- ◦ Create directory using `odbc_package` and navigate to `cd odbc_package`.
- ◦ Run the following commands.

  - ▪
    ```
    wget
     https://downloads.mariadb.com/Connectors/odbc/connector-odbc-3.1.0/mariadb-connector-odbc
    -3.1.0-rc-rhel7-x86_64.tar.gz
    tar -xvzf mariadb-connector-odbc-3.1.0-ga-rhel7-x86_64.tar.gz
    sudo install lib64/libmaodbc.so /usr/lib64/
    MariaDB JAVA Client (JDBC Client): v2.4.0 -
    ```

    Note: Download link: https://mariadb.com/kb/en/library/about-mariadb-connector-j/

4. Create ODBC DSN for MariaDB:

   Sample example odbc.ini file entry

   `[MariaDB-server]`

   `Description=<ANY DESCRIPTION>`

   `Driver=<DRIVER_PATH>/libmaodbc.so>`

   `SERVER=<SERVER IP ADDRESS>`

   `USER=<DSN_NAME>`

5. `PASSWORD=<DSN_NAME>`

   `DATABASE=<DSN_NAME>`

   `PORT=<3306>`

   InitStmt=SET SQL_MODE='PIPES_AS_CONCAT,ANSI_QUOTES,IGNORE_SPACE,NO_KEY_OPTIONS';

   Note:
   - ◦ DSN_NAME - an appropriate name for data source which should be used in Campaign. This same value has to be set as a value for DATABASE, UID and PASSWORD params.
   - ◦ DRIVER_PATH - a path to maria db odbc driver installed on machine
   - ◦ SERVER_IP_ADDRESS - an address of a machine where MariaDB SERVER has installed
   - ◦ SQL_MODE - The sql_mode system variable can be used to override the default behavior of the server in several contexts, such as:

     How SQL statements are parsed.

     How SQL statements are executed.

     How SQL statements handle errors.

> How SQL statements interpret data types.
>
> This is needed for MariaDB server 10.4 onwards to resolve some basic issues.

6. Setup up setenv.sh with below entries.

   ```
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$lib_path:$Maria_DB_Driver_path

   export LD_LIBRARY_PATH

   ODBCINI=/etc/odbc.ini

   export ODBCINI

   ODBCINST=/etc/odbcinst.ini

   export ODBCINST
   ```

7. Test the connection using `isql` or the Campaign `CxnTest` utility.
8. Import `MariaODBCTemplate.xml` for MariaDB and configure it.

Preparing DB2 on z/OS user datasource for Unica Campaign

Follow these guidelines to use a DB2® database on z/OS® as a data source for Unica Campaign.

About this task

Unica Campaign supports DB2® 10.1 and 11 on z/OS® for customer tables only, not system tables. For information about supported versions, see the *Recommended Software Environments and Minimum System Requirements*.

1. The database user account must have SYSADM privileges. The DBADM privilege is not sufficient because it does not provide access to all of the required system objects. Without SYSADM privileges, flowcharts might not run successfully.
2. Database loaders are supported. However, there is not an out-of-the-box approach for loaders on z/OS®. You must set up USS Pipes on z/OS®, write a stored procedure to invoke the DSNUTILU and a script to invoke the stored procedure, then configure the loader to invoke the script. For details, see the Unica Campaign Administrator's Guide.
3. Set **DB2NotLoggedInitially** and **DB2NotLoggedInitiallyUserTables** to `FALSE`, under `Campaign|partitions| partition[n]|dataSources|dataSourcename`. For details, see the Unica Campaign Administrator's Guide.

Preparing Vertica datasources for Unica Campaign

Follow these guidelines to use a Vertica database as a data source for Unica Campaign.

About this task

For information about supported versions, see the *Recommended Software Environments and Minimum System Requirements.*

To download the latest Vertica documentation in zip or tar format: http://www.vertica.com/v-zone/product_documentation

1. Install the HP Vertica client driver on the server where the Unica Campaign analytical server (listener) is installed.

   Obtain the driver here: http://www.vertica.com/resources/vertica-client-drivers/

2. Configure the `odbc.ini` file.

   Sample odbc.ini file entry

```
[ODBC Data Sources]

DriverManagerEncoding=UTF-16
ODBCInstLib=/usr/local/lib/libodbcinst.so
ErrorMessagesPath=/opt/vertica/lib64
LogLevel=4
LogPath=/tmp

[Vertica]
Description = VerticaDSN ODBC driver
Driver = /opt/vertica/lib64/libverticaodbc.so
Database = exampleDB
Servername = example.com
UserName = dbadmin
Password = example_password
Port = 5433
```

Preparing Amazon Redshift user data sources for Unica Campaign

Follow these guidelines to use Amazon Redshift as a data source for Unica Campaign.

About this task

Unica Campaign supports Amazon Redshift for customer tables. Redshift is not supported for system tables.

Use the Amazon native ODBC driver for connectivity and configure the PostgreSQL template in Unica Campaign for each datasource. See the Unica Campaign Administrator's Guide for important information about configuring `Campaign| partitions|partition[n]|dataSources|PostgreSQL`. Some properties do not have default values, so you must supply them. Other properties might need to be adjusted for your specific configuration. For information about supported versions, see the *Recommended Software Environments and Minimum System Requirements*. For information about Amazon Redshift, see http://aws.amazon.com/redshift/

1. Download and configure the Amazon Redshift ODBC driver for Redshift. Driver Name : Amazon Redshift ODBC driver (version - 1.4.11.1000 –64-bit), Unix ODBC driver manager (Version - 2.3.7). See https://docs.aws.amazon.com/redshift/latest/mgmt/install-odbc-driver-linux.html

   ✏️ **Note:** Use unicaODBC driver version 2.3.7 or above.

2. Validate connection and authentication details: Use `isql` to test the connection on Linux™. Use the Microsoft™ Windows™ ODBC Driver Manager for Windows™.
3. Import `PostgreSQLTemplate.xml` using `configTool` (a utility supplied with Unica Platform).
4. Fill in the newly added `PostgreSQL` template with the user database details.
5. Configure the Amazon Redshift user database.
6. Restart the application server and the Unica Campaign listener server.

What to do next

Follow the steps below to test the connection with Amazon Redshift.

✏️ **Note:** See the following link to get the cluster connection string: http://docs.aws.amazon.com/redshift/latest/mgmt/configuring-connections.html#connecting-drivers

On Windows™:

1. Open Windows™ ODBC Driver Manager.
2. Add a System DSN and select the Amazon Native driver.
3. Provide the connection string and other details.
4. Test the connection using `ODBC driver test connection button` or the Unica Campaign `CxnTest` utility.

On Linux™:

1. Configure LD_LIBRARY_PATH (or LIBPATH on AIX®) to include the location of the driver library files.
2. Configure ODBCINI to point to the odbc.ini file location.
3. Configure the odbc.ini file for the Redshift user database. For example:

```
[ODBC]

Trace = yes

TraceFile = /tmp/OdbcTrace.log

#[ODBC Data Sources]

AmazonRedshift=Amazon Redshift

[AmazonRedshift]

This key is not necessary and is only to give a description of the data source.

driver_name=Amazon_Redshift

Description=Amazon Redshift ODBC Driver (64-bit) DSN

#[Driver: The location where the ODBC driver is installed to.]
```

```
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so

Required: These values can also be specified in the connection string.

Host=<Redshift_Database_Name>

Port=<Port>

Database=<database_Name>

locale=en-US

BoolsAsChar=0
```

4. Test the connection using `isql` or the Unica Campaign `CxnTest and odbctest` utility.
5. Import `PostgreSQLTemplate.xml`for Amazon Redshift and configure it.
6. For Amazon Redshift loader utility. Please refer the Campaign Administrator guide for more details.

Note: Extract PB fails to run if you select Target Data source as Amazon Redshift . So as recommended by AWS support team we need to use "BoolsAsChar=0" in odbc.ini

Preparing Apache Hadoop Hive user data sources for Unica Campaign

Follow the steps below to enable Hive-based Apache Hadoop data sources to be used in Unica Campaign

About this task

Unica Campaign supports Apache Hadoop Hive for customer tables only, not system tables. For details about supported versions, see the *Recommended Software Environments and Minimum System Requirements*.

Task	Description
A. Install a Hive ODBC driver on page 43	You can install the DataDirect Hive ODBC Driver from Progress Software, the Cloudera Hive ODBC driver from Cloudera, Inc., or the Hortonworks Hive ODBC driver from Hortonworks, Inc.
B. Configure the Hive ODBC Driver on page 45	Configuration includes modifying .ini files and setting path values and environment variables. Be sure to follow the appropriate instructions for the driver that you installed.
C. Map existing HBase tables to Hive on page 50 (OPTIONAL)	This step is required only if you have existing tables that were created in Apache HBase.
D. Import and configure the BigDataODBCHiveTemplate data source template in Unica Campaign on page 51	Use the `configTool` utility to import the template `BigDataODBCHive.xml` into Unica Campaign.

Task	Description
	Then go to `Campaign|partitions|partition[n]|dataSources` and create a datasource based on the imported `BigDataODBCHiveTemplate`.
E. Configure SSH on the Unica Campaign listener server on page 53	To enable data file transfers between the Unica Campaign listener (analytic) server and the Hive-based Hadoop big data instance, you must configure SCP and SSH seamless login.
F. Map the Hive datasource in Unica Campaign on page 54	Mapping user tables is the process of making external data sources accessible in Unica Campaign.

Results

You can now use your Hive-based Hadoop big data system as a user data source for Unica Campaign. For example, create a marketing campaign flowchart that uses customer account data from your big data instance to target customers with specific account types and balances.

Requirements and restrictions for Hive-based Hadoop data sources

The following requirements and restrictions apply to using Hive-based Hadoop data sources with Unica Campaign.

- One of the following drivers is required. The customer is responsible for obtaining the driver.
    - DataDirect Apache Hive ODBC driver from Progress.com: DataDirect Connect64(R) for ODBC Release 7.1.5.
    - Cloudera ODBC Driver for Apache Hive Version 2.5.16 for Apache Hadoop Hive from Cloudera, Inc.
    - Hortonworks 64-bit ODBC driver for HDP 2.3 (v2.0.5) available at http://hortonworks.com/hdp/addons/.
- The following Hadoop distributions are supported, with Apache Hive as the connection point: Cloudera, Hortonworks, IBM® BigInsights® ™, MapR.
- Minimum supported Hive version: 0.14
- The big data integration is currently supported on Linux™ RHEL 6.3 or higher.
- Hive-based Hadoop is supported as a user data source only. It is not supported for Unica Campaign system tables.
- The big data integration does not currently support the Unica Campaign Cube, Optimize, or Interact List process boxes or Deliver Landing Pages in an Extract process box.

> **Note:** It is mandatory for Hive database users to configure the Loader.

Overview of the Unica Campaign integration with Hadoop

Integrating Unica Campaign with Hadoop data sources demands extra software that is not part of the standard Unica Campaign installation requirement. You must be familiar with the terminology and structure of the integration and be able to access various resources that are available to support the integration.

Integration architecture: Hive-based Hadoop and Unica Campaign

This diagram shows the integration architecture for Unica Campaign and Hive-based Hadoop user data sources.
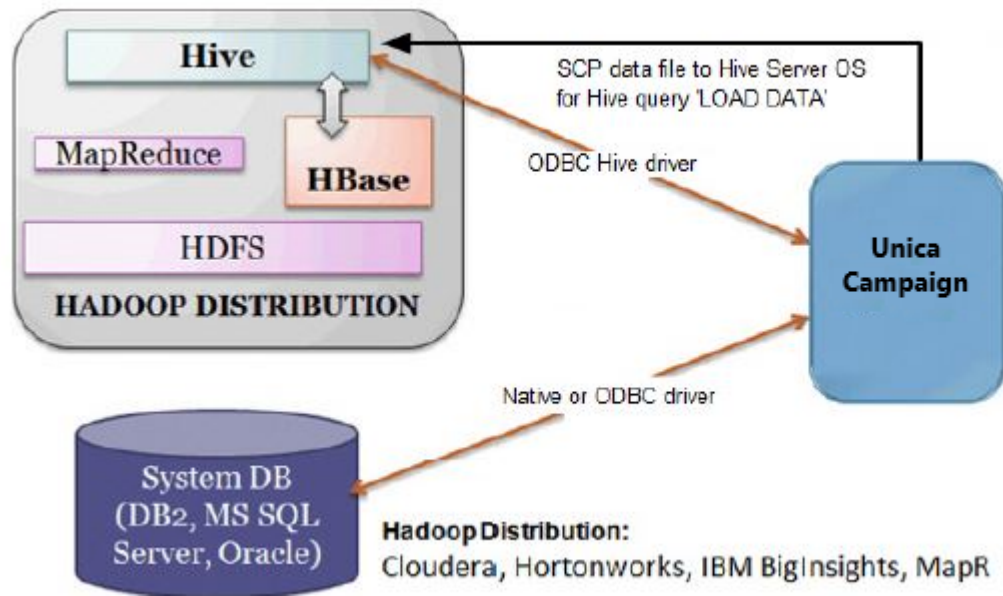
The Apache Hive data warehouse software facilitates querying and managing large datasets that reside in distributed storage. Built on top of Apache Hadoop, Hive provides:

- Tools to enable easy data extract/transform/load (ETL)
- A mechanism to impose structure on a variety of data formats
- Access to files stored either directly in Apache HDFS or in other data storage systems such as Apache HBase
- Query execution via MapReduce

Hive defines a simple SQL-like query language, called HiveQL (or HQL), that enables users familiar with SQL to query the data.

You can use the Hue editor (Hadoop UI) to work with your big data instance (for example: connect, view, create tables and databases).

Terminology and resources for Hive Hadoop

To understand the Hive Hadoop integration with Unica Campaign, refer to this list of definitions and links to online resources.

Terms

Apache Hadoop® is an open-source software framework written in Java™ for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware.

Apache Hive™ is a data warehouse infrastructure built on top of Hadoop to facilitate querying and managing large datasets that reside in distributed storage. Hive provides a mechanism to project structure onto this data and query the data using an SQL-like language called HiveQL.

Apache HBase™ is an open source, non-relational, distributed database written in Java™. It runs on top of HDFS, providing BigTable-like capabilities for Hadoop.

Hadoop Distributed File System (HDFS™**)** is a distributed file system that runs on commodity hardware. It is designed to reliably store very large files across machines in a large cluster.

Hue is a Web interface for analyzing data with Apache Hadoop.

HiveQL (or HQL) is the Hive query language.

MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. MapReduce is the heart of Hadoop®. It is this programming paradigm that allows for massive scalability across hundreds or thousands of servers in a Hadoop cluster.

Big data **distributions** of Apache Hadoop: Several vendors have developed their own distributions of Hadoop, including **Cloudera, Hortonworks, IBM® BigInsights®**, and **MapR**.

User tables is an Unica Campaign term that indicates any data source that contains an organization's marketing data for access by Unica Campaign flowcharts. Typically, user tables contain data about customers, prospects, and products. For example, customer account data pulled from user tables could be used in a flowchart to target customers with specific account types and balances.

Online resources

The following links were accurate at the time of publication but are subject to change:

- Cloudera ODBC driver: https://www.progress.com/products/data-sources/apache-hadoop-hive
- Hive ODBC driver: https://www.progress.com/products/data-sources/apache-hadoop-hive
- Hive: https://cwiki.apache.org/confluence/display/Hive/Home
- HiveQL: https://cwiki.apache.org/confluence/display/Hive/LanguageManual
- Hive HBase integration: https://cwiki.apache.org/confluence/display/Hive/HBaseIntegration
- Hue and Hadoop: http://gethue.com

A. Install a Hive ODBC driver

This is the first step to integrate Unica Campaign with Hive-based Apache Hadoop data sources. You must install either the DataDirect Hive ODBC Driver from Progress Software or the Cloudera Hive ODBC driver from Cloudera, Inc.

You can install any one of the Hive ODBC drivers that are described below. Be sure to follow the appropriate instructions for the driver that you select.

Install the Hive ODBC driver on the system where you have installed the Unica Campaign Analytical Server. If the Analytical Server is installed in a cluster, install the Hive ODBC driver on each system.

Installing the DataDirect Hive ODBC driver

This topic explains how to install the DataDirect driver for Apache Hive, which is a fully compliant ODBC driver that supports multiple Hadoop distributions.

Before you begin

KornShell (ksh) must be installed on the Unica Campaign listener (analytic) server.

1. Obtain the Progress DataDirect Connect ODBC driver (Progress DataDirect Connect64(R) for ODBC Release 7.1.5) for Apache Hadoop Hive: https://www.progress.com/products/data-sources/apache-hadoop-hive
2. Download and install the DataDirect Hive driver on the Unica Campaign listener (analytic server):
   Example

   PROGRESS_DATADIRECT_CONNECT64_ODBC_7.1.5_LINUX_64.tar.Z

   [DataDirectNew]# **gunzip** PROGRESS_DATADIRECT_CONNECT64_ODBC_7.1.5_LINUX_64.tar.Z

   [DataDirectNew]# **tar -xvf** PROGRESS_DATADIRECT_CONNECT64_ODBC_7.1.5_LINUX_64.tar
3. Run the following command to start the installation:
   Example

   >> ksh ./unixmi.ksh
4. Follow the prompts to complete the installation.
5. Perform basic testing of the driver:
   Example

   >> ./ddtestlib /opt/Progress/DataDirect/Connect64_for_ODBC_71/lib/ddhive27.so

What to do next
See .

Installing the Cloudera ODBC driver

This topic explains how to install the Cloudera ODBC driver, which is a fully compliant ODBC driver that supports multiple Hadoop distributions.

1. Obtain the Cloudera ODBC Driver for Apache Hive Version 2.5.16 for Apache Hadoop Hive at http://www.cloudera.com/content/cloudera/en/downloads/connectors/hive/odbc/hive-odbc-v2-5-16.html. Download and save the Cloudera Hive ODBC driver on the Unica Campaign listener (analytic server).
   The default location is `/opt/cloudera/`.
2. Install and configure the Cloudera driver. Installation instructions are downloaded to where you install the driver.
   By default, the instructions are downloaded to `/opt/cloudera/hiveodbc/Cloudera-ODBC-Driver-for-Apache-Hive-Install-Guide.pdf`. The PDF file is also available online at www.cloudera.com.

3. Download and install UnixODBC Manager, `unixODBC-2.3.2` on the Unica Campaign listener (analytic server). Download UnixODBC Manager from the unixODBC Project at http://www.unixodbc.org/. The UnixODBC Manager is available in a gzip, tar format.

   The steps to install UnixODBC are available at http://www.unixodbc.org/download.html

   To create a unixODBC directory that contains the required source files:

   a. Copy the unixODBC-2.3.2.tar.gz file to the Unica Campaign listener (analytic server).
   b. gunzip unixODBC-2.3.2.tar.gz
   c. tar xvf unixODBC-2.3.2.tar

4. On the Unica Campaign listener, run `rpm -ivh ClouderaHiveODBC-2.5.16.1005-1.el6.x86_64.rpm`.

What to do next

See .

Installing the Hortonworks Hive ODBC driver

This topic explains how to install the Hortonworks driver for Apache Hive, which is a fully compliant ODBC driver that supports multiple Hadoop distributions.

Before you begin

KornShell (ksh) must be installed on the Unica Campaign listener (analytic) server.

1. Obtain the 64-bit version of the Hortonworks Hive ODBC driver: http://hortonworks.com/hdp/addons
2. Install the Hortonworks ODBC driver on the Unica Campaign listener (analytic server):
   Example

   rpm -ivh hive-odbc-native-2.0.5.1005-1.el6.x86_64.rpm

   The default installation location of the Hortonworks Hive ODBC driver is `/usr/lib/hive/lib/native`.

   For more information on installing the Hortonworks Hive ODBC driver see: http://hortonworks.com/wp-content/uploads/2015/10/Hortonworks-Hive-ODBC-Driver-User-Guide.pdf.

3. Follow the prompts to complete the installation.

What to do next

See .

B. Configure the Hive ODBC Driver

This is the second step to integrate Unica Campaign with Hive-based Apache Hadoop data sources. You must configure the driver that you installed (either the DataDirect Hive ODBC Driver from Progress Software or the Cloudera Hive ODBC driver from Cloudera, Inc.).

Configuring the DataDirect Hive ODBC driver

If you are using the DataDirect Hive ODBC driver to integrate Unica Campaign with a big data source, you must configure the driver after installing it.

Before you begin

Complete the DataDirect driver installation that is described in .

1. Modify the ODBC.ini file to specify the Hive server information, using the values shown in the following example. Be sure to customize the items in **bold** to match your own configuration.
   Example

   [MapRHive]
   Driver=/opt/Progress/DataDirect/Connect64_for_ODBC_71/lib/ddhive27.so
   Description=DataDirect 7.1 Apache Hive Wire Protocol
   ArraySize=16384
   Database=**\<database-name\>**
   DefaultLongDataBuffLen=1024
   EnableDescribeParam=0
   HostName=**\<hostname or ip of Hive server on Hadoop Distribution machine\>**
   LoginTimeout=30
   LoginID=**\<username of Hadoop Distribution machine\>**
   MaxVarcharSize=2147483647
   Password=**\<password of Hadoop Distribution machine\>**
   PortNumber=**\<port number of Hive server on Hadoop Distribution machine\>**
   RemoveColumnQualifiers=1
   StringDescribeType=12
   TransactionMode=0
   UseCurrentSchema=0
   WireProtocolVersion=0
   GetTablesWithQuery=1

2. Assuming that your ODBC driver is installed in the following location:
   Example

   /opt/Progress/DataDirect/Connect64_for_ODBC_71

   Make the following changes:

   ◦ Ensure that your LD_LIBRARY_PATH includes the following path: =/opt/Progress/DataDirect/ Connect64_for_ODBC_71/lib
   ◦ Ensure that your PATH includes the following path: =/opt/Progress/DataDirect/Connect64_for_ODBC_71/ tools

- Set your ODBCINI variable to point to the correct INI file. For example: ODBCINI=/opt/Progress/DataDirect/
Connect64_for_ODBC_71/odbc.ini ; export ODBCINI
- Set your ODBCINST variable to point to the correct INI file. For example: ODBCINST=/opt/Progress/
DataDirect/Connect64_for_ODBC_71/odbcinst.ini ; export ODBCINST

3. Verify the connectivity of the DataDirect ODBC driver and your Hive-based Hadoop big data system:

Example

cd /opt/Progress/DataDirect/Connect64_for_ODBC_71/samples/example

>> ./example

4. On the Unica Campaign Analytical Server, set the ODBCINI and CAMPAIGN_HOME environment variables in the
`Campaign/bin/setenv` file. Run the Unica Campaign `odbctest` utility to verify connectivity to Unica Campaign:

Example

cd <Campaign_Home>/bin

>> ./odbctest

What to do next

Configuring the Cloudera ODBC driver

If you are using the Cloudera ODBC driver to integrate Unica Campaign with a big data source, you must configure the driver
after installing it.

Before you begin

Complete the Cloudera driver installation that is described in .

1. Modify the ODBC.ini file to specify the Hive server information, using the values shown in the following example. Be
sure to customize the items in **bold** to match your own configuration.

> 📝 **Note:** Do not enable Native Query mode in the configuration.

Example

```
[Cloudera_HIVE]

Driver=/opt/cloudera/hiveodbc/lib/64/libclouderahiveodbc64.so

Description=Hive Cloudera ODBC Driver

Host=<hostname or ip of Hive server on Hadoop Distribution machine>

Port=<port number of Hive server on Hadoop Distribution machine>

Schema=<database-name>

ServiceDiscoveryMode=0

ZKNamespace=

HiveServerType=2
```

```
AuthMech=3

ThriftTransport=1

UseNativeQuery=0

UID=cloudera

GetTablesWithQuery=1
```

> **Note:** The UID is the user ID of the Cloudera cluster database.

2. Assuming that your ODBC driver is installed in the following location:

   Example

   `/opt/cloudera/hiveodbc/lib/64/libclouderahiveodbc64.so`

   Make the following change in `Campaign/bin setenv.sh`: Set `LD_LIBRARY_PATH` (`LIBPATH` for AIX®) to include `Cloudera LIB` and `UnixODBC LIB` as follows.

   ◦ `LD_LIBRARY_PATH=/opt/HCL/UCD912_Install/Campaign/bin:/opt/ibm/db2/V10.1/lib64/:/usr/`**`lib:/lib:/opt/`**
     `cloudera/hiveodbc/lib/64/:/usr/local/lib/`

     `export LD_LIBRARY_PATH`

   ◦ `ODBCINI=/usr/local/etc/odbc.ini export ODBCINI`

3. Open `/opt/cloudera/hiveodbc/lib/64/cloudera.hiveodbc.ini`

   During Cloudera installation, a connectivity option is set by default. To connect Cloudera with ODBC, you must disable the connectivity option.

   Change the following lines in `cloudera.hiveodbc.ini`:

   ```
   # Generic ODBCInstLib
   # iODBC
   ODBCInstLib=libiodbcinst.so
   ```

   The updated lines appear as:

   ```
   # Generic ODBCInstLib
   # iODBC
   ODBCInstLib=libodbcinst.so
   ```

What to do next

Configuring the Hortonworks Hive ODBC driver

If you are using the Hortonworks Hive ODBC driver to integrate Unica Campaign with a big data source, you must configure the driver after installing it.

Before you begin

Complete the Hortonworks driver installation that is described in Installing the Hortonworks Hive ODBC driver on page 45.

1. Modify the `ODBC.ini` file to specify the Hive server information, using the values shown in the following example. Be sure to customize the items in **bold** to match your own configuration.

   The default location of odbc.ini file is: `/usr/lib/hive/lib/native/hiveodbc/Setup/odbc.ini`

   Example

   [ODBC] [ODBC Data Sources] HWDS [HWDS] Description=Hortonworks Hive ODBC Driver (64-bit)
   DSN Driver=/usr/lib/hive/lib/native/Linux™-amd64-64/libhortonworkshiveodbc64.so
   HOST=**<hostname or ip of Hive server on Hadoop Distribution machine>**
   PORT=**<port number of Hive server on Hadoop Distribution machine>**
   Schema=**<schema-name>**
   ServiceDiscoveryMode=0
   ZKNamespace=
   HiveServerType=2
   AuthMech=2
   ThriftTransport=1
   UseNativeQuery=0
   UID=hue
   KrbServiceName=**<Hive Server 2 Kerberos service name>**
   KrbRealm=**<Hive Server 2 Kerberos realm>**
   SSL=0
   TwoWaySSL=0
   ClientCert=
   ClientPrivateKey=
   ClientPrivateKeyPassword=

2. Enable the unixODBC manager and disable `Generic ODBCInstLib` in the `hortonworks.hiveodbc.ini` as follows.

   The default location of `hortonworks.hiveodbc.ini` is `/usr/lib/hive/lib/native/Linux-amd64-64/ hortonworks.hiveodbc.ini`.

   Example

   Comment out the following line by adding a hash character (#) at the beginning: `# ODBCInstLib=libiodbcinst.so`

   Un-comment the following line by removing the hash character (#) at the beginning: `ODBCInstLib=/usr/local/ unixODBC/lib/libodbcinst.so`

3. Edit the `Campaign/bin/setenv.sh` as follows.

   a. Ensure that your LD_LIBRARY_PATH includes the following path: `/usr/local/unixODBC/lib`

   b. Set your ODBCINI variable to point to the correct INI file.

For example:

```
ODBCINI=/usr/lib/hive/lib/native/hiveodbc/Setup/odbc.ini
```

```
export ODBCINII
```

```
HORTONWORKSHIVEINI=/usr/lib/hive/lib/native/Linux-amd64-64/hortonworks.hiveodbc.ini
```

```
export HORTONWORKSHIVEINI
```

What to do next

C. Map existing HBase tables to Hive

This is the third step to integrate Unica Campaign with Hive-based Apache Hadoop data sources. If you do not have existing tables that were created in Apache HBase, you can skip this step.

Before you begin

Complete B. Configure the Hive ODBC Driver on page 45.

About this task

This step is required only if you have existing tables that were created in Apache HBase. In this case, you must make the existing HBase tables available to Apache Hive by running the CREATE EXTERNAL TABLE query. After you expose the HBase tables to Hive, the tables are then available to Unica Campaign for table mapping within Unica Campaign.

The EXTERNAL keyword lets you create a table and provide a LOCATION so that Hive does not use a default location for the table. This is useful if you already have data generated. An EXTERNAL table points to any HDFS location for its storage, rather than being stored in a folder specified by the configuration property. When dropping an EXTERNAL table, data in the table is NOT deleted from the file system.

For information about Hive HBase integration: https://cwiki.apache.org/confluence/display/Hive/HBaseIntegration

1. Open the Hue editor and open the Hive Query Editor.
2. Create and execute the CREATE EXTERNAL TABLE command.

   Example

   Use the following query as an example, substituting your own table name, field names, and other parameters. This example uses 'CampaignAccounts' as the table name and 'f' as the family name.

   Example

   CREATE EXTERNAL TABLE **HiveExt_CampaignAccounts**(Acct_ID INT,Indiv_ID INT,HHold_ID INT,Acct_Type_Code STRING,Acct_Status_Code INT,Acct_Open_Date INT,Acct_Balance STRING,Acct_Balance_Last_Month STRING,Acct_Balance_Avg_6Month STRING,Credit_Limit STRING,Acct_Number STRING,Last_Contact_Date STRING,Due_Date STRING) STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH

SERDEPROPERTIES ('hbase.columns.mapping' = ':key,f:Indiv_ID, f:HHold_ID, f:Acct_Type_Code, f:Acct_Status_Code, f:Acct_Open_Date, f:Acct_Balance, f:Acct_Balance_Last_Month, f:Acct_Balance_Avg_6Month, f:Credit_Limit, f:Acct_Number, f:Last_Contact_Date, f:Due_Date') TBLPROPERTIES ('hbase.table.name' = '**CampaignAccounts**');

What to do next

D. Import and configure the BigDataODBCHiveTemplate data source template in Unica Campaign

This is the fourth step to integrate Unica Campaign with Hive-based Apache Hadoop data sources.

Before you begin

Complete .

About this task

Hive datasource does not provide other configuration parameters like Oracle/DB2 database datasource templates. The configuration properties, which are not supported are not provided within hive template. Configuration parameters for hive are supported selectively based on the implementation support.

To enable Unica Campaign to communicate with your Hive-based Hadoop system, you must do the following actions.

- Import the BigDataODBCHive.xml template into Unica Campaign. You must import the template only once. Importing a template makes it available for creating data sources.
- Use the template to create and configure a data source for each Hive implementation that communicates with Unica Campaign.
- For each data source, configure the `HiveQueryMode` property in the Unica Campaign configuration.

1. Use the `configTool` utility to import the `BigDataODBCHive.xml` template into Unica Campaign.
    ◦ `BigDataODBCHive.xml` is in `<Campaign_Home>/conf`.
    ◦ `configTool` is in `<Platform_Home>/tools/bin`. For more information, see the Unica Platform Administrator Guide.

    The following example imports the template into the default Unica Campaign partition, `partition1`. Replace `<Campaign_Home>` with the complete path to the Unica Campaign installation directory.

    ```
    ./configTool -i -p "Affinium|Campaign|partitions|partition1|dataSources" -f
    <Campaign_Home>/conf/BigDataODBCHive.xml
    ```

2. Create a data source based on `BigDataODBCHiveTemplate`. Do this for each Hive implementation that communicates with Unica Campaign. For example, if you have four implementations (MapR, Cloudera, Hortonworks, BigInsights®), create four separate data sources, and configure each one.
    Example
    a. In Unica Campaign, choose **Settings > Configuration**
    b. Go to `Campaign|partitions|partition[n]|dataSources`.
    c. Select `BigDataODBCHiveTemplate`.

d. Supply a New category name that identifies the Hive dataSource, for example **Hive_MapR** or **Hive_Cloudera** or **Hive_HortonWorks** or **Hive_BigInsights**.

e. Complete the fields to set the properties for the new data source, then save your changes.

> 🔔 **Important:** Some properties do not have default values, so you must supply them. Pay special attention to the properties described below. This is only a partial list of the properties included in this template. For complete information, see the Unica Campaign Administrator's Guide.

Configuration property	Description
ASMUserForDBCredentials	No default value defined. Specify the Unica Campaign system user.
DSN	DSN Name as specified in the odbc.ini file for the Hive-based Hadoop big data instance.
HiveQueryMode	For data sources that use the DataDirect ODBC driver, use `Native`. For data sources that use the Cloudera ODBC driver or Hortonworks Hive ODBC driver, use `SQL`.
JndiName	Not needed for user data source.
SystemTableSchema	No default value defined. Specify the user of the database that you connect to.
OwnerForTableDisplay	No default value defined. Specify the user of the database that you connect to.
LoaderPreLoadDataFileCopyCmd	SCP is used to copy data from Unica Campaign to a temp folder called `/tmp` on the Hive-based Hadoop system. The location must be called `/tmp` and it must be on the Hive server (the file system location, not the HDFS location). This value can either specify the SCP command or call a script that specifies the command. For more information and detailed instructions about how to export data from Unica Campaign to a Hive-based Hadoop system, see the Unica Campaign Administrator's Guide.
LoaderPostLoadDataFileRemoveCmd	Data files are copied from Unica Campaign to a temp folder on the Hive-based Hadoop system. You must use the SSH "rm" command to remove the temporary data file. For more information and detailed instructions about how to export data from Unica Campaign to a Hive-based Hadoop system, see the Unica Campaign Administrator's Guide.
LoaderDelimiter	No default value defined. Specify the delimiter such as comma (,) or semi-colon (;) that separates fields in the temporary data files that are loaded into the big data instance. Tab (/t) is not supported. The delimiter value must match the ROW format delimiter that was used when the big data database table was created. In this example, a comma is used: ROW FORMAT DELIMITED FIELDS TERMINATED BY '; ;"
SuffixOnTempTableCreation SuffixOnSegmentTableCreation SuffixOnSnapshotTableCreation SuffixOnExtractTableCreation	No default value defined. Use the same character as specified for LoaderDelimiter.

Configuration property	Description
SuffixOnUserBaseTableCreation SuffixOnUserTableCreation	
UseExceptForMerge	Set to FALSE. Hive does not support the EXCEPT clause, so a setting of TRUE can result in process failures.
DateFormat DateTimeFormat DateTimeOutputFormatString	All Date strings must use the dash "-" character to format dates. Hive does not support any other characters for dates. Example: %Y-%m-%d %H:%M:%S
Type	BigDataODBC_Hive
UseSQLToRetrieveSchema	Set to FALSE.

DataFileStagingFolder	Default location value is set to `/tmp`. You can change the location value. Example: `/opt/campaign/`
	
	Note: The value for this folder must have a trailing slash.
	
	If you have written shell script to copy the Campaign data file to the Hive server, you need to modify it. Example:
	
	```#!/bin/sh\nscp $1 root@emm52.in.hcl.com:/opt/campaign/\nssh root@emm52.in.hcl.com "chmod 0666 /opt/campaign/\n`basename $1`"```
	
	If you are using `LoaderPreLoadDataFileCopyCmd`, then you need to update the file location. Example:
	
	```scp <DATAFILE> <USER>@[hostname]:/opt/campaign/```
	
	If you are using `LoaderPostLoadDataFileRemoveCmd`, then you need to update the file location. Example:
	
	```ssh <USER>@[hostname] "rm /opt/campaign/<DATAFILE>"```

What to do next

E. Configure SSH on the Unica Campaign listener server

This is the fifth step to integrate Unica Campaign with Hive-based Apache Hadoop data sources.

Before you begin

Complete .

About this task

To enable data file transfers between the Unica Campaign listener (analytic) server and the Hive-based Hadoop big data instance, you must configure SCP and SSH seamless login. SSH allows secure connection between two computers; most systems use the OpenSSH client.

1. On the machine that is running the Unica Campaign listener, configure SSH for no password prompt for authentication. Log in as the user who is running the Unica Campaign listener and run the following commands, substituting the **username@IP address** of your Hive server (in this example, the MapR machine):
   Example

   >> ssh-keygen -t rsa

   >> ssh **mapr@192.0.2.0** mkdir -p .ssh

   >> cat .ssh/id_rsa.pub | ssh **mapr@192.0.2.0** 'cat >> .ssh/authorized_keys'

   >> ssh **mapr@192.0.2.0** "chmod 700 .ssh; chmod 640 .ssh/authorized_keys"

2. Verify password-less authentication using RSA-based authorized keys. Run each command, substituting the actual **username@IP address**, and verify that it works. You need a local file called test1 for this test to work:
   Example

   >> ssh **mapr@192.0.2.0**

   >> scp test1 **mapr@192.0.2.0**:/tmp

   >> ssh **mapr@192.0.2.0** "rm /tmp/test1"

What to do next

F. Map the Hive datasource in Unica Campaign

This is the sixth and final step to integrate Unica Campaign with Hive-based Apache Hadoop data sources.

Before you begin

- Complete
- You must define audience levels in Unica Campaign before you map user tables.

About this task

Mapping user tables makes external data sources accessible in Unica Campaign. A typical user table contains information about your company's customers, prospects, or products, for use in marketing campaigns. You must map any data sources that you configured to make that data accessible to processes in flowcharts.

1. Select **Settings > Campaign Settings > Manage Table Mappings**.
2. In the Table Mappings dialog, click **Show User Tables**.
3. Click **New Table**. The New Table Definition dialog opens.
4. Click **Next**.
5. Select **Map to Existing Table in Selected Database**.
6. Select the **BigDataODBCHive** datasource, then click **Next**.
7. Follow the prompts to map the table, as explained in the *Unica Campaign Administrator's Guide*.

What to do next

You have completed the installation and configuration process. Unica Campaign flowchart designers can now select Hive-based Hadoop data sources in processes. For example, a Select process can query a Hive-based data source to target customer data for a marketing campaign.

Préparation des sources de données Apache Hadoop Impala pour Unica Campaign

Suivez les étapes ci-dessous pour activer les sources de données Cloudera Hadoop basées sur Impala à utiliser dans Unica Campaign

About this task

Unica Campaign prend en charge Cloudera Hadoop en utilisant Impala pour les tables client uniquement, pas pour les tables système. Pour plus d'informations sur les versions prises en charge, consultez les *environnements logiciels recommandés et la configuration système minimale requise*.

Tâche	Description
Installez un pilote ODBC Cloudera pour Impala	Vous pouvez installer le pilote ODBC Cloudera pour Impala de Cloudera, Inc.
Configurez le pilote ODBC pour Impala	La configuration comprend la modification des fichiers .ini et la définition des chemins et des variables d'environnement. Suivez les instructions qui correspondent au pilote installé.
Importez et configurez le modèle de source de données du modèle BigDataODBCImpalaTemplate dans Unica Campaign.	Utilisez l'utilitaire `configTool` pour importer le modèle `BigDataODBCImpalaTemplate.xml` dans Unica Campaign. Allez ensuite dans `Campagne|partitions|partition[n]| dataSources` et créez une source de données basée sur le `BigDataODBCImpalaTemplate` importé.
Configurer SSH sur le serveur de programme d'écoute Unica Campaign	Pour permettre le transfert des fichiers de données entre le serveur (analytique) du programme d'écoute de Unica Campaign et l'instance big data Hadoop utilisant Hive, vous devez configurer la connexion transparente SCP et SSH.

Tâche	Description
Mapper la source de données Impala dans Campaign	Le mappage des tables utilisateur est le processus constituant à rendre les sources de données externes accessibles dans Unica Campaign.

Results

Vous pouvez désormais utiliser votre système Big Data Cloudera basé sur Impala comme source de données utilisateur pour Unica Campaign. Par exemple, vous pouvez créer un diagramme de campagne marketing qui utilise les données des comptes provenant de votre instance big data pour cibler les clients avec certains types de compte et solde.

Requirements and restrictions for Impala-based Hadoop data sources

The following requirements and restrictions apply to using Impala-based Hadoop data sources with Unica Campaign.

- The following drivers is required. The customer is responsible for obtaining the driver.
- Cloudera ODBC Driver for Impala Version 2.5.41.
- Minimum supported Impala version: 2.9.0.

- Impala based Hadoop user data source integration is supported with Cloudera.
- The big data integration is currently supported on Linux™ RHEL 6.3 or higher.
- Impala-based Hadoop is supported as a user data source only. It is not supported for Unica Campaign system tables.
- The big data integration does not currently support the Unica Campaign Cube, Optimize, or Interact List process boxes or Unica Deliver Landing Pages in an Extract process box.

A. Install a Cloudera Impala ODBC driver

This is the first step to integrate Unica Campaign with Impala-based Cloudera data sources. You must install the Cloudera Impala ODBC driver from Cloudera, Inc.

Install the Impala ODBC driver on the system where you have installed the Unica Campaign Analytical Server. If the Analytical Server is installed in a cluster, install the Impala ODBC driver on each system.

Installing the Cloudera Impala ODBC driver. This topic explains how to install the Cloudera Impala ODBC driver, which is a fully compliant ODBC driver that supports multiple Hadoop distributions.

Procedure

1.Obtain the Cloudera Impala ODBC Driver for version 2.5.41 at https://www.cloudera.com/downloads/connectors/impala/odbc/ . Download and save the Cloudera Impala ODBC driver on the Unica Campaign listener (analytic server). The default location is /opt/cloudera/

2. Install and configure the Cloudera driver. Installation instructions are downloaded to where you install the driver. By default, the instructions are downloaded to /opt/cloudera/impalaodbc/Cloudera-ODBC-Driver-for-Impala-Install-Guide.pdf. The PDF file is also available online at www.cloudera.com.

3. Download and install UnixODBC Manager, unixODBC-2.3.2 on the Unica Campaign listener (analytic server). Download UnixODBC Manager from the unixODBC Project at http://www.unixodbc.org/. The UnixODBC Manager is available in a gzip, tar format. The steps to install UnixODBC are available at http://www.unixodbc.org/download.html. To create a unixODBC directory that contains the required source files:

a. Copy the unixODBC-2.3.2.tar.gz file to the Unica Campaign listener (analyticserver).

b. gunzip unixODBC-2.3.2.tar.gz

c. tar xvf unixODBC-2.3.2.tar

4. On the Unica Campaign listener, run rpm -ivh ClouderaImpalaODBC-2.5.41.1029-1.el6.x86_64.rpm.

B. Installing the Impala ODBC driver for Windows

About this task

Obtain the Cloudera Impala ODBC Driver for version 2.5.42 at https://www.cloudera.com/ for Windows and install it.

To install the Cloudera ODBC Driver for Impala:

1. Depending on the configuration of your client application, double-click to run `ClouderaImpalaODBC32.msi` or `ClouderaImpalaODBC64.msi`.
2. Click `Next`.
3. Select the check box to accept the terms of the `License Agreement` if you agree, and then click `Next`.
4. To change the installation location, click the `Change` button, then browse to the desired folder, and then click `OK`. To accept the installation location, click `Next`.
5. Click `Install`.
6. When the installation completes, click `Finish`.

C. Configure the Impala ODBC Driver

This is the second step to integrate Unica Campaign with Impala-based Cloudera data sources. You must configure the driver that you installed (Cloudera Impala ODBC driver from Cloudera, Inc.).

Before you begin

Complete the Cloudera Impala ODBC driver installation that is described in "Installing the Cloudera ODBC driver"

Procedure

1. Modify the ODBC.ini file to specify the Impala connector information, using the values shown in the following example. Be sure to customize the items in bold to match your own configuration.

Note: Do not enable Native Query mode in the configuration.

[Cloudera Impala]

Description=Cloudera ODBC Driver for Impala (64-bit) DSN

Driver=/opt/cloudera/impalaodbc/lib/64/libclouderaimpalaodbc64.so

HOST=<Impala Host>

PORT=<Impala port - default is 21050>

Database=<Database name>

Schema=<Database schema>

AuthMech=3

UseSASL=0

UID=<impala user id>

PWD=<password>

SSL=0

TSaslTransportBufSize=1000

RowsFetchedPerBlock=10000

SocketTimeout=0

StringColumnLength=32767

UseNativeQuery=1

Note: The UID is the user ID of the Cloudera cluster database.

2. Assuming that your ODBC driver is installed in the following location /opt/cloudera/impalaodbc/lib/64/libclouderaimpalaodbc64.so

Make the following change in Campaign/bin/setenv.sh:

Set LD_LIBRARY_PATH (LIBPATH for AIX) to include Cloudera LIB and UnixODBC LIB as follows.

LD_LIBRARY_PATH=/opt/HCL/Campaign/bin:/opt/ibm/db2/V10.1/lib64/:/usr/lib:/lib:**opt/cloudera/impalaodbc/lib/64/:/usr/local/lib/**

export LD_LIBRARY_PATH

ODBCINI=/usr/local/etc/odbc.ini # You can use the ODBC INI from Cloudera Impala ODBC Driver install location as well

export ODBCINI

C. Import and configure the BigDataODBCImpalaTemplate data source template in Unica Campaign

This is the third step to integrate Unica Campaign with Impala-based Cloudera Hadoop data sources.

About this task

To enable Unica Campaign to communicate with your Impala-based Hadoop system, you must do the following actions:

- Import the BigDataODBCImpalaTemplate.xml template into Unica Campaign. You must import the template only once. Importing a template makes it available for creating data sources.

1. Use the `configTool` utility to import the `BigDataODBCImpalaTemplate.xml` template into Unica Campaign.
   ◦ `BigDataODBCImpalaTemplate.xml` is in `<Campaign_Home>/conf`.
   ◦ `configTool` is in `<Marketing_Platform_Home>/tools/bin`. For more information, see the *Unica Platform Administrator's Guide*.

   The following example imports the template into the default Unica Campaign partition, `partition1`. Replace `<Campaign_Home>` with the complete path to the Unica Campaign installation directory.

   ```
   ./configTool -i -p "Affinium|Campaign|partitions|partition1|dataSources" -f
   <Campaign_Home>/conf/BigDataODBCImpalaTemplate.xml
   ```

2. Create a data source based on `BigDataODBCImpalaTemplate`.
   Example
   a. In Unica Campaign, choose **Settings > Configuration**
   b. Go to `Campaign|partitions|partition[n]|dataSources`.
   c. Select `BigDataODBCImpalaTemplate`.
   d. Complete the fields to set the properties for the new data source, then save your changes.

   Important: Some properties do not have default values, so you must supply them. Pay special attention to the properties described below. This is only a partial list of the properties included in this template. For complete information, see the *Unica Campaign Administrator's Guide*.

Configuration property	Description
ASMUserForDBCredentials	No default value defined. Specify the Unica Campaign system user.
DSN	DSN Name as specified in the odbc.ini file for the Hive-based Hadoop big data instance.
JndiName	Not needed for user data source.
SystemTableSchema	No default value defined. Specify the user of the database that you connect to.
OwnerForTableDisplay	No default value defined. Specify the user of the database that you connect to.
LoaderPreLoadDataFileCopyCmd	SCP is used to copy data from Unica Campaign to any location on the Impala-based Hadoop system. The location must a valid HDFC location on the Impala server. This value can either specify the SCP command or call a script that specifies the command. For more information and detailed instructions about how to export data from Unica Campaign to a Impala-based Hadoop system, see the Unica Campaign Administrator's Guide.
LoaderPostLoadDataFileRemoveCmd	Data files are copied from Unica Campaign to a temp folder on the Hive-based Hadoop system. You must use the SSH "rm" command to remove the temporary data file.

Configuration property	Description
	For more information and detailed instructions about how to export data from Unica Campaign to a Impala-based Hadoop system, see the Unica Campaign Administrator's Guide.
LoaderDelimiter	No default value defined. Specify the delimiter such as comma (,) or semi-colon (;) that separates fields in the temporary data files that are loaded into the big data instance. Tab (/t) is not supported. The delimiter value must match the ROW format delimiter that was used when the big data database table was created. In this example, a comma is used: ROW FORMAT DELIMITED FIELDS TERMINATED BY '; ;"
SuffixOnTempTableCreation SuffixOnSegmentTableCreation SuffixOnSnapshotTableCreation SuffixOnExtractTableCreation SuffixOnUserBaseTableCreation SuffixOnUserTableCreation	No default value defined. Use the same character as specified for LoaderDelimiter.
UseExceptForMerge	Set to FALSE. Hive does not support the EXCEPT clause, so a setting of TRUE can result in process failures.
DateFormat DateTimeFormat DateTimeOutputFormatString	All Date strings must use the dash "-" character to format dates. Hive does not support any other characters for dates. Example: %Y-%m-%d %H:%M:%S
Type	BigDataODBC_Impala
UseSQLToRetrieveSchema	Set to FALSE.

DataFileStagingFolder	Default location value is set to `/tmp`. You can change the location value. Example: `/opt/campaign/`
	
	Note: The value for this folder must have a trailing slash.
	
	If you have written shell script to copy the Unica Campaign data file to the Hive server, you need to modify it.
	
	Example:
	
	<pre>#!/bin/sh
	ssh <USER>@<HADOOP NODE WITH IMPALA> "mkdir -p $2" #Operating system location
	scp $1 <USER>@<HADOOP NODE WITH IMPALA>:$2 # SCP data file to Impala Server
	filename=$(basename "$1")
	extension="${filename##*.}"
	filename="${filename%.*}"
	#Moving Unica Campaign data file with name containing special characters is prohibited, so
	 extracting extension and file names
	ssh <USER>@<HADOOP NODE WITH IMPALA> "mv $2/`basename $1` $2/$filename"
	ssh <USER>@<HADOOP NODE WITH IMPALA> "hadoop fs -put -f $2/$filename $2"
	ssh <USER>@<HADOOP NODE WITH IMPALA> "hadoop fs -mv $2/$filename $2/$filename.$extension"
	ssh <USER>@<HADOOP NODE WITH IMPALA> "hadoop fs -chmod 0777 $2/$filename.$extension"</pre>
	If you are using `LoaderPreLoadDataFileCopyCmd`, then you need to update the file location. Example:
	
	<pre>scp <DATAFILE> <USER>@[hostname]:/opt/campaign/</pre>

Configuration property	Description
	If you are using `LoaderPostLoadDataFileRemoveCmd`, then you need to update the file location. Example: `ssh <USER>@<HADOOP NODE WITH IMPALA> "hadoop fs -rm $2/`basename $1`"`
DeleteAsRecreate	Please set this configuration parameter value to FALSE
DeleteAsTruncate	Please set this configuration parameter value to TRUE

What to do next

D. Configure SSH on the Unica Campaign listener server

This is the fourth step to integrate Unica Campaign with Impala-based Cloudera Hadoop data sources.

About this task

To enable data file transfers between the Unica Campaign listener (analytic) server and the Impala-based Hadoop big data instance, you must configure SCP and SSH seamless login. SSH allows secure connection between two computers; most systems use the OpenSSH client.

1. On the machine that is running the Unica Campaign listener, configure SSH for no password prompt for authentication. Log in as the user who is running the Unica Campaign listener and run the following commands, substituting the **username@IP address** of your Impala server (in this example, the MapR machine):
   Example

   >> ssh-keygen -t rsa

   >> ssh **cloudera@192.0.2.0** mkdir -p .ssh

   >> cat .ssh/id_rsa.pub | ssh **cloudera@192.0.2.0** 'cat >> .ssh/authorized_keys'

   >> ssh **cloudera@192.0.2.0** "chmod 700 .ssh; chmod 640 .ssh/authorized_keys"

2. Verify password-less authentication using RSA-based authorized keys. Run each command, substituting the actual **username@IP address**, and verify that it works. You need a local file called test1 for this test to work:
   Example

   >> ssh **cloudera@192.0.2.0**

   >> scp test1 **cloudera@192.0.2.0**:/tmp

   >> ssh **cloudera@192.0.2.0** "rm /tmp/test1"

What to do next

F. Configuring connectivity between Unica Campaign listener server and Hadoop cluster

About this task

The loader Script is needed to copy temporary data file at `<Campaign_Home>/partitions/partition[n]/tmp` to Impala Server which is created during flowchart execution.

1. We are making use of pscp (Putty utility) to connect to Cloudera Hadoop Cluster from Unica Campaign server and upload the data files on Hadoop.

   Note: Here the PSCP utility needs password in argument in plain text.

2. Configuration for LoaderPreLoadDataFileCopyCmd: LoaderPreLoadDataFileCopyCmd = `C:\Unica11\Campaign\bin\CopytoHadoop.bat <DATAFILE> /tmp/`

3. Configuration for LoaderPostLoadDataFileRemoveCmd: LoaderPreLoadDataFileCopyCmd = `C:\Unica10\Campaign\bin\removeFromHadoop.bat <DATAFILE>`

4. Configuration for LoaderPostLoadDataFileRemoveCmd = `LoaderPostLoadDataFileRemoveCmd= C:\Unica10\Campaign\bin\removeFromHadoop.bat <DATAFILE> /tmp/`

Example

This is a sample script . You can write your own batch script:

```
Sample Script of copytoHadoop.bat:
==========================================
REM This script is used to copy a file from Campaign to any location on the
Impala-based Hadoop system. The location must a valid HDFS location
@echo off
REM Operating System location
pscp -pw <password> "%1" <username>@<Hadoop Node with Impala>:"%2"
set filepath="%1"
for /F "delims=" %%i in (%filepath%) do set basename="%%~nxi"
for /F "delims=" %%i in (%filepath%) do set filename="%%~ni"
for /F "delims=" %%i in (%filepath%) do set fileExtension="%%~xi"
REM Moving Campaign data with name containing special character is prohibited,
 so extracting
plink -pw <password> -ssh <username>@<Hadoop Node with Impala> "mv %2/%baseName% %2/%fileName%"
plink -pw <password> -ssh <username>@<Hadoop Node with Impala> "hadoop fs -put -f %2/%fileName% %2"
plink -pw <password> -ssh <username @<Hadoop Node with Impala> "hadoop fs -mv
%2/%fileName% %2/%baseName%"
plink -pw <password> -ssh <username @ Hadoop Node with Impala> "hadoop fs -chmod 0777 %2/%baseName%
"==========================================
```

Data files are copied from Unica Campaign to a temp folder on the Imapala-based Hadoop system. You must use the SSH `"rm"` command to remove the temporary data file.

1. Remove temp data from Hadoop file System.
2. Remove temp data from OS file System.

📝 **Note:**

1. For `LoaderPostLoadDataFileCopyCmd` and `LoaderPostLoadDataFileRemoveCmd` to work, the Hadoop System should have TGT and have access to write, modify and remove files from Hadoop File System.
2. The above configuration is Sample configuration, you can refer *Unica Campaign Installation Guide* for details.

G. Map the Impala datasource in Unica Campaign

This is the Fifth and final step to integrate Unica Campaign with Impala-based Hadoop data sources.

About this task

Mapping user tables makes external data sources accessible in Unica Campaign. A typical user table contains information about your company's customers, prospects, or products, for use in marketing campaigns. You must map any data sources that you configured to make that data accessible to processes in flowcharts.

1. Select **Settings > Campaign Settings > Manage Table Mappings**.
2. In the Table Mappings dialog, click **Show User Tables**.
3. Click **New Table**. The New Table Definition dialog opens.
4. Click **Next**.
5. Select **Map to Existing Table in Selected Database**.
6. Select the **BigDataODBCImpala** datasource, then click **Next**.
7. Follow the prompts to map the table, as explained in the *Unica Campaign Administrator's Guide*.

What to do next

You have completed the installation and configuration process. Unica Campaign flowchart designers can now select Impala-based Hadoop data sources in processes. For example, a Select process can query a Impala-based data source to target customer data for a marketing campaign.

Preparing Cloudera BigData instance using Kerberos authentication

About this task

With the v11.1 release, Unica Campaign supports Cloudera based BigData instance using Kerberos authentication. The Unica Campaign Analaytical Server connects with the Kerberos enabled Cloudera BigData instance with the valid Kerberos enabled user login or with user impersonation.

Note: In case the kerberos authentication token expires, user will not able to connect to the Bigdata instance. Also, the flowcharts execution will fail due to the authentication issue. User must make sure that the authentication using kerberos is seamless using TGT while Unica Campaign is running.

The Unica Campaign analytical should always be connected with the BigData instance - if the Operating System user is not Kerberos enabled, there is a need to authenticate this user using Kerberos keytab.

Note: In case of kerberos authentication token expired, user will not able to connect to Bigdata instance. Also flowcharts execution will get failed due to authentication issue. User needs to make sure authentication using kerberos is seamless using TGT while Unica Campaign is running.

Note: Kerberos authentication is currently only certified for RHEL Linux and Windows operating system.

Table 6.

Task	Description
Download, Install and Configure Kerberos packages on Campaign listener machine	You can install and configure the required Kerberos packages on Windows and Linux operating systems as mentioned in the respective section. On Windows MIT Kerberos packages should be installed.
Establish seamless connection with Cloudera Hadoop instance using Kerberos on Unica Campaign listener machine	On Windows a script is provided to have a seamless connection with Cloudera Hadoop instance. This Script obtains the Ticket for the Kerberos principal , renews and regenerates the ticket before expiry. On Linux environment user is recommended to write cron jobs to achieve using kinit and k5start utility.
Install Cloudera Impala ODBC driver.	Please follow the steps to install and configure ODBC driver as mentioned in section – (Préparation des sources de données Apache Hadoop Impala pour Unica Campaign on page 55).
Configure Cloudera Impala ODBC Driver using Kerberos.	Configuration include setting hostname, port number, schema, authentication mechanism as Kerberos.
Import and Configure the BigDataODBCImpala template Data Source in Campaign	Please follow the steps to configure BigDataODBCImpala template as mentioned in section (Préparation des sources de données Apache Hadoop Impala pour Unica Campaign on page 55).
Map the Impala Data Source in Campaign	Please follow the steps to map Impala datasource in Unica Campaign as mentioned in section (Préparation des sources de données Apache Hadoop Impala pour Unica Campaign on page 55).

Installing and configuring MIT Kerberos for Windows

Downloading and installing MIT Kerberos for Windows 4.0.1

About this task

For information on Kerberos and download links for the installer, see the MIT Kerberos website: http://web.mit.edu/kerberos/. To download and install MIT Kerberos for Windows 4.0.1:

1. Download the appropriate Kerberos installer:
   Choose from:

      ◦ For a 64-bit machine, use the following download link from the MIT Kerberos website:http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-amd64.msi.

      ◦ For a 32-bit machine, use the following download link from the MIT Kerberos website:http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-i386.msi.

2. To run the installer, double-click the .msi file that you downloaded above.

3. Follow the instructions in the installer to complete the installation process.

4. When the installation completes, click **Finish**.

Setting up the Kerberos configuration file

Settings for Kerberos are specified through a configuration file. You can set up the configuration file as an .ini file in the default location, which is the `C:\ProgramData\MIT\Kerberos5` directory. Normally, the `C:\ProgramData\MIT\Kerberos5` directory is hidden. For information about viewing and using this hidden directory, refer to *Microsoft Windows documentation*. **Note**: For more information on configuring Kerberos, refer to the *MIT Kerberos documentation*.

About this task

To set up the Kerberos configuration file:

1. Obtain a `krb5.conf` configuration file. You can obtain this file from your Kerberos administrator, or from the `/etc/krb5.conf` folder on the machine that is hosting the Impala server.

2. Rename the configuration file from `krb5.conf` to `krb5.ini`.

3. Copy the `krb5.ini` file to the `C:\ProgramData\MIT\Kerberos5` directory and overwrite the empty sample file.

Setting up the Kerberos credential cache file

Kerberos uses a credential cache to store and manage credentials.

To set up the Kerberos credential cache file:

1. Create a directory where you want to save the Kerberos credential cache file. For example, create a directory named `C:\temp`.

2. Open the System window:

Choose from:

      ◦ If you are using Windows 7 or earlier, click `Start (Windows icon)`, then right-click `Computer`, and then click `Properties`.

      ◦ Or, if you are using Windows 8 or later, right-click `This PC` on the `Start` screen, and then click `Properties`.

3. Click `Advanced System Settings`.

4. In the `System Properties` dialog box, click the `Advanced` tab and then click `Environment Variables`.

5. In the `Environment Variables` dialog box, under the `System Variables` list, click `New`.

6. In the `New System Variable` dialog box, in the `Variable Name` field, type `KRB5CCNAME`.

7. In the `Variable Value` field, type the path to the folder you created above, and then append the file name `krb5cache`. For example, if you created the folder `C:\temp`, then type `C:\temp\krb5cache`.

   Note: `krb5cache` is a file (not a directory) that is managed by the Kerberos software, and it should not be created by the user. If you receive a permission error when you first use Kerberos, make sure that the `krb5cache` file does not already exist as a file or a directory.

8. Click `OK` to save the new variable.
9. Make sure that the variable appears in the `System Variables` list.
10. Click `OK` to close the `Environment Variables` dialog box, and then click `OK` to close the `System Properties` dialog box.
11. To make sure that Kerberos uses the new settings, restart your machine.

Downloading and installing Kerberos Packages for Linux

Enable Kerberos authentication on Cloudera Bigdata for Linux Server.

1. Install Kerberos Packages on Linux, if they don't already exists. Below packages should be installed on Unica Campaign listener machine
   Example
   a. krb5-lib
   b. libkadm
   c. krb5-devel
   d. krb5-workstation
2. Contact your Kerberos administrator and configure /etc/krb5.conf on machine where Campaign listener is running.

Seamless Kerberos connection on Windows

About this task

You need to run unica_kerbKeyGenRenew.bat script to obtain the Ticket for the Kerberos principal. The purpose of this script is to run continuously and :

1. Renew the TGT before its renewable time.
2. Regenerate the token before expiry.

By default, it renews the ticket every 24 hours and regenerates a new ticket after 7 days. You can also override the default setting for regenerating and renewing the tickets by setting environment variables.

Prerequisite for running the Script

1. Requires MIT Kerberos to be installed and configured.
2. `Keytab` file is already present for the said principal.
3. The following environment variables are set:
   a. `TGT_EXPIRY_DAYS`: number of days after which this TGT will expire, and can no longer be renewed (specify value in days)
   b. `TGT_RENEWAL_HOURS`: number of hours within which this TGT needs to be renewed (specify value in hours)
   c. `KRB5CCNAME`: path to the cache file
   d. `KEYTAB_FILE_PATH`: path to keytab file for this principal

4. From command line run "`Where kinit`" command should point `MIT Kerberos`

   For example: `C:\Kerbores\Campaign\bin>where kinit`

   `C:\Program Files\MIT\Kerberos\bin\kinit.exe`

5. Copy the `Keytab` file to the Campaign System for the principal you want to authenticate.

Procedure to run the Script

1. Make sure the prerequisites are satisfied.

2. Then run the following:

   `<campaign_home>/bin/unica_kerbKeyGenRenew.bat <principal>`

   For example: `unica_kerbKeyGenRenew.bat impala/quickstart.cloudera@CLOUDERA`

3. The script will start running, and do the following:

   a. Generate the TGT.

   b. Check every 1 minute for renewal and regeneration of the TGT.

   c. Using the values in TGT_EXPIRY_DAYS and TGT_RENEWAL_HOURS, keep checking for renewal or expiry.

   d. Before renewal time - renew the TGT.

   And before expiry time - regenerate the TGT.

   Note:

   a. TGT_RENEWAL HOURS AND TGT_EXPIRY_DAYS values needs to be same as Kerberos Server configuration. Please contact the Kerberos administrator to get the values.

   b. By default Script sets 24 hours for TGT renewal and 7 days for TGT_EXPIRY.

   Possible errors:

   1. Usage

   2. Principal is incorrect

   3. Keytab file is not present for the Principal

   4. One or more of the prerequisites is not set

   5. Keytab file is not valid

Seamless Kerberos connection on Linux

About this task

You need to create 2 cron jobs using kinit and k5start utility. These utility perform following operations

1. Renew the TGT before its renewable time. (k5start)
2. Regenerate the token before expiry.(kinit)

1. Renew TGT using the k5start utility. Add this to cron. . Please contact the Kerberos administrator to get the values.

   a. Establish the Cloudera BigData instance authentication using the Kerberos keytab file:

   ```
   k5start -f <KEYTABFILE> -K <minutes> -l <time> [-u <PRINCIPAL>]
   [-v]
   You can add this command in the corn job and make sure that the kerberos token is always
   alive.
   ```

   b. See https://linux.die.net/man/1/k5start

   ```
   c. k5start -f <KEYTABFILE> -K <minutes> -l <time> [-u <PRINCIPAL>] [-v]
   -f keytab
   Authenticate using the keytab rather than asking for a password. A key for the client principal must be
    present in keytab.

   -K minutes
   Run in daemon mode to keep a ticket alive indefinitely. The program reawakens after minutes minutes,
    checks if the ticket will expire before or less than two minutes after the next scheduled check, and
    gets a new ticket if needed. If this option is not given but a command was given on the command line,
    an interval appropriate for the ticket lifetime will be used.

   -l time string
   Set the ticket lifetime. time string should be in a format recognized by the Kerberos libraries for
    specifying times, such as "10h" (ten hours) or "10m" (ten minutes). Known units are "s", "m", "h", and
    "d"

   -v
   Be verbose. This will print out a bit of additional information about what is being attempted and what
    the results are.
   ```

   ```
   For example:
   k5start -f <FilePath>/impala.keytab -K 1 -l 3m impala/quickstart.cloudera@CLOUDERA
   k5start -f <FilePath>/impala.keytab -K 1 -l 3m impala/quickstart.cloudera@CLOUDERA -v
   ```

2. Regenerate the Token using kinit utility. Add to cron so that ticket gets regenerated before it expires.

   ```
    kinit -k -t <keytab_filename> <principal>
   Refer https://linux.die.net/man/1/kinit .
   -k [-t keytab_file]
   ```

   requests a ticket, obtained from a key in the local host's keytab file. The name and location of the keytab file may be specified with the -t keytab file option; otherwise the default name and location will be used. By default a host ticket is requested but any principal may be specified. On a KDC, the special keytab location KDB: can be used to indicate that kinit should open the KDC database and look up the key directly. This permits an administrator to obtain tickets as any principal that supports password-based authentication.

   e.g kinit -k -t /root/impala.keytab impala/quickstart.cloudera@CLOUDERA

Configuring the Impala-based Cloudera data sources using Kerberos

1. In the `ODBC Data Source Administrator`, click the `Drivers` tab and verify that the `Cloudera ODBC Driver for Impala` appears in the list of `ODBC` drivers that are installed on your system.
2. In the `Create New Data Source` dialog box, select `Cloudera ODBC Driver for Impala` and then click `Finish`.
3. Use the options in the `Cloudera ODBC Driver for Impala DSN Setup` dialog box to configure your DSN:
   a. In the `Data Source Name` field, type a name for your DSN.
   b. Optionally, in the `Description` field, type relevant details related to the DSN.
   c. In the `Host` field, type the IP address or hostname.
   d. In the `Port` field, type the listening port for the Impala service.
      Note: The default port number for the Impala service is 21050.
   e. In the `Database` field, type the name of the database schema to use.

What to do next

In the Authentication area, configure authentication as Kerberos:

1. Realm: Type the Kerberos realm of the Impala server host in the `Realm` field.
2. In the `Host FQDN` field, type the fully qualified domain name of the Impala host.
3. In the `Service Name` field, type the service name of the Impala server. For example, if the principle for the Impala server is `impala/fully.qualified.domain.name@yourrealm.com`, then the value in the service name field is `impala`. If you are unsure of the correct service name to use for your particular Hadoop deployment, contact your Hadoop administrator

Note: Test the connection to make sure Connection is working fine with Kerberos enabled.

Configuring the Impala-based Cloudera data sources using Kerberos on Linux

About this task

Configuring ODBC datasource is same as of mentioned under section (Préparation des sources de données Apache Hadoop Impala pour Unica Campaign on page 55). User will need to configure Cloudera Impala ODBC.ini file to authenticate using Kerberos authentication. User will need to add the Kerberos authentication/connection information for the below mentioned parameters in ODBC.ini file.

```
AuthMech=1 #which mention to connect using Kerberos authentication
# Kerberos related settings with sample values.
KrbRealm=CLOUDERA
KrbFQDN=quickstart.cloudera
KrbServiceName=impala
```

Import and Configure the BigDataODBCImpala template Data Source in Unica Campaign

Please follow the steps to configure BigDataODBCImpala template as mentioned in section (Préparation des sources de données Apache Hadoop Impala pour Unica Campaign on page 55)

> **Note:** 1. All the Unica Campaign users should use the same Kerberos principal for authentication.
> 2. Please configure dataSource name with a dummy user name and password because Kerberos uses TGT and does not require username and password. Restart the Unica Campaign Web application and Unica Campaign listener.

Map the Impala Data Source in Unica Campaign

Mapping user tables is the process of making external data sources accessible in Unica Campaign

Troubleshooting

About this task

If the loaders fail while executing the flowchart:

1. Verify if you have access and permission to the Hadoop file System on which Kerberos is enabled. You can use Hadoop commands to verify the same.
2. For Windows, verify if `unica_kerbKeyGenRenew.bat` is running on the Unica Campaign Server.
3. Verify `odbctest.exe` to see if the connection works. If the `odbctest.exe` fails, implies that there is some issue with the environment / configuration.

Preparing Actian Vector user data source for Campaign

On Windows

1. Install ODBC driver:
   - Open link -> https://esd.actian.com/product/Vector
   - Choose appropriate driver and download .zip format file. Please note that 'community' type includes both db server as well as client.
   - Unzip it and Install driver by running '<extracted_path>\setup.exe' file with default options.
   - Confirm that 'Actian Vector [VW]' db server service is running under windows service lists.
   - At installation location you will find below directories:
     - Director - the actian vector db client
     - Vector Documentation X.X
     - VectorVW - the actian vector db server
2. Use Director db client to connect to server instance
   - Run<ActianInstallationPath>\Director\director.exe client

   - For a new connection to the actian instance running on your machine, Select **Connection** tab and click on **Connect** option.

   - On 'Connect to instance' dialog
   - Use **hsedemo** as password and click on **Connect** button.

> **Note:** On your machine Instance name should be different, it should match with your machine's hostname.

- You can see that its connected successfully to actian db instance running on your machine.
3. Create ODBC DSN
    - Open ODBC Data Source Administratior and click on **Add...** button.
        ▪ You can see that installed Ingres and Ingres VW actian db odbc drivers in the list.
    - Select **Ingres VW** from the list and click on **Finish** button.
    - Fill all the required information, click on **Apply** button and then click on **Test** button. It will allow you to fill only Data Source and Database field values.

        > **Note:** Here it will take user name and password of currently logged in user.

    - If you tried to change **Host Name** field value it will ask you to provide **Listen Address** field value too. Valid values for **Host Name** fields are localhost / <machine hostname> / (LOCAL) Valid value for **Listen Address** field is VW / II. This time it will ask you to enter user name and password on clicking **Test** button.
    - Enter User Id: demo and Password: hsedemo. Click on **OK** button.
4. Create Campaign Data Source
    - Use SQLServerTemplate.xml as there is no separate template created for Actian Vector database.
    - Import above mentioned .xml file in Campaign using config tool and set **DisallowTempTableDirectCreate = TRUE**. This is important.
    - Fill other fields as per requirements.

On Linux

> **Note:** Use actian db server installed on windows machine using above steps and install actian db client on linux machine using, actian-vector-client-6.0.0-129-free-linux-ingbuild-x86_64.tgz.

1. Install ODBC driver
    - Download actian-vector-client-6.0.0-129-free-linux-ingbuild-x86_64.tgz and run following commands:
        ▪ $>tar -xf actian-vector-client-6.0.0-129-free-linux-ingbuild-x86_64.tgz
        ▪ $>cd actian-vector-client-6.0.0-129-free-linux-ingbuild-x86_64
        ▪ Swtich to root user
        ▪ $>sudo su
        ▪ $>./client_install.sh <ENTER>
        ▪ Answer to the asked questions accordingly.
2. Create ODBC DSN
    - Edit odbc.ini file and add DSN for actian db in it as shown in following example [abc]

        Driver=/opt/Actian/Vector_Client/ingres/lib/libiiodbcdriver.1.so

Description=

Vendor=Actian Corporation

DriverType=Ingres

HostName=< machine name on which actian db server is running >

ListenAddress=< running instance name of actian db server >

Database=< db name >

ServerType=ACTIAN

3. Create Campaign Data Source
   ◦ Use SQLServerTemplate.xml as there is no separate template created as such for Actian Vector database.
   ◦ Import above mentioned .xml file in Campaign using config tool and set **DisallowTempTableDirectCreate = TRUE**. This is important.
   ◦ Fill other fields as per requirements.

   Note: Before running campaign listener on LINUX platform, please run following command on same console/terminal where you want to run listener. Below commands need not be run on windows platform

   ▪ $>cd /opt/Actian/Vector Client/ingres
   ▪ $>. ./.ingCLsh
   ▪ $>cd <CAMPAIGN_HOME>/bin
   ▪ $>./rc.unica_ac start

Chapter 4. Installing Unica Campaign and Unica Optimize

You must run the Unica installer to start the installation of Unica Campaign and Unica Optimize. The Unica installer starts the product installer during the installation process. Make sure that the Unica installer and the product installer are saved at the same location.

Each time that you run the Unica suite installer, you must first enter database connection information for the Unica Platform system tables. When the Unica Campaign installer starts, you must enter the required information for Unica Campaign.

After installing Unica Campaign and Unica Optimize, you can create an EAR file for your product, and you can install the reports package for your product. Creating the EAR file and installing the reports package are not mandatory actions.

> **Important:** Before you install Unica Campaign and Unica Optimize, execute - slibclean command using root credentials if you are installing on AIX operating system.

> **Note:** If you are planning to implement SSL in Campaign application. See the Platform Administrator guide> Implementation of one-way SSL section for more details.

Installation files

The installation files are named according to the version of the product and the operating system on which they must be installed, except UNIX™. For UNIX™, different installation files exist for the X Window System mode and the console mode.

Example
The following table displays examples of the installation files that are named according to the product version and the operating system:

> **Note:** For Suse Linux operating system use the corresponding product Linux build except for Unica Campaign.

> **Note:** : EAR deployments are not supported on Tomcat application server

Table 7. Installation files

Operating system	Installation file
Windows™: GUI and console mode	`Product_N.N.N.N_win64.exe`, where *Product* is the name of your product, *N.N.N.N* is the version number of your product, and Windows™ 64-bit is the operating system on which the file must be installed. `Product_N.N.N.N_win.exe`,

Table 7. Installation files

(continued)

Operating system	Installation file
	where *Product* is the name of your product, *N.N.N.N* is the version number of your product, and Windows™ 64-bit is the operating system on which the file must be installed.
UNIX™: X Window System mode	`Product_N.N.N.N_linux.bin`, where *Product* is the name of your product, and *N.N.N.N* is the version number of your product.
	`Product_N.N.N.N_linux.bin`, where *Product* is the name of your product, and *N.N.N.N* is the version number of your product.
	`Product_N.N.N.N_linuxrhel64.bin`, where *Product* is the name of your product, and *N.N.N.N* is the version number of your product.
UNIX™: Console mode	`Product_N.N.N.N.bin`, where *Product* is the name of your product, and *N.N.N.N* is the version number of your product. This file can be used for installation on all UNIX™ operating systems.
SUSE Linux	`Product_N.N.N.N_linuxsuse64.bin`, where *Product* is the name of your product, and *N.N.N.N* is the version number of your product. This file can be used for installation on all ™SUSE Linux operating systems.

Installing Unica Campaign and Unica Optimize by using the GUI mode

For Windows™, use the GUI mode to install Unica Campaign and Unica Optimize. For UNIX™, use the X Window System mode to install Unica Campaign.

Before you begin

⚠️ **Important:** Before you use the GUI mode to install Unica Campaign and Unica Optimize, ensure that the available temporary space on the computer where you install Unica Campaign and Unica Optimize is more than three times the size of the Unica Campaign installer. The Unica Campaign installer installs both Unica Campaign and Unica Optimize.

Make sure that the Unica installer and the Unica Campaign installers are in the same directory on the computer where you want to install Unica Campaign.

Complete the following actions to install Unica Campaign by using the GUI mode:

1. Go to the folder where you have saved the Unica installer and double-click the installer to start it.
2. Click **OK** on the first screen to see the **Introduction** window.

3. Follow the instructions on the installer and click **Next**.

   Use the information in the following table to take the appropriate actions on each window in the Unica installer.

Table 8. Unica Installer

Window	Description
Introduction	This is the first window of the Unica suite installer. You can open the Campaign installation and upgrade guides from this window. You can also see a link for the installation and upgrade guides for the products whose installers are saved in the installation directory.
	Click **Next**.
Response Files Destination	Click the **Generate Response File** check box if you want to generate response files for your product. Response files store the information that is necessary for the installation of your product. You can use response files for an unattended installation of your product, or to get pre-filled answers if you rerun the installer in GUI mode
	Click **Choose** to browse to a location where you want to store the response files.
	Click **Next**.
Unica Products	In the **Install Set** list, select **Custom** to select the products that you want to install.
	In the **Install Set** area, you see all of the products whose installers are in the same directory on your computer.
	The **Description** field shows the description of the product that you select in the **Install Set** area.
	Click **Next**.
Installation Directory	In the **Specify the installation directory** field, click **Choose** to browse to the directory where you want to install your product.
	If you want to install the product in the folder where the installers are stored, click **Restore Default Folder**.

Window	Description
	Click **Next**.
Select Application Server	Select one of the following application servers that you configured for Campaign installation: ◦ WebSphere® ◦ Oracle WebLogic ◦ Jboss ◦ Tomcat Click **Next**.
Platform Database Type	Select the appropriate Unica Platform database type. Click **Next**.
Platform Database Connection	Enter the following information about your database: ◦ Database host name ◦ Database port ◦ Database name or System ID (SID) ◦ Database user name ◦ Database password Click **Next**.
Platform Database Connection (continued)	Review and confirm the JDBC connection. Click **Next**. The URL can be customized with additional parameters if needed.
Preinstallation Summary	Review and confirm the values that you added during the installation process. Click **Install** to start the installation process. The Campaign installer opens.

4. Follow the instructions on the Unica Platform installer to install or upgrade Unica Platform. See *Unica Platform Installation Guide* for more information.
5. In the **Installation Complete** window, click **Done**.
   Result
   The Unica Platform installation is complete and the Campaign installer opens.

6. Use the information in the following table to navigate the Campaign installer. In the **Platform Database Connection** window, enter all the required information and click **Next** to start the Campaign installer.

Table 9. Unica Campaign Installer GUI

Window	Description
Introduction	This is the first window of the Campaign installer. You can open the Campaign installation and upgrade guides from this window. Click **Next**.
Software License Agreement	Carefully read the agreement. Use **Print** to print the agreement. Click **Next** after you accept the agreement.
Installation Directory	Click **Choose** to browse to the directory where you want to install Campaign and Optimize. Click **Next**.
Components	Select the components that you want to install. When you select the components, the installer displays information about the components. A new component is added as Reports. This component installs Unica Insights Report files for Unica Campaign reports. Click **Next**.
Single or Multiple Listeners	Select **Non-clustered listener configuration** (single node) **Note:** If you are doing a clustered listener installation, see Appendix B on page 173 for installation instructions. Click **Next**.
Listener Node Selection	Listener Node selection provides three options:

Window	Description
	◦ Run Campaigns only ◦ Run Optimize Sessions only ◦ Run Campaigns and Optimize session both Select any of these options and click Next.
Default locale	Select a default locale for your installation. English is selected by default. Click **Next**.
Campaign Database Setup	Select one of the following options for setting up the Campaign database: ◦ Automatic database setup ◦ Manual database setup If you select **Automatic database setup**, select **Run Unicode SQL Script** if your system tables are configured for Unicode. Click **Next**.
Campaign Database Type	Select the appropriate database type. Click **Next**.
Campaign Database Connection	Enter the following details for the Campaign database: ◦ Database host name ◦ Database port ◦ Database system ID (SID) ◦ Database user name ◦ Password ⚠ **Important:** If Unica products are installed in a distributed environment, you must use the machine name rather than an IP address in the navigation URL for all of the applications in the suite. Also, if you are on a clustered environment and choose to use ports that are different from the default ports 80 or 443 for your deployment,

Window	Description
	do not use a port number in the value of this property.
	Click **Next**.
JDBC Connection	Review and confirm the JDBC connection.
	Click **Next**.
Campaign Connection Settings	Enter the following connection settings:

Enter the following connection settings:

- Network domain name

  Note:

  When you add the network domain name, you might see the following message:

  ```
  Warning-Server name includes
  domain name, final URL contains
  several occurrences of domain
  name
  ```

  Select **Modify** to change the domain name or click **Cancel** to cancel the message.

- Host name
- Port number - If you have load balancer or HTTP server, and during installation you specify host name of that and do not left the port empty.

Select the **Use secure connection** check box if necessary.

Click **Next**.

Platform Utilities Settings

Review and confirm or change the following information for connecting to the Unica Platform database:

- JDBC driver class
- JDBC connection URL
- JDBC driver classpath

Window	Description
	Click **Next** to validate your entries and to go to the next window.
Datasource creation	The installer can create the JDBC data source for one partition in your web application server (WebSphere® or WebLogic). This data source is required to enable the Campaign web application to connect to the Campaign system tables. You have the option to skip this step in the installer and create the data source in the administration console of your web application server after installation is complete.
	If you select the **Create Campaign Datasource** check box, the installer creates the data source using information you provide.
	◦ Your application server must be running. ◦ Enter any name for the datasource.
	The installer creates the JNDI name automatically (campaignPartition1DS).
	◦ Enter the directory in which the application server is installed.
	For WebSphere®, this is the directory that contains the `profiles` directory. You must restart WebSphere® after the installer finishes and before you use this data source.
	For WebLogic, this is the directory that contains the `common` directory.
	Click **Next** to go to the next window.
Application server information (WebSphere®)	Applies only if your application server is WebSphere®. ◦ Enter the application server profile where you plan to deploy Campaign. ◦ Enter the name of the server in the profile. ◦ If security is enabled in WebSphere®, enter the administrator user ID and password.
	Click **Next** to go to the next window.

Window	Description
Application server information (WebLogic)	Applies only if your application server is WebLogic.
	Enter the domain server name and the administrator user ID and password. If SSL is enabled, enter the HTTP port for the domain.
	Note that you must add the database driver classpath to WebLogic in the WebLogic administration console.
	Click **Next** to go to the next window.
Preinstallation Summary	Review and confirm the values that you added during the installation process.
	Click **Install** to start the installation process.
	The Campaign installer opens.
Installation Complete	Click **Done** to close the Unica Campaign installer and go back to the Unica installer.

7. In the **Installation Complete** window, click **Done** to exit the Campaign installer and go back to the Unica installer.
8. Follow the instructions on the Unica installer to finish installing Campaign.

Use the information in the following table to take the appropriate actions on each window in the Unica installer.

Table 10. HCL Unica Installer GUI

Window	Description
Deployment EAR file	Specify whether you want to create an enterprise archive (EAR) file to deploy your Unica products.
	Click **Next**.
Package EAR file	You see this window if you selected **Create an EAR file for deployment** in the **Deployment EAR file** window.
	Select the applications that you want to package in the EAR file.

Window	Description
Package EAR file	You see this window if you selected **Create an EAR file for deployment** in the **Deployment EAR file** window. Select the applications that you want to package in the EAR file.

EAR file details	Enter the following information for your EAR file:
	
	    ◦ Enterprise application identifier
	    ◦ Display name
	    ◦ Description
	    ◦ EAR file path

Deployment EAR file	Specify whether you want to create another EAR file to deploy your Unica products. Click **Next**.
Installation Complete	This window provides the locations of the log files that are created during installation. Click **Previous** if you want to change any installation details. Click **Done** to close the Unica installer.

Creating an EAR file after running the installer

You can create an EAR file after you install Unica products. You might do this to create an EAR file with a desired combination of products.

About this task

📝 **Note:** Run the installer in console mode from the command line.

Use the following procedure if you want to create an EAR file after you install Unica products:

1. If this is the first time you are running the installer in console mode, make a backup copy of the installer's `.properties` file for each of your installed products.

   ◦ Each product installer creates one or more response files with a `.properties` extension. These files are in the same directory where you placed the installers. Be sure to back up all files with the `.properties` extension, including the `installer_<product initials><product version number>.properties` files and the file for the installer itself, which is named `installer.properties`.

      For example, the properties files for Unica Platform `installer_ump12.1.0.0.properties` and the properties file for Campaign `installer_uo12.1.0.0.properties`.

   ◦ If you plan to run the installer in unattended mode, you must back up the original `.properties` files, because when the installer runs in unattended mode, it clears these files. To create an EAR file, you need the information that the installer writes in the `.properties` files during the initial installation.

2. Open a command window and change directories to the directory that contains the installer.
3. Run the installer executable with this option:

   ```
   -DUNICA_GOTO_CREATEEARFILE=TRUE
   ```

   On UNIX™ type systems, run the `.bin` file rather than the `.sh` file.

   The installer wizard runs.

4. Follow the instructions in the wizard.
5. Before you create more EAR files, overwrite the `.properties` file or files with the backups you created before you ran in console mode for the first time. **Please note EAR is not supported for Tomcat application server.**

Installing Unica Campaign and Unica Optimize using the console mode

The console mode allows you to install Unica Campaign and Unica Optimize using the command-line window. You can select various options in the command-line window to complete tasks such as selecting the products to install or selecting the home directory for the installation.

Before you begin

Before you install Unica Campaign and Unica Optimize, ensure that you have configured the following.

- An application server profile
- A database schema

To display the installer screens correctly in console mode, configure your terminal software to support UTF-8 character encoding. Other character encodings, such as ANSI, do not render the text correctly, and some information is not readable with these encodings.

1. Open a command-line prompt window and navigate to the directory where you have saved the Unica installer and the Unica Campaign installer.
2. Complete one of the following actions to run the Unica installer.

Choose from:

◦ For Windows™, enter the following command:

`HCL_Unica_installer_12.0.0.0` -i console

For example, HCL_Unica_Installer_12.0.0.0_win.exe -i console

◦ For UNIX™, invoke the `HCL_Unica_installer_12.0.0.sh` file.

For example: HCL_Unica_installer_12.0.0.0.sh

3. Follow the directions that are displayed in the command-line prompt. Use the following guidelines when you must select an option in the command-line prompt:

   ◦ The default options are defined by the symbol [X].
   ◦ To select or clear an option, type the number that is defined for the option, and then press Enter.

Example

For example, suppose that the following list displays the components that you can install:

   1 [X] Unica Platform
   2 [X] Unica Campaign
   3 [ ] Unica Interact
   4 [ ] Unica Plan

If you want to install Unica Plan, and do not want to install Unica Campaign, enter the following command: 2,4

The following list would then display the options that you selected:

   1 [X] Unica Platform
   2 [ ] Unica Campaign
   3 [ ] Unica Interact
   4 [X] Unica Plan

> **Note:** Do not clear the option for Unica Platform unless you have already installed it.

4. The Unica installer launches the Unica Campaign installer during the installation process. Follow the instructions in the command-line prompt window of the Unica Campaign installer.

5. After you enter `quit` in the Unica Campaign installer command-line prompt window, the window shuts down. Follow the instructions in the command-line prompt window of the Unica installer to complete the installation of Campaign.

> **Note:** If any error occurs during the installation, a log file is generated. You must exit the installer to view the log file.

Installing Unica Campaign and Unica Optimize silently

Use the unattended or silent mode to install Unica Campaign and Unica Optimize multiple times.

Before you begin

Before you install Unica Campaign and Unica Optimize, make sure that you have configured the following elements:

- An application server profile
- A database schema

About this task

When you install Unica Campaign and Unica Optimize by using the silent mode, response files are used to obtain the information that is required during installation. You must create response files for a silent installation of your product. You can create response files by using one of the following methods:

- Using the sample response files as a template for creating your response files. The sample response files are included with your product installers in the `ResponseFiles` compressed archive. For more information about sample response files, see Sample response files on page 86.
- Running the product installers in the GUI (Windows™) or X Window System (UNIX™) mode or the console mode before you install the product in the silent mode. One response file is created for the Unica suite installer, and one or more response files are created for your product installer. The files are created in the directory that you specify.

  ❗ Important: For security reasons, the installer does not save database passwords in the response files. When you create response files, you must edit each response file to enter database passwords. Open each response file and search for PASSWORD to find where you must edit the response file.

When the installer runs in the silent mode, it looks for the response files in the following directories sequentially:

- In the directory where the Unica installer is saved
- In the home directory of the user who installs the product

Make sure that all response files are in the same directory. You can change the path where response files are read by adding arguments to the command line. For example: -DUNICA_REPLAY_READ_DIR="myDirPath" -f myDirPath/installer.properties

Use the following command for Windows™:

- `HCL_Unica_installer_12.1.0.0`   -i silent

  Example

  For example:

  HCL_Unica_installer_12.1.0.0_win.exe -i silent

Use the following command for Linux™:

- `HCL_Unica_installer_12.1.0.0 _operating_system`   .bin -i silent

  Example

For example:

HCL_Unica_installer_12.1.0.0_linux.bin -i silent

Sample response files

You must create response files to set up a silent installation of Unica Campaign . You can use sample response files to create your response files. The sample response files are included with the installers in the `ResponseFiles` compressed archive.

The following table provides information about sample response files:

Table 11. Description of sample response files

Sample response file	Description
`installer.properties`	The sample response file for the Unica suite installer.
`installer_product initials and product version number.properties`	The sample response file for the Unica Campaign installer.
	For example, `installer_ucn.n.n.n.properties` is the response file of the Unica Campaign installer, where *n.n.n.n* is the version number.
	For example, `installer_umpn.n.n.n.properties` is the response file of the Unica Platform installer, where `n.n.n.n` is the version number.
`installer_report pack initials, product initials, and version number.properties`	The sample response file for the reports pack installer.
	For example, `installer_urpcn.n.n.n.properties` is the response file of the Unica Campaign reports pack installer, where *n.n.n.n* is the version number.

Chapter 5. Configuring Unica Campaign and Unica Optimize before deployment

Before you deploy, create and populate the Unica Campaign , Unica Optimize, and Unica Deliver system tables and register Unica Campaign, Unica Optimize, and Unica Deliver manually.

> **Note:** Information about using web application clustering in Unica Campaign is provided in Unica Campaign web application clustering on page 163.

Creating and populating Unica Campaign and Unica Optimize system tables manually

By default, the Unica Campaign installer automatically creates and populates system tables. However, if the system tables are not created and populated automatically during the installation, you must populate the system tables manually. Use your database client to run the Unica Campaign and Unica Optimize SQL scripts against the appropriate database to create and populate the Unica Campaign and Unica Optimize system tables.

> **Note:** If you plan to enable Unica Deliver, you must also manually create and populate the Unica Deliver system tables if the installer did not do so automatically. For details, see Creating and populating the Unica Deliver system tables manually on page 89.

If you selected the **Campaign System Table DDL Files** option on the **Campaign Components** page during installation, the Campaign installer installs a collection of SQL scripts that you can use to create and populate the Unica Campaign and Unica Optimize system tables. The SQL scripts are installed in the `ddl` directory under your installation of the Unica Campaign server. If your system tables are configured for Unicode, the appropriate scripts are in the `ddl/unicode` directory under your Unica Campaign installation.

To use the SQL scripts, run your database client to apply the scripts to the database or schema that holds the Unica Campaign and Unica Optimize system tables. See the documentation for your database software for instructions on running SQL scripts.

The following tables list the SQL scripts that are provided for you to use to create and populate the Unica Campaign and Unica Optimize system tables manually.

Table 12. Scripts for creating Unica Campaign system tables

Data source type	Script name
IBM® DB2®	ac_systab_db2.sql
Microsoft™ SQL Server	ac_systab_sqlsvr.sql
Oracle	ac_systab_ora.sql
MariaDB	ac_systab_mariadb.sql

Table 13. Scripts for populating Unica Campaign system tables

Data source type	Script name
IBM® DB2®	ac_populate_tables_db2.sql
Microsoft™ SQL Server	ac_populate_tables_sqlsvr.sql
Oracle	ac_populate_tables_ora.sql
MariaDB	ac_populate_tables_mariadb.sql

Table 14. Scripts for creating Unica Optimize system tables

Data source type	Script name
IBM DB2®	aco_systab_db2.sql
Microsoft™ SQL Server	aco_systab_sqlsvr.sql

Table 14. Scripts for creating Unica Optimize system tables

(continued)

Data source type	Script name
Oracle	aco_systab_ora.sql
MariaDB	aco_systab_mariadb.sql

Table 15. Scripts for populating Unica Optimize system tables

Data source type	Script name
IBM® DB2®	aco_populate_tables.sql
Microsoft™ SQL Server	aco_populate_tables.sql
Oracle	aco_populate_tables.sql
MariaDB	aco_populate_tables.sql

Table 16. Scripts for adding extra indexes for Unica Optimize system tables

Data source type	Script name
IBM® DB2®	aco_extra_indexes_db2.sql
Microsoft™ SQL Server	aco_extra_indexes_sqlsvr.sql
Oracle	aco_extra_indexes_ora.sql
MariaDB	aco_extra_indexes_maridb.sql

You can create tables in the Unica Campaign schema if you want to integrate Unica Collaborate with Unica Campaign.

Creating and populating the Unica Deliver system tables manually

For Unica Deliver, you must create additional system tables in the Unica Campaign schema and then populate the tables with initial data. The Unica Campaign installer automatically creates and populates the Unica Deliver system tables in the Unica Campaign schema if you select the option to create system tables automatically. However, if you do not select the option, you must create and populate the Unica Deliver system tables manually.

Use your database client to run the appropriate script against the Unica Campaign database.

📝 **Note:** The Unica Deliver installation directory is a subfolder within the Unica Campaign directory.

Scripts for creating Unica Deliver tables

Campaign provides the `ace_op_systab` scripts for creating Unica Deliver tables in your local environment.

If your Unica Campaign system tables are configured for Unicode, find the appropriate script in the `ddl/unicode` directory under your Unica Deliver installation.

If your Unica Campaign system tables are not configured for Unicode, use the non-Unicode scripts in the `ddl` directory under your Unica Deliver installation. Use the following table to see the scripts that you must use:

Table 17. Scripts for creating Unica Deliver tables

Data source type	Script name
IBM® DB2®	ace_op_systab_db2.sql
	The user tablespace and system temporary tablespace where the system tables reside must each have a page size of 16K or greater.
Microsoft™ SQL Server	ace_op_systab_sqlsvr.sql
MariaDB	ace_op_systab_mariadb.sql
Oracle	ace_op_systab_ora.sql

Scripts for populating Unica Deliver tables

HCL provides the `ace_op_populate_systab` scripts for populating the Unica Deliver tables in your local environment.

The populate scripts are stored in the `ddl` directory of your Unica Deliver installation. HCL provides only one version of the populate scripts because you can use them for either Unicode or non-Unicode tables.

📝 **Note:** The Unica Deliver installation directory is a subfolder within the Unica Campaign directory.

Use the following table to see the scripts that you must use:

Table 18. Scripts for populating Unica Deliver tables

Data source type	Script name
IBM® DB2®	ace_op_populate_systab_db2.sql
MariaDB	ace_op_populate_systab_mariadb.sql
Microsoft™ SQL Server	ace_op_populate_systab_sqlsvr.sql
Oracle	ace_op_populate_systab_ora.sql

Registering Unica Campaign manually

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

About this task

You can use the configTool and populateDb utilities to import and populate the Unica Campaign information in the Unica Platform system tables.

1. Run the populateDb utility by using the following sample command as a guideline.

   populateDb.bat -n Campaign

   The command imports security roles and permissions into the default partition.

2. **If you are upgrading Unica Campaign**, unregister Unica Campaign by using the following command:

   configTool -x -p "Affinium|Campaign|about" -f exportedAbout.xml

   The command exports the Unica Campaign **About** node to the `exportedAbout.xml` file.

3. Run the `configTool` utility by using the following sample commands as guidelines:
   - configTool -r Campaign -f "*full_path_to_Campaign_installation_directory*\conf\campaign_configuration.xml"
   - configTool -v -i -p "Affinium|suite|uiNavigation|settingsMenu" -f "*full_path_to_Campaign_installation_directory*\conf\campaign_setup_navigation.xml"
   - configTool -v -i -p "Affinium|suite|uiNavigation|mainMenu" -f "*full_path_to_Campaign_installation_directory*\conf\campaign_navigation.xml"
   - configTool -v -i -p "Affinium|suite|uiNavigation|mainMenu|Analytics" -f "*full_path_to_Campaign_installation_directory*\conf\campaign_analysis_navigation.xml"
   - configTool -v -i -p "Affinium|suite|uiNavigation|alerts" -f "*full_path_to_Campaign_installation_directory*\conf\campaign_alerts.xml"
   - ```
     configTool -v -i -p "Affinium|suite|uiNavigation|mainMenu|Campaign" -f
     "full_path_to_Campaign_installation_directory\conf\optimize_navigation.xml"
     ```

- ° `configTool -v -i -p "Affinium|suite|security|apiSecurity|campaign" -f`
  `"full_path_to_Campaign_installation_directory\conf\API_Filter.xml"`
- ° `configTool -v -i -p "Affinium|suite|security|apiSecurity|campaign" -f`
  `"full_path_to_Campaign_installation_directory\conf\API_V2_Filter.xml"`
- ° `configTool -v -i -p "Affinium|suite|security|apiSecurity|campaign" -f`
  `"full_path_to_Campaign_installation_directory\conf\API_V3_Filter.xml"`
- ° `configTool -v -i -p "Affinium|suite|security|apiSecurity|campaign" -f`
  `"full_path_to_Campaign_installation_directory\conf\Engage_API_Filter.xml"`

The commands import configuration properties and menu items. You must run the utility as many times as there are files.

4. Run the following command after you register Unica Campaign:

configtool -i -p "Affinium|Campaign" -f exportedAbout.xml

The command imports the Unica Campaign **About** node to the `exportedAbout.xml` file.

Registering Unica Deliver manually

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

About this task

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

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

To register Unica 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 Unica Deliver installation directory is a subdirectory of the Unica Campaign installation directory.

Setting data source variables in the Unica Campaign startup script (UNIX only)

Data source variables are automatically set by the installer during installation of Unica Campaign. You can change these settings in the `Campaign/bin/setenv.sh` file. You must restart the server every time you modify the `setenv.sh` file.

About this task

During Unica Campaign installation, the Installer collects database information from you and uses that information to automatically configure the database and environment variables that are required to create and use the Unica Campaign system tables. Those settings are stored in the `setenv.sh` file in the `bin` directory under your Unica Campaign server installation.

For access to data sources (such as Unica Campaign customer tables) that do not use the same type of database as the system tables, you must manually configure the `setenv.sh` file to add the database and library environment variables that are described in Database and library environment variables on page 93.

Note that if you modify this file when the Unica Campaign server is already running, you must restart the server before changes to the setenv file are recognized. See Starting the Unica Campaign server on page 109 for more details.

See Distributed Marketing database information worksheet for the required information to add to the setenv file.

Database and library environment variables

Set database and library environment variables that are required for your databases (customer tables and system tables, if you chose Manual Database Setup during installation) and operating system. You can set the database and library variables in the `setenv.sh` file.

Following table describes database names with their syntax and descriptions.

Table 19. Database environment variables

Database	Syntax and Description
DB2®	DB2DIR=*full_dir_path* export DB2DIR DB2® installation directory (for example, `/usr/lpp/db2_06_01`) `. full_path_to_db2profile` Sources the database configuration for DB2® users (e.g.`/home/db2inst1/sqllib/db2profile`). Note the ". " (period and then a space).
Netezza®	NZ_ODBC_INI_PATH=*full_dir_path* export NZ_ODBC_INI_PATH Directory location of the `odbci.ini` file (e.g. `/opt/odbc64v51`)

Table 19. Database environment variables

(continued)

Database	Syntax and Description
	`ODBCINI=`*`full_path_and_file_name`*
	`export ODBCINI`
	Full path to the `odbc.ini` file
Oracle	`ORACLE_BASE=`*`full_dir_path`*
	`export ORACLE_BASE`
	Oracle installation directory
	`ORACLE_HOME=`*`full_dir_path`*
	`export ORACLE_HOME`
	Oracle's home directory (e.g. `/home/oracle/OraHome1`).
Teradata	`ODBCINI=`*`full_path_and_file_name`*
	`export ODBCINI`
	Full path to the `obdc.ini` file
MariaDB	
	`Maria DB ODBCINI=full_path_and_file_name`
	`export ODBCINI`
	Full path to the `obdc.ini` file
ODBCINI	Full path to the obdc.ini file
	`export ODBCINI`
ODBCINST	Full path to the obdcinst.ini file
	`export ODBCINST`

Define the library environment variable as mentioned in the following table, depending on your UNIX™ operating system.

Table 20. Library environment variable

Operating System	Value

SUSE and RHEL Linux™	`LD_LIBRARY_PATH`
	
	For example:
	
	`LD_LIBRARY_PATH=<Campaign_Home>/bin:<path to DB lib directory>:$LD_LIBRARY_PATH`
	
	`export LD_LIBRARY_PATH`
	
	✏️ **Note:** If `LD_LIBRARY_PATH_64` (for 64-bit linking) is set, remove it. The `LD_LIBRARY_PATH` variable will be ignored if `LD_LIBRARY_PATH_64` is set.

AIX®	`LIBPATH` For example: `LIBPATH=<Campaign_Home>/bin:<path to DB lib directory>:/usr/lib:$ORACLE_HOME/lib32:$ORACLE_HOME/lib`
AIXÂ®	LIBPATH for OneDB `LIBPATH=<Campaign_Home>/bin:/opt/IBM/Informix_Client-SDK/lib/esql:/opt/IBM/Informix_Client-SDK/lib:/opt/IBM/Informix_Client-SDK/lib/client:$LD_LIBRARY_PATH`
HP-UX	`SHLIB_PATH` For example: `SHLIB_PATH=<Campaign_Home>/bin:<path to DB lib directory>:/usr/lib:$ORACLE_HOME/lib32:$ORACLE_HOME/lib`

Library directories for Oracle databases

Different versions of Oracle have different naming conventions for their `lib` directories. Older versions used `lib` for 32-bit and `lib64` for 64-bit. Newer versions use `lib32` for 32-bit and `lib` for 64-bit.

If you install 32-bit Unica Campaign, you must include either `$ORACLE_HOME/lib32` or `$ORACLE_HOME/lib`, whichever one contains the 32-bit Oracle libraries.

If you install 64-bit Unica Campaign, you must include either `$ORACLE_HOME/lib` or `$ORACLE_HOME/lib64`, whichever one contains the 64-bit Oracle libraries.

Note: Do not include paths to both the 32- and 64-bit libraries; include only the path to the library you are using for your version of Unica Campaign.

Chapter 6. Deploying the Unica Campaign web application

You can deploy the Unica Campaign web application by using an EAR file or you can deploy the individual WAR files.

To deploy Unica Campaign, follow the guidelines in this section and then start the Unica Campaign server.

When you ran the Unica Campaign installer, you may have included Unica Campaign in an EAR file, or you may choose to deploy the Campaign WAR file. If you included the Unica Platform or other products in an EAR file, you must follow all the deployment guidelines detailed in the individual install guides for the products included in the EAR file.

You must know how to work with your web application server. Consult your web application server documentation for details such as navigation in the Administration console.

> **Note:** Information about using web application clustering in Unica Campaign is provided in .

Configuring JVM parameters for Unica products

You must configure the following JVM parameters in the Unica porducts.

Table 21. Unica Products and the JVM Parameters to be Configured

Unica Product Name	JVM Parameters
Unica Campaign	`-DUNICA_PLATFORM_CACHE_ENABLED=true` `-DLICENSE_ENABLED=false` `-Dcampaign.log4j.async=true` `-Dcom.sun.management.jmxremote` `-Dcom.sun.management.jmxremote` `-Dcom.sun.management.jmxremote.port=1007` `-Dcom.sun.management.jmxremote.ssl=false` `-Dcom.sun.management.jmxremote.authenticate=false` `-Dcom.sun.management.jmxremote.password.file=../conf/jmxremote.password` `-Dcom.sun.management.jmxremote.access.file=../conf/jmxremote.access` `-DUNICA_GOTO_CREATEEARFILE=TRUE` `-Dfile.encoding=UTF-8` `-Dclient.encoding.override=UTF-8`

Table 21. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters

	`-DUNICA_GOTO_CREATEEARFILE=TRUE` `-Dclient.encoding.override=UTF-8` `-Djboss.as.management.blocking.timeout=3600`
	SET SSL_OPTIONS=-Djavax.net.ssl.keyStoreType="JKS"
	    `-Djavax.net.ssl.trustStore="C:\security\myTrustStore.jks"` `-Djavax.net.ssl.trustStorePassword=myPassword`
	Configuring the `ACOOptAdmin` tool (`ACOOptAdmin.sh` (UNIX) or `ACOOptAdmin.bat` (Windows) file)
	    `-Djavax.net.ssl.keyStoreType=keyStoreTypeValue` `-Djavax.net.ssl.keyStore=<keyStoreValue>` `-Djavax.net.ssl.keyStorePassword=<keyStorePasswordValue>` `-Djavax.net.ssl.trustStore=<trustStoreValue>` `-Djavax.net.ssl.trustStorePassword=<trustStorePasswordValue>` `-Djava.awt.headless=true` `-Dcampaign.deliveretl.disabled=true` `-Dcampaign.interactetl.disabled=true` `-Dcampaign.journeyetl.disabled=false` `-Dcampaign.journeyetl.disabled=true`
	`EMAIL_TLS_ENABLED`
	
	`EMAIL_SSL_TRUST_ENABLED`
	
	`SCH_ASYNCH_EXECUTION_ENABLED`

Unica Centralized Offer Management	`-DUNICA_PLATFORM_CACHE_ENABLED=true` `-DUNICA_PLATFORM_LOCAL_CACHE_ENABLED=true`

Table 21. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	• `-Dclient.encoding.override=UTF-8` • `-DOFFER_HOME=D:\sourcecode\latest\CentrOM\Offers\om-server`
Unica Contact Central	• `CONTACTCENTRAL_HOME` • `UNICA_PLATFORM_HOME`
Unica Content Integration	• `-DASSET_PICKER_HOME` • `-DUNICA_PLATFORM_HOME`
Unica Deliver	• `-Dcampaign.deliveretl.disabled=true`
Unica Interact	**Interact design time** The following are the JVM parameters for Interact design time: • `Interact.UsernameToAlwaysDeployFor` • `com.unicacorp.interact.deliver.templateTimeout` • `DeploymentServletParameterEncoding` • `com.unicacorp.interact.flexoffers.defaultDateTimeFormat` • `com.unicacorp.interact.flexoffers.defaultDateFormat` • `com.unicacorp.Campaign.interact.offermapping.batchsize` • `com.unicacorp.Campaign.interact.offermapping.service.synctimeout` • `com.unicacorp.interact.cacheTTL` • `com.unicacorp.interact.cacheRefreshIntervalInMin` • `com.unicacorp.interact.enableDTPerfLogging` • `com.unicacorp.interact.compressAPIResponse` • `ignoreSpecialCharacterValidator` • `Interact.CustomStringDelimiter` **Interact run time** The following are the JVM parameters for Interact run time:

Table 21. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	`INTERACT_HOME` `com.unicacorp.interact.deliver.templateTimeout` `INBOUND_GATEWAYS_USING_MAPPING_FROM_PROPERTIES` `com.hcl.interact.http.proxyProtocol` `com.hcl.interact.http.proxyHost` `com.hcl.interact.http.proxyPort` `com.hcl.interact.http.proxyUsername` `com.hcl.interact.http.proxyPassword` `interact.jmx.monitoring.port` `interact.runtime.instance.name` `interact.offerserving.maxOfferAllocationInMemoryPerInstance` `interact.offerserving.maxDistributionPerIntervalPerInstanceFactor` `interact.ignitePort` `com.unicacorp.interact.chDupeCheckLimit` `com.unicacorp.interact.rhDupeCheckLimit` `com.unicacorp.interact.chSuppressDupe` `com.unicacorp.interact.rhSuppressDupe` `com.unicacorp.interact.testclient.nullValue` `interact.ehcache.config` `interact.api.dateFormat` `com.hcl.interact.testrun.rowlimit` `Interact.DisableExceptionStackTracesInMacros` `com.unicacorp.interact.enableDetailStats` `com.unica.interact.deployment.timeoutInSecs` `com.ibm.interact.instance.name` `com.unicacorp.interact.invalidPaths` `interact.XSessResponseConsumerManager.generateOnlyOneResponse` `tryToPreserveInexactFloatValues` `com.unicacorp.interact.propertyRefreshInterval` `com.unicacorp.interact.scheduledTasksProcessInterval` `com.unicacorp.interact.eventpatterns.parallelism` `com.unicacorp.interact.eventpatterns.restartRetries` `com.unicacorp.interact.eventpatterns.evaluateTimeoutMilli` `com.unicacorp.interact.eventpatterns.restartRetryDelayInSec`

Table 21. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	• `Interact.advisoryMessageEncodingOverrides` • `com.unica.interact.api.insertSessionIDAsCookie` • `com.unica.interact.api.SessionIDCookieName` • `InteractMsgCode` • `com.ibm.interact.triggeredmessage.enableJMSConsumer` • `com.unicacorp.interact.maxStringLengthInFormatMacro` • `ContinueEvaluatingBranchAndAdvOptTreatmentLogicDespiteE` `xMessageList` • `DisableDecisionProcessBoxAndAdvOptTreatmentLogging` • `TwoDigitYearStartDate` • `Interact.enableTwoDigitYearFix` • `com.ibm.interact.evpatetl.conf` • `com.unicacorp.interact.minTreatmentsPerThread` • `com.unicacorp.interact.maxTreatmentPoolSize` • `CircuitBreaker.processTimeoutMillis` • `com.unicacorp.interact.event.asyncTimeoutMSec` • `com.unicacorp.interact.eventActionTimeout` • `Interact.HTMI.Enabled` • `Interact.HTMI.MaxRequestDurationInMs` • `Interact.HTMI.RecordIndividualAPIs` • `Interact.HTMI.MaxStartSessionDurationInMs` • `Interact.HTMI.MaxGetOffersDurationInMs` • `Interact.HTMI.MaxPostEventDurationInMs` • `Interact.HTMI.MaxGetProfileDurationInMs` • `Interact.HTMI.LogErrorsEveryNthTime` • `Interact.HTMI.UseMillisecondTimers` • `Interact.HTMI.Debug` • `com.unicacorp.interact.suppressWarningOnAnonymousUser` • `com.hcl.interact.eventpatterns.printPatternAction` • `com.hcl.interact.eventpatterns.eagerPersist` • `com.ibm.interact.triggeredmessage.addPerfData` • `com.unicacorp.interact.learning.disableAggregator` • `com.unicacorp.interact.learning.disableDeletion` • `com.unicacorp.interact.learning.ignoreInterval` • `interact.services.loader.saveLoaderFiles` • `ConvertEveryNULLAttributeValueToAJEPNullConstant` • `includeJoinInfo` • `com.unicacorp.interact.deployment.reloadTimeout` • `com.ibm.interact.lockTimeWarningThreshold`

Table 21. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	• `com.unicacorp.interact.cache.maxWaitTime` • `DEFAULT_PERSISTENCE_PROVIDER`
Unica Journey	
Unica Plan	• `-Dplan.log4j.config=<plan_home>/conf/plan_log4j.xml` • `-Dplan.log4j.async=TRUE` • `-Dplan.home=<plan_home>`
Unica Platform	• `-REMOVE_REQUESTED_AUTHN_CONTEXT=TRUE` • `-DENABLE_PLATFORM_LOG4J_XML_LOGGING=TRUE` • `-DENABLE_PERSISTENT_NAMEID_FORMAT=TRUE` • `-DUNICA_PLATFORM_HOME=<platform_home_directory_path>`

Setting the web application session timeout

Session timeout determines how long an inactive HTTP session can remain open before the session expires. If required, you can set the web application session timeout for Unica Campaign by using the WebSphere® console or the WebLogic console to adjust the value of session timeout in seconds or minutes.

Set the session timeout on your web application server:

- WebSphere®: Use the WebSphere® Application Server Administrative Console to set the Session timeout in minutes. You can adjust this setting at the server and Enterprise Application levels. See your WebSphere® documentation for details.
- WebLogic: Use the WebLogic console to set the Session Timeout in seconds or adjust the **TimeoutSecs** parameter value for the **session-descriptor** element in the weblogic.xml file.

Deploying Unica Campaign on WebSphere Application Server

You can deploy the product runtime environment on supported versions of WebSphere® Application Server (WAS) from a WAR file or EAR file. The design time environment is deployed automatically with the product EAR or WAR file.

About this task

- Ensure that multiple language encoding is enabled in WAS.
- When you run the Install New Application wizard, ensure that you set the JDK Source Level to 18.

- Ensure that you add `javax.el-3.0.1-b11.jar` in WAS server lib directory. You can download `javax.el-3.0.1-b11.jar` from https://mvnrepository.com/artifact/org.glassfish/javax.el/3.0.1-b11
- Add JVM argument `-Dhttps.protocols=TLSv1.2` in Websphere for Unica Deliver to work.

Deploying Campaign on WAS from a WAR file

You can deploy the product application from a WAR file on WAS.

Before you begin

Complete the following tasks before you deploy the product:

• Confirm that your version of WebSphere meets the requirements in the Recommended Software Environments and Minimum System Requirements document, including any necessary fix packs or upgrades.

• Confirm that you created the data sources and database provider in WebSphere.

1. Go to the WebSphere® Integrated Solutions Console.
2. Go to **Applications > Application Types > WebSphere enterprise applications** and click **Install**.
3. In the **Preparing for the application installation** window, select the **Detailed - Show all options and parameters** check box and click **Next**.
4. Click **Continue** to see the **Install New Application** wizard.
5. Accept the default settings on the windows of the **Install New Application** wizard except the following windows:
   - In step 1 of the **Install New Application** wizard, select the **Precompile JavaServer Pages files** check box.
   - In step 3 of the installation wizard, set the **JDK Source Level** to 18.
   - In step 9 of the installation wizard, set the **Context Root** to /Campaign.
6. In the left navigation panel of WebSphere® Integrated Solutions Console, navigate to **Applications > Application Types > WebSphere enterprise applications**.
7. In the **Enterprise Applications** window, click the `Campaign.war` file.
8. In the **Web Module Properties** section, click **Session Management** and select the following check boxes.
   Choose from:
   - Override session management
   - Enable Cookies

   Click **Enable Cookies**, and in the **Cookie name** field, enter a unique cookie name.

9. In the **Applications > Enterprise Applications** section of the server, select the WAR file that you deployed.
10. In the **Detail Properties** section, select **Class loading and update detection**.
11. In the **Class loader order** section, select the **Classes loaded with local class loader first (parent last)** option. In the **WAR class loader policy** section, select the **Single class loader for application** option.
12. Enter the context root for the application as `/Campaign`.
13. Go to **Application Servers > your server > Process definition > Java Virtual Machine**.
14. Users can add the following JVM arguments to enable unicode support with this JVM.

      ◦ `-Dfile.encoding=UTF-8`

      ◦ `-Dclient.encoding.override=UTF-8`

15. Start your deployment.

Deploying Campaign on WAS from an EAR file

You can deploy the product by using an EAR file if you included Campaign.

Before you begin

- Confirm that your version of WebSphere® meets the requirements in the *Recommended Software Environments and Minimum System Requirements* document, including any necessary fix packs or upgrades.
- Confirm that you created the data sources and database provider in WebSphere®.

1. Go to the WebSphere® Integrated Solutions Console.
2. Go to **Applications > Application Types > WebSphere enterprise applications** and click **Install**.
3. In the **Preparing for the application installation** window, select the **Detailed - Show all options and parameters** check box and click **Next**.
4. Click **Continue** to see the **Install New Application** wizard.
5. Accept the default settings on the windows of the Install New Application wizard except the following windows:
   - In step 1 of the **Install New Application** wizard, select the **Precompile JavaServer Pages files** check box.
   - In step 3 of the installation wizard, set the **JDK Source Level** to 18.
   - In step 9 of the installation wizard, set the **Context Root** to /Campaign.
6. In the left navigation panel of WebSphere® Integrated Solutions Console, navigate to **Applications > Application Types > WebSphere enterprise applications**.
7. In the Enterprise Applications window, select the EAR file that you want to deploy.
8. In the **Web Module Properties** section, click **Session Management** and select the following check boxes:
   - **Override session management**
   - **Enable Cookies**
9. Click **Enable Cookies**, and in the **Cookie name** field, enter a unique cookie name.
10. Select **Servers > WebSphere application server > server 1 > Session management > Enable Cookies** and clear the check box for **Set session cookies to HTTPOnly to help prevent cross-site scripting attacks**.
11. In the **Detail Properties** section, select **Class loading and update detection**.
12. In the **Class loader order** section, select the **Classes loaded with parent class loader first** option.
13. In `campaign.ear`, open the **Manage Modules** section and select the **Classes loaded with local class loader first (parent last)** option.
14. For **WAR class loader policy**, select the **Class loader for each WAR file in application** option.
15. Go to **Application Servers > your server > Process definition > Java Virtual Machine**.
16. Users can add the following JVM arguments to enable unicode support with this JVM.
    - `-Dfile.encoding=UTF-8`
    - `-Dclient.encoding.override=UTF-8`
17. Start your deployment.

> 📝 **Note:** For Unica Campaign Websphere on HTTP and Unica Link (for native installation) on HTTPS, user need to set below parameters in Websphere JVM, SSL configuration and retrieve SSL certificates. To avoid problem "javax.net.ssl.SSLException: Received fatal alert: protocol_version"

JVM Arguments: -Dhttps.protocols=TLSv1.2 -Djdk.tls.client.protocols=TLSv1.2

**

Set the protocol to TLSv1.2 from WAS admin console in below mentioned path and restart your Websphere Application Server.

SSL Certificate and Key management -> SSL configurations ->NodeDefaultSSLSetings ->Quality of protection(Qop) settings

protocol: TLSv1.2

Retrieve the signer certificates from hip-rest server and add into NodeDefaultKeyStore and NodeDefaultTrustStore

*path: SSL certificate and key management > Key stores and certificates > NodeDefaultKeyStore > Signer certificates > Retrieve from port*

Deploying Unica Campaign on WebLogic

You can deploy Unica products on WebLogic. If a console is not accessible or does not exist, you can emulate a console using VNC or Exceed kind of applications. Contact your technical administrator to enable GUI mode display on Linux environments.

About this task

Use the following guidelines when you deploy Unica Campaign on WebLogic:

- Unica products customize the JVM used by WebLogic. You might need to create a WebLogic instance that is dedicated to Unica products if you encounter JVM-related errors.
- Verify that the SDK selected for the WebLogic domain you are using is the Sun SDK by looking in the startup script (`startWebLogic.cmd`) for the JAVA_VENDOR variable. It should be set to: `JAVA_VENDOR=Sun`. If it is set to `JAVA_VENDOR=BEA`, JRockit has been selected. JRockit is not supported. To change the selected SDK, refer to the WebLogic documentation.
- Deploy the Unica products as web application modules.

- On UNIX™ systems, you must start WebLogic from the console to allow correct rendering of graphical charts. The console is usually the machine on which the server is running. However, in some cases the web application server is set up differently.

  Connecting to the UNIX™ machine via telnet or SSH always causes problems rendering charts.

- If you are configuring WebLogic to use the IIS plug-in, review the WebLogic documentation.
- Add the following parameters in the `JAVA_OPTIONS` section of `startWeblogic.cmd` or `startWeblogic.sh`:

```
-Dfile.encoding=UTF-8
```

- Add the following parameters if you require to enable unicode support with Unica Campaign environment.

```
-Dfile.encoding=UTF-8
```

```
-Dclient.encoding.override=UTF-8
```

- If you are deploying in a production environment, set the JVM memory heap size parameters to at least 1024 by adding the following line to the `setDomainEnv` script:

Set `MEM_ARGS=-Xms1024m -Xmx1024m -XX:MaxPermSize=256m`

- Under certain circumstances, deploying older legacy interactive channels or interactive channels with large deployment histories can stress the system and require 2048mb or greater of Unica Campaign designtime and/or Interact runtime Java™ heap space.

  System administrators can adjust the amount of memory available to the deployment systems via the following JVM parameters:

  -Xms####m -Xmx####m -XX:MaxPermSize=256m

  Where the characters #### should be 2048 or higher (depending on their system load.) Note that a 64-bit application server and JVM are usually necessary for values greater than 2048.

These are the suggested minimum values. Analyze your sizing requirements to determine correct values for your needs.

Configuring WebLogic to display reports (UNIX)

If you install Unica Campaign with Unica Optimize on a UNIX™ system, you must enable the `java.awt.headless` JVM property so that the WebLogic web application server can display the graphs in Unica Optimize reports.

About this task

To enable graph display in Optimize reports in the Weblogic JVM:

1. Shut down the Weblogic server if it is already running.
2. Locate and open the startup script for your Weblogic server (`startWebLogic.sh`) in any text editor.
3. Modify the JAVA_OPTIONS parameter to add the following value:

```
-Djava.awt.headless=true
```

4. Save the startup script, then restart the Weblogic server.

Deploying Unica Campaign on JBoss Application Server

You can deploy Unica products on Jboss application server.

About this task

Use the following guidelines when you deploy Unica Campaign on Jboss:

- Verify that your version of JBoss meets the requirements that are described in the *Recommended Software Environments and Minimum System Requirements* document, including any necessary fix packs or upgrades.
- HCL Unica products customizes the JVM used by Jboss. You might need to create a Jboss instance that is dedicated to HCL Unica products if you encounter JVM-related errors
- If you are deploying in a production environment, set the JVM memory heap size parameters to at least 1024 by adding the following line to the `standalone.conf` in the `JBOSS_EAP_HOME/bin`

  e.g : `JAVA_OPTS="-Xms1024m -Xmx1024m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m "`

  These are the suggested minimum values. Analyze your sizing requirements to determine correct values for your needs. Depending upon system load **-Xmx** and **MaxMetaspaceSize** value should be adjusted. Note that a 64-bit application server and JVM are usually required for values greater than 2048.

- Modify the JAVA_OPTIONS parameter to add the following value in `setenv.bat/sh`.

  set "`JAVA_OPTS=%JAVA_OPTS%`

  `-DUNICA_PLATFORM_CACHE_ENABLED=true`

  `-Dclient.encoding.override=UTF-8`

  `-Djboss.as.management.blocking.timeout=3600"`

- Go to the JBoss Enterprise Application management Console e.g *http://<Jboss_Host_IP>:<Port>/management*
- Deploy Campaign.war/EAR on JBOSS.
- Alternately you can also deploy Campaign.war using jboss-cli.bat/sh.

  e.g Execute the command: jboss-cli.bat/sh and connect to Jboss instance using the command - connect <Jboss_Host_IP>:<JBOSS management Port>

  deploy <Campaign_Install_Path/Campaign/Campaign.war

- After all these changes restart the Jboss application server.

Deploying Unica Campaign on Tomcat Application Server

You can deploy Unica products on Tomcat application server.

About this task

Use the following guidelines when you deploy Unica Campaign on Tomcat:

- HCL Unica products customize the JVM used by Tomcat. You may require to create a Tomcat instance that is dedicated to HCL Unica products if you encounter JVM-related errors
- If you are deploying in a production environment, set the JVM memory heap size parameters to at least 1024 by adding the following line to the setenv.bat/sh e.g : set `CATALINA_OPTS=%CATALINA_OPTS% -Xms1024m -Xmx1024m -XX:MaxPermSize=512m`.

  These are the suggested minimum values. Analyze your sizing requirements to determine correct values for your needs. Depending upon system load **-Xmx** value should be adjusted. Note that a 64-bit application server and JVM are usually necessary for values greater than 2048.

- Modify the `JAVA_OPTIONS` parameter to add the following value in `setenv.bat/sh`.

  set `JAVA_OPTS=%JAVA_OPTS% -DUNICA_PLATFORM_CACHE_ENABLED=true -Dclient.encoding.override=UTF-8`.

- When Unica Insights war is deployed on Tomcat, complete the following configuration in server.xml, where Campaign.war is deployed.

  Add `relaxedQueryChars` under Campaign server.xml as mentioned below.

  ```
  <Connector port="8080" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443"
  relaxedQueryChars="|,[,],\,`,{,},^" />
  ```

- You must add Unica Platform and Unica Campaign data source in Campaign.xml along with the path of Campaign.war to the Unica Campaign Tomcat instance. For example:

  ```
  <Context docBase="<Campaign_Install_Path>/Campaign/Campaign.war">
  <Resource name="<UnicaPlatformDS>" factory="com.unica.manager.tomcat.utils.TomcatDSFactory"
   auth="Application" type="javax.sql.DataSource"
  maxActive="30" maxIdle="10" maxWait="10000"
  username="<db user for Platform schema>" password="<db password>" driverClassName=
  "<db specific class name>" url="<db specific jdbc url>"/>

  <Resource name="<campaignPartition1DS>" factory="com.unica.manager.tomcat.utils.TomcatDSFactory"
   auth="Application" type="javax.sql.DataSource"
  maxActive="30" maxIdle="10" maxWait="10000"
  username="<db user for Campaign schema>" password="<db password>" driverClassName=
  "<db specific class name>" url="<db specific jdbc url> </Context>"/>
  ```

  Note: {{You can encrypt DB password using }} encryptTomcatDBPasswords utility located in <Platform_Home>/tools/bin

- Restart the Tomcat application server.

  Note:

  Unica products are installed on an AWS ELB with Nginx individual products tomcat instances. AWS ELB is HTTPS and nginx is on http and all products are also on http. This causes issue on campaign settings and

campaign pages in Chrome browser. To avoid this problem user must configure Tomcat for SSL offloading and add below parameter:

scheme="https" secure="true" in the <Connector> element in the server.xml file.

URL for reference : https://backstage.forgerock.com/knowledge/kb/article/a37878699

Tomcat Parameters for Flowchart crash

Please include below parameters in the instance_name/conf/server.xml file.
  ◦ maxHttpHeaderSize="209715200"
  ◦ maxPostSize="-1"

Starting the Unica Campaign server

Make sure that Unica Platform and Unica Campaign web applications are deployed and running when you start the Unica Campaign server.

About this task

You can start the Unica Campaign server directly or install it as a service.

Starting the Unica Campaign listener manually

Start the Unica Campaign listener by running the `cmpServer.bat` file for Windows™, or the `rc.unica_ac` command for UNIX™.

About this task

Follow the instructions appropriate for your operating system.

Windows™

Start the Unica Campaign listener by running the `cmpServer.bat` file in the `bin` directory under your Unica Campaign installation. When the `unica_aclsnr.exe` process appears on the Windows™ Task Manager Processes tab, the server has started successfully.

UNIX™

Start the Unica Campaign listener by executing the `rc.unica_ac` program with a `start` argument. You must run this command as root. For example:

```
./rc.unica_ac start
```

To determine whether the `unica_aclsnr` process has started successfully, enter this command:

```
ps -ef | grep unica_aclsnr
```

To determine the process ID for the server you started, view the `unica_aclsnr.pid` file located in the `conf` directory under your Unica Campaign installation.

Note: Since Unica Campaign supports listener clustering, a loop is added in the `rc.unica_ac` which automatically restarts the listener if in case the listener goes down or crashes. Also, process `rc.unica_ac` still remains after starting listener and the parent process may change. For example, if you are upgrading from Release 8.6.0.4 to Release 9.1.1, the parent process of `unica_aclsnr` can be changed to `rc.unica_ac` instead of `init`, which was the former parent process and the process `rc.unica_ac` still remains after starting the listener.

Installing the Unica Campaign listener as a Windows service

You can install the Unica Campaign listener as a Windows™ service so that it starts automatically whenever you start Windows™.

1. Add the `bin` directory under the Unica Campaign installation directory to the user `PATH` environment variable. If the `PATH` environment variable does not exist for the user, create it.

   Make sure that you add this path to the user `PATH` variable, not the system `PATH` variable.

   If the Unica Campaign `bin` directory exists in the system `PATH` environment variable, remove it. You do not need it in the system `PATH` environment variable to install the Unica Campaign listener as a service.

2. If you are upgrading from a previous version of Unica Campaign that had the server installed as a service, stop the service.
3. Open a command window and change directories to the `bin` directory under your Unica Campaign installation.
4. Run the following command to install the Unica Campaign listener as a Windows™ service:

   `unica_aclsnr -a`

   Note: The `-a` option includes auto-restart functionality. If you do not want the service to attempt to restart automatically, use `unica_aclsnr -i`.

   Result

   The listener is now installed as a service.

   Note: Make sure `CAMPAIGN_HOME` has been created as a system environment variable before starting the Unica Campaign listener service.

   Note: You must add the appropriate listener type.

5. Open the **Unica Campaign Listener Service** properties dialog box. Click the **Log On** tab.
6. Select **This account**.
7. Enter the user name (system user) and password and start the services.

Chapter 7. Configuring Unica Campaign and Unica Optimize after deployment

After you deploy Unica Campaign, you must verify that the Unica Campaign listener is running, set up the Unica Campaign system user, set Unica Campaign configuration properties, and verify the Unica Campaign installation.

If you are using the Unica reporting feature, you must complete the tasks that are described in the *Unica Reports Installation and Configuration Guide*.

Note: If you plan to enable Unica Deliver for hosted email, you must use the Unica reporting feature to view standard Unica Deliver performance reports.

Setting the Unica Optimize configuration properties manually

Set configuration properties for the Unica Optimize listener. You can set the host server name and port for the Unica Optimize. You can configure Unica Optimize to connect to the Unica Platform server by using SSL as well.

About this task

If the Unica Campaign installer cannot access the Unica Platform system tables during the installation process, the installer displays an error message. The installation process proceeds, but go to **Settings > Configuration** to set the Unica Optimize configuration properties manually.

If required, set the following Unica Optimize configuration properties under the **Campaign > unicaACOListener** category manually:

- **serverHost**
- **serverPort**
- **useSSL**

Note: If you have a single listener (Non-Cluster)then

- If the listener is FOR Campaign flowcharts only, then configure the configuration properties under

  Campaign > unicaACListener >

- If the listener is FOR Optimize sessions only, then configure the configuration properties under

  Campaign > unicaACOListener >

- If the listener is FOR BOTH Campaign flowcharts and Optimize sessions, then configure the both configuration properties under

  Campaign > unicaACListener >

  Campaign > unicaACOListener >

> ✏️ Following properties under unicaACListener and unicaACOListener, should have same values , since there is only one listener serving Campaign and Optimize both.
>
> - serverHost
>
> - serverPort
>
> - useSSL

> ✏️ **Note:** If you have a clustered listener then listener configuration is defined as usual, under following configuration property Campaign > unicaACListener > listenerNode .
>
> There is one additional configuration parameter as "listenerType" which defines whether the listener is for Campaign or Optimize or BOTH.
>
> ListenerType - 1 - Run only Campaign flowcharts
>
> ListenerType - 2 - Run only Optimize sessions
>
> ListenerType - 3 - Run Campaign flowchart and Optimize session

And additional properties present under Campaign > unicaACListener are applicable for the listener.

Verifying that the Unica Campaign listener is running

The Unica Campaign listener must be running before users can work with any Unica Campaign features. The listener automatically creates a separate `unica_acsvr` process for each login and each active flowchart. For example, if one user logs in and then opens a flowchart, the listener creates two instances of `unica_acsvr.exe`.

About this task

Use the following procedure to verify that the Unica Campaign listener is running.

1. Use the appropriate procedure for your operating system:

   In Windows™, look for `unica_aclsnr.exe` on the **Processes** tab of the Windows™ Task Manager.

   In UNIX™, use the `ps` command to look for the Unica Campaign server, as in the following example: `ps -ef | grep unica_aclsnr.`

2. If the listener is not running, restart it:

   In Windows™, run the `cmpServer.bat` script that is in the `bin` directory under your Unica Campaign installation.

   In UNIX™, enter the following command at the system prompt: rc.unica_ac start

   For important details about running the listener, including how to start it automatically, see the *Unica Campaign Administrator's Guide*.

Setting up the Unica Campaign system user

Set up the Unica Campaign system user to directly access databases. For multiple partitions in Unica Campaign, create a system user for each partition.

A system user is a Unica user account that is configured for use by applications.

To avoid presenting users with repeated prompts for login credentials you can associate a system user with one or more data sources. Each data source specifies a user name and password. This allows you to provide a user name and password to access a database or other protected resource by referencing the data source. By adding multiple data sources to the configuration of the system user account, you can allow the system user to access multiple databases.

In Unica Campaign, the system user holds login credentials for accessing system tables and other data sources.

Use an existing or new Unica user account to save credentials for the data sources described here.

You set up Unica users and assign user data sources in the **Settings > Users** area of Unica. See the online help in that section for instructions on how to do this.

Set up a user account to hold credentials for the following data sources.

- Unica Campaign system tables (UA_SYSTEM_TABLES)
- All customer (user) tables

On UNIX™, for the **Alternate Login** attribute of the system user, enter the UNIX™ account of a user in a group that shares privileges with the UNIX™ users of Unica Campaign.

> **Note:** If you have multiple partitions, each partition must have its own system user. The system user cannot be the same across partitions.

Adding data source properties on the **Configuration** page

Use the appropriate data source template to add data source properties on the **Configuration** page for each Unica Campaign data source.

About this task

The Unica Campaign installer imports the template for the database type that you specified for the Unica Platform database when you ran the installer.

If you need other data source templates for additional database types, you must import them manually using the Unica Platform `configTool` utility. You can import as many templates as you need, to correspond to each different type of database you have.

For example, suppose your installation of Unica Platform and Unica Campaign used the following databases:

- Oracle - system tables
- DB2® - customer (user) tables
- DB2® - additional customer (user) tables

In this case, you would import the `DB2Template.xml` template for the two sets of customer (user) tables.

If your Unica Platform and Unica Campaign system tables databases are of the same database type, the installer automatically imports the template that is used for the system tables; in this example, it imports the Oracle template.

For instructions, see Importing data source templates on page 114.

You create a new category from the template, which creates a new set of data source configuration properties. Create as many new categories as needed, one for each data source of that type. In the example above, you would use the Oracle template to create one new category and the DB2® template to create two new categories. See Duplicating a data source template on page 114.

After you add the data source properties, set the data source configuration properties in the categories you created from the templates.

For instructions, see Data source properties on page 115.

Importing data source templates

The Unica Campaign system tables data source (UA_SYSTEM_TABLES) is supported only on Oracle, DB2®, and SQLServer. Import the data source templates for the user tables by using the configTool utility to support the database types that are not supported for the Unica Campaign system tables.

About this task

The Unica Campaign data source templates are located in the `conf` directory under your Unica Campaign installation.

To import and export templates, you use the Unica Platform `configTool` utility, located in the `tools/bin` directory under your Unica Platform installation. If you are not familiar with `configTool`, see configTool on page 142 for details on performing this task.

Here is an example of the command you would use to import the Oracle template into the default partition (Windows™ environment).

```
configTool -i -p "Campaign|partitions|partition1|dataSources" -f full_path_to_directory_containing_your_Oracle_template
\OracleTemplate.xml
```

Duplicating a data source template

Duplicate the data source templates to create a new set of configuration properties in the data source category.

1. On the Configuration page, navigate to the data source template you want to duplicate.

   Unlike other categories, template category labels are in italics and enclosed in parentheses.

2. Click the data source template.
   Result

   The Create category from Template page displays.

3. Enter a name in the **New category name** field (required).

   ✏️ **Note:** The data source category name for the Unica Campaign system tables MUST be `UA_SYSTEM_TABLES`.

4. If desired, edit properties within the new category. You can also do this later.
5. Click **Save and Finish**.

Results

The new category appears in the navigation tree.

Unica Campaign configuration properties

You must specify configuration properties on the **Configuration** page for a basic installation of Unica Campaign. Additionally, you can use the **Configuration** page to specify the properties that perform important functions that you can optionally adjust.

Data source properties

The following table provides information abouint the properties that you must specify for each Unica Campaign data source:

Table 22. Properties for each Unica Campaign data source

Property name	Description
`ASMUserForDBCredentials`	This property must be the same user you already created as the Unica Campaign system user in Setting up the Unica Campaign system user on page 113.
`DSN`	For SQL server, set this property to the DSN (data source name) you created. For Oracle and DB2®, set this property to the database name or the SID (service) name.
`JndiName`	Set this property to the JNDI you created in your application server to connect to this specific data source. For Tomcat, JNDI name should be prefixed with "java:/comp/env/" for example: campaignPartition1DS should be mentioned as java:/comp/env/campaignPartition1DS
`SystemTableSchema`	Not required for SQL server. For other data sources, set this property to the user of the database to which you are trying to connect.
`OwnerForTableDisplay`	Not required for SQL server. For other data sources, set this property to the user of the database to which you are trying to connect.

The data sources are the Unica Campaign system table database and any customer (user) databases you plan to use with Unica Campaign.

Note: The data source category name for the Unica Campaign system tables **must** be `UA_SYSTEM_TABLES`.

For details on setting the values, see the context help for these properties or the *Unica Platform Administrator's Guide*.

In addition to creating and setting data source properties, you must set the following properties on the **Configuration** page for a basic installation of Unica Campaign.

- `Campaign > unicaACOListener > serverHost`
- `Campaign > unicaACOListener > serverPort`
- `Campaign > unicaACListener > serverHost`
- `Campaign > unicaACListener > serverPort`

   For single-node Campaign environment, where the same listener is serving both Campaign flowchart execution and Optimize session execution, `unicaACOListener > serverPort` must have the same value as unicaACListener > serverPort

- For the default partition, set values as required in the categories under `Campaign > partitions > partition1`.

Mapping user tables in Unica Campaign

Mapping user tables is the process of making external data sources accessible in Unica Campaign. A typical user table contains information about your company's customers, prospects, or products. The data source can be a database table or ASCII flat file. You must map any data sources that you configured to make that data accessible to processes in flowcharts.

About this task

For instructions on how to map user tables, see the *Unica Campaign Administrator's Guide*.

Note: User tables are different from system tables. Most Unica Campaign system tables are mapped automatically during initial installation and configuration if you use the system table data source name UA_SYSTEM_TABLES. If a connection issue forces you to map system tables manually, you must log out of Unica Campaign and log in again after you map the tables.

Mapping tables for Unica Optimize

You can exchange data between Unica Optimize and Unica Campaign by mapping the Unica Optimize system tables, contact history tables, and segment membership tables to the Unica Campaign tables.

About this task

Note: Mapping the segment membership system tables in Unica Campaign for use with strategic segments is optional. Map the segment membership system table for an audience only if you use the audience with flowcharts or Unica Optimize sessions that use strategic segments.

For more information about mapping system tables in Unica Campaign, see the *Unica Campaign Administrator's Guide*.

Complete the following steps to map tables for Unica Optimize:

1. Map all system tables.
2. Map all contact history tables.

   Confirm that each audience level in your Unica Campaign configuration has a contact history table. You must map each contact history table. For more information about additional contact history tables, see the *Unica Campaign Administrator's Guide*.

3. **Optional:** For each audience used in Unica Optimize sessions that use strategic segments, map the segment membership system table for the audience to the database table that defines the segment members.

   Requirements when you use strategic segments:

   If you plan to use strategic segments, ensure that there is a segment membership table for each audience level in your Unica Campaign configuration. You must map each segment membership table.

   Create two indexes on the audience segment membership tables. Create the first index on SegmentID and create the second on the audience level columns. The `UA_SegMembership` table ships with an index on CustomerID and on SegmentID.

   For more information about system tables and strategic segments, see the *Unica Campaign Administrator's Guide*.

Unica Optimize system table mapping reference

To integrate Unica Optimize with Unica Campaign, map all the Unica Optimize system tables to the Unica Campaign tables.

You can exchange data between Unica Optimize and Unica Campaign by mapping the Unica Optimize tables in Unica Campaign. Map all the Unica Optimize system tables, contact history tables, and segment membership tables to the Unica Campaign tables.

Use the following table to get information about the Unica Optimize system tables and the corresponding database tables:

Table 23. Unica Optimize system table mapping

Unica Optimize System Table	Database Table
Optimize Session Table	UACO_OptSession

Table 23. Unica Optimize system table mapping

(continued)

Unica Optimize System Table	Database Table
Optimize Process Table	UACO_OptimizeProcess
Optimize Run History Table	UACO_SesnRunHist
Optimize Session Segment Table	UACO_SesnSegment
Optimize Offer Segment Table	UACO_OfferSeg
Optimize Session Offer Table	UACO_SesnOffer
Optimize Rule Table	UACO_Rule
Optimize Exception Rule Table	UACO_RException
Optimize Exception Rules Constraint Table	UACO_RExConstraint
Optimize Rules Constraint Table	UACO_RConstraint
Optimize Rule OfferList Table	UACO_ROfferList
Optimize Rule Offer Table	UACO_ROffer
Optimize Rule Segment Table	UACO_RSegment
Optimize Proposed Contact Base Table	UACO_PCTBase
Optimize Proposed Attribute Base Table	UACO_POABase
Optimize Optimized Contact Base Table	UACO_OCTBase

Unica Optimize contact history table mapping reference

To integrate Unica Optimize with Unica Campaign, map all the Unica Optimize contact history tables to the Unica Campaign tables.

Use the following table to get information about the Unica Optimize contact history tables and the corresponding database tables:

Table 24. Unica Optimize contact history table mapping

Contact History Table	Database Table
Customer Contact History Table	UA_ContactHistory
Customer Detailed Contact History Table	UA_DtlContactHist

The names of additional audience level tables are unique according to your configuration of Unica Optimize.

Verifying the Unica Campaign installation

If you performed all of the steps to install and configure Unica Campaign, deploy the Unica Campaign web application, and configure Unica Campaign after deployment, you are ready to verify your installation.

Before you begin

If you have not already done so, log in to Unica as a user that exists in the Unica Campaign Administrators role (such as asm_admin). In **Settings > Users**, assign at least one security role (for example, Global Policy/Admin) to the new user. After the new user is assigned the role, you can log in to Unica Campaign as that new user.

About this task

Use the following procedure to verify the installation.

1. Log in to Unica.
2. Verify that all system tables exist in the **Settings > Campaign Settings > Manage Table Mappings** window.
3. Create a campaign and create a flowchart in that campaign.

Verifying the Unica Optimize installation

If Unica Optimize is installed correctly, you must be able to use the Optimize process in Unica Campaign when you build the Unica Campaign batch flowcharts. To verify, log in to Unica and confirm that you can access **Campaign > Optimizes**.

Before you begin

📝 **Note:** Before you verify that Unica Optimize is installed or upgraded correctly, make sure that the **Affinium|Campaign| acoInstalled** property is set to true. If the **Affinium|Campaign|acoInstalled** is set to false, you get an error message that indicates Unica Optimize is not installed.

About this task

After you map the Unica Optimize tables to the Unica Campaign tables, you might have to log out and log in to the Unica console to show **Campaign > Optimize sessions**.

Complete the following steps to verify the Unica Optimize installation:

1. Log in to the Unica console.
2. Select **Campaign > Optimizes** from the menu.
   Result
   The **All Optimize sessions** page is displayed. Verify that you can view and manage your Unica Optimize sessions.

Setting properties for integration with Unica products

Unica Campaign integrates with various applications. If required, you can specify configuration properties to set up an integration between Unica Campaign and other products.

Additional configuration for enhanced security

The procedures in this section describe some additional configurations in the web application server. They are optional, but you might want to perform them for enhanced security.

Disabling the X-Powered-By flag

If your organization is concerned that the X-Powered-By flag in the header variable is a security risk, you can disable it using this procedure.

1. If you are using WebLogic, in the administration console, under *domainName* **> Configuration > Web Application**, set **X-Powered-By Header** to **X-Powered-By Header will not be sent**.
2. If you are using WebSphere®, perform the following steps.
   a. In the WebSphere® administration console, navigate to **Servers > Server Types > WebSphere application servers >** *server_name* **> Web Container Settings > Web container**.
   b. Under **Additional Properties** select **Custom Properties**.
   c. On the Custom Properties page, click **New**.
   d. On the Settings page, create a custom property named `com.ibm.ws.webcontainer.disablexPoweredBy` and set the value to `false`.
   e. Click **Apply** or **OK**.
   f. Click **Save** on the console task bar to save your configuration changes.
   g. Restart the server.

Configuring a restrictive cookie path

In your web application server, you can restrict cookie access to a specific application for enhanced security. If not restricted, cookies are valid across all the deployed applications.

1. If you are using WebLogic, perform the following steps.

    a. Extract the `weblogic.xml` file from the WAR or EAR package where you want to add the restrictive cookie path.

    b. Add the following to the `weblogic.xml` file, where *context-path* is the context path for the deployed application. For Unica applications the context path is typically /unica.

    ```
    <session-descriptor>
        <session-param>
          <param-name>CookiePath</param-name>
          <param-value>/context-path> </param-value>
        </session-param>
    </session-descriptor>
    ```

    c. Rebuild the WAR or EAR file.

2. If you are using WebSphere®, perform the following steps.

    a. In the WebSphere® administration console, navigate to the **Session Manager > Cookie** tab.

    b. Set the **Cookie Path** to the context path of your application.

    For Unica applications the context path is typically /unica.

Starting the Unica Campaign Advance search agent

You can use the Unica Campaign Advance search agent to search for flowcharts that you have created in Unica Campaign. You can search for flowcharts only if the Advanced search agent is running when the a flowchart is created or modified.

Before you begin
Unica Platform and Unica Campaign web applications must be deployed and running before you start the Unica Campaign Advance search agent.

To start the Unica Campaign Advance search agent manually, complete the following steps:

1. For Windows, go to the bin directory under your Unica Campaign installation and run `advSrchAgent.bat`.
   Result

   The `advSrchAgent.bat` file keeps running at the prompt.

2. For Unix, go to the bin directory under your Unica Campaign installation and run `advSrchAgent.sh`.
   Result

   The `advSrchAgent.sh` file keeps running at the prompt.

Example

For a Unix operating system, run the following command to start the Unica Campaign Advance search agent.

```
./advSrchAgent.sh partition1
Usage:
./advSrchAgent.sh <start | stop> <partition_name> [cluster_home] [-u <user_name>] [-p <password>]
Where:
<start | stop>    : Start or stop the agent.
<partition_name> : Name of the partition from which flowchart files are to be exported.
[cluster_home]    : Home directory of clustered environment, if clustering is enabled.
[-u <user_name>] : User name for user who has read access to all flowchart files in the partition specified.
[-p <password>]   : Password of the user specified
```

Chapter 8. Configuring multiple partitions for Unica Campaign

In the Unica family of products, partitions provide a way to secure the data that is associated with different groups of users. When you configure Unica Campaign or a related Unica application to operate with multiple partitions, users can see each partition as a separate instance of the application. There is no indication that other partitions exist on the same computer.

How multiple partitions work

For the Unica applications that operate with Unica Campaign, you can configure the application within the partitions where you have configured an instance of Unica Campaign. Application users within each partition can access the Unica Campaign functions, data, and customer tables that are configured for Unica Campaign in the same partition.

Partition benefits

Multiple partitions are useful for setting up strong security between groups of users, because each partition has its own set of Unica Campaign system tables. Multiple partitions cannot be used when groups of users must share data.

Each partition has its own set of configuration settings, and Unica Campaign can be customized for each group of users. However, all partitions share the same installation binaries. With the same binaries for all partitions, you can minimize the installation and upgrade efforts for multiple partitions.

Partition user assignment

Access to partitions is managed through membership in Unica Platform groups.

Except for the partition superuser (platform_admin), each HCL user can belong to one partition. A user who requires access to multiple partitions must have a separate user account for each partition.

If there is only one partition, users do not have to be explicitly assigned to that partition to have access to Unica Campaign.

Partition data access

In a multi-partition configuration, partitions have the following security characteristics:

- A user cannot access a partition if the user is not a member of a group that is assigned to a partition.
- A user of one partition cannot view or change data in another partition.
- Users cannot navigate the Unica Campaign file system above the root directory of their partition from the browse dialog boxes in Unica Campaign. For example, if two partitions, named partition1 and partition2 exist, and you belong to the group that is associated with partition1, you cannot navigate the directory structure for partition2 from a dialog box.

Setting up multiple partitions

Configure multiple partitions in Unica Campaign to isolate and secure data for different groups of Unica Campaign users. As each partition has its own set of configuration properties, you can customize Unica Campaign for each group of users.

Before you begin

Before you configure additional partitions in Unica Campaign, complete the following tasks for each partition that you are configuring:

1. Create database or schema for the Unica Campaign system tables
2. Create ODBC or native connections
3. Create JDBC connections in the web application server

Complete the following tasks to set up multiple partitions for Campaign:

1. Create the system tables database and any other data sources required for your partitions. Then configure the JDBC and ODBC or native connections required to access your data sources.
2. For each partition, create a different schema in your database to hold the system tables.
   Use the database-specific scripts packaged with Campaign to create and populate the system tables.
3. Complete the following tasks to create a directory structure for each additional partition:

   Note: Save a clean copy of the original partition1 directory to serve as a backup.

   a. In the `partitions` directory under your Unica Campaign installation, create a duplicate of the default `partition1` directory for each partition that you are adding, ensuring that you include all sub-directories.

   b. Name each partition directory with a unique name. Use these exact same names for the partitions when you later create their configuration trees on the **Configurations** page.
   Example
   If you want to create a second partition and created the directory named `Campaign/partitions/partition2`, when you create the configuration tree on the **Configurations** page, you must use the name "partition2" to identify this partition.

   c. Delete any files that exist in the duplicate partition sub-directories.

4. Complete the following tasks to use the partitionTool utility with the -s option to clone the default partition:

   Note: If you decide not to use this option, you must stop the web application server where Unica Platform is deployed before performing this procedure.

   a. Set the `JAVA_HOME` environment variable, either in the `setenv` script located in the `tools/bin` directory of your Unica Platform installation, or in the command line window in which you run the `partitionTool` utility.

   b. Open a command line window and run the utility from the `tools/bin` directory under your Unica Platform installation. Use the appropriate command and options (as described in the *Unica Platform Administrator's Guide*) to achieve your desired results.
   Example
   partitionTool -c -s partition1 -n partition2

   c. Repeat this procedure for each new partition you need to create.

d. When you are finished, stop and restart the application server where Unica Platform is deployed to see the groups that were created.

> **Note:** For complete information on using this utility, see partitionTool on page 151.

5. Complete the following tasks to create the partition structure on the **Configuration** page by using the `New partitionTemplate` for each new partition:

   a. On the **Configuration** page, navigate to **Campaign > partitions** and click *(partitionTemplate)*.

   If you do not see the *(partitionTemplate)* property in the list, use the `configTool` utility to import the partition template, using a command similar to the following:

   ```
   configTool -i -p "Affinium|Campaign|partitions"
     -f <CAMPAIGN_HOME>/conf/partitionTemplate.xml
   ```

   Replace *CAMPAIGN_HOME* with the actual path to your Unica Campaign installation.

   The `configTool` utility is found in the `tools` directory under your Unica Platform installation. See the *Unica Platform Administrator's Guide* for information on using this utility.

   Result

   In the right pane, you see the partitionTemplate pane with an empty **New category name** field.

   b. Enter the name for the new partition, using the same name that you used when creating the partition's directory structure in the file system in Setting up multiple partitions on page 123.

   c. Click **Save Changes**.
   Result
   You see the new partition structure, with the same categories and properties as in the partition template.

> **Note:** If campaign is having multiple partitions, then during upgrade user need to upgrade all the partitions.

Partition superuser

To administer security across all users of Unica Platform, a user account must exist that has access to all security settings and user accounts in the system.

By default, this user account is platform_admin. This user account does not belong to any one partition; instead, it has access to all user accounts in all partitions.

The administrator can create additional users with the same level of access. To be a partition superuser, an account must have Admin access to the Unica Platform and have full access to the Users, User Groups, and User Permissions pages. The partition superuser does not require access to product-specific security pages, such as Unica Campaign security pages.

Configuring data source properties for partitions

You must configure data source properties for each partition that you create. Use the appropriate data source template to create data source properties.

About this task

The Unica Campaign installer imports the template for the database type that you specified for the Unica Platform database when you ran the installer.

If you need other data source templates for additional database types, you must import them manually using the Unica Platform configTool utility. You can import as many templates as you need to correspond to each different type of database you have.

For example, suppose your installation of Unica Platform and Unica Campaign used the following databases:

- Oracle - system tables
- DB2® - customer (user) tables
- DB2® - additional customer (user) tables

In this case, you would import the `DB2Template.xml` template for the two sets of customer (user) tables.

If your Unica Platform and Unica Campaign system tables databases are of the same database type, the installer automatically imports the template used for the system tables; in this example, it imports the Oracle template.

> **Note:** When you create a new partition, you must use the configTool utility to import the data source templates for the system tables and user tables.

For instructions, see Importing data source templates on page 114.

You create a new category from the template, which creates a new set of data source configuration properties. Create as many new categories as needed, one for each data source of that type. In the example above, you would use the Oracle template to create one new category and the DB2® template to create two new categories. See Duplicating a data source template on page 114.

After you add the data source properties, set the data source configuration properties in the categories you created from the templates.

For instructions, see Unica Campaign configuration properties on page 115.

Complete the following tasks to configure data source properties for each partition:

1. Add data source configuration properties to the **Configuration** page for each Unica Campaign data source by using the appropriate data source template.
   The Unica Campaign installer imports the template for the database type that you specified for the Unica Platform database when you ran the installer. If you need other data source templates for additional database types, you must

import them manually using the Unica Platform `configTool` utility. You can import as many templates as you need, to correspond to each different type of database you have.

Example

For example, suppose your installation of Unica Platform and Unica Campaign used the following databases:

- Oracle - system tables
- DB2® - customer (user) tables
- DB2® - additional customer (user) tables

In this case, you would import the `DB2Template.xml` template for the two sets of customer (user) tables.

Result

If your Unica Platform and Unica Campaign system tables databases are of the same database type, the installer automatically imports the template used for the system tables; in this example, it imports the Oracle template. For instructions, see Importing data source templates on page 114.

2. Create a new category from the template, which creates a new set of data source configuration properties. Create as many new categories as needed, one for each data source of that type. In the example above, you would use the Oracle template to create one new category and the DB2® template to create two new categories. See Duplicating a data source template on page 114.

3. Set the data source configuration properties for each Unica Campaign data source.
   See Unica Campaign configuration properties on page 115 for more information.

Setting up system users for Unica Campaign

You can associate a system user with one or more Unica Platform data sources to avoid presenting users with repeated prompts for login credentials. Each data source specifies a user name and password. You can provide a user name and password to access a database or other protected resources by referencing the data source. By adding multiple data sources to the configuration of the system user account, you can allow the system user to access multiple databases.

About this task

Unica applications might require a system user account configured with the following attributes:

- Login credentials for accessing system tables or other data sources.
- Specific permissions to create, modify, and delete objects within the system.

For more information about setting up a new user and assigning a data source to a user, see the *Unica Platform Administrator's Guide*.

Complete the following actions to set up system users for Unica Campaign:

1. Use an existing or a new user account to save credentials for the following data sources:
   - Unica Campaign system tables
   - All customer (user) tables

2. On UNIX™, for the **Alternate Login** attribute of the system user, enter the UNIX™ name for a user in a group that shares privileges with the UNIX™ users of Unica Campaign.

> 📝 **Note:** If you have multiple partitions, you must create a system user for each partition.

Using IBM Cognos Reports for multiple partitions

If you want to use IBM Cognos® Reports for multiple partitions in Unica Campaign, Unica Deliver, or Unica Interact, you must configure IBM Cognos® reports packages for each partition.

For instructions, see the *Unica Reports Installation and Configuration Guide*.

Using Unica Insights Reports for multiple partitions

If you want to use Unica Insights Reports for multiple partitions in Unica Campaign or Unica Interact, you must configure Unica Insights reports packages for each partition. For instructions, see the Unica Insights Reports installation and configuration guide.

Assigning roles, permissions, and groups for partitions

Before you can use the partitions that you have configured for Unica Campaign, you must assign roles to the user with administrator role in each partition. You must also assign a group to each partition.

Use the partitionTool utility to create a default admin user for each partition that you create.

Assign roles to each partition Admin user - The `partitionTool` utility creates a default Admin user for each partition you create. On the **Users** page, assign at least one security role (for example, Global Policy/Admin) to your new user. Once the new user has been assigned the role, you can log into the Unica Campaign partition as that new user.

If you plan to enable Unica Deliver in multiple Unica Campaign partitions, you must configure a corresponding Unica Deliver partition for each Unica Campaign partition. For more information about creating additional partitions for Unica Deliver, see Configuring multiple partitions in Unica Deliver on page 131.

Chapter 9. Configuring multiple partitions for Unica Optimize

In the Unica family of products, partitions provide a way to secure the data that is associated with different groups of users. When you configure Unica Campaign or a related Unica application to operate with multiple partitions, users can see each partition as a separate instance of the application. There is no indication that other partitions exist on the same computer.

How multiple partitions work

For the Unica applications that operate with Unica Campaign, you can configure the application within the partitions where you have configured an instance of Unica Campaign. Application users within each partition can access the Unica Campaign functions, data, and customer tables that are configured for Unica Campaign in the same partition.

Partition benefits

Multiple partitions are useful for setting up strong security between groups of users, because each partition has its own set of Unica Campaign system tables. Multiple partitions cannot be used when groups of users must share data.

Each partition has its own set of configuration settings, and Unica Campaign can be customized for each group of users. However, all partitions share the same installation binaries. With the same binaries for all partitions, you can minimize the installation and upgrade efforts for multiple partitions.

Partition user assignment

Access to partitions is managed through membership in Unica Platform groups.

Except for the partition superuser (platform_admin), each HCL user can belong to one partition. A user who requires access to multiple partitions must have a separate user account for each partition.

If there is only one partition, users do not have to be explicitly assigned to that partition to have access to Unica Campaign.

Partition data access

In a multi-partition configuration, partitions have the following security characteristics:

- A user cannot access a partition if the user is not a member of a group that is assigned to a partition.
- A user of one partition cannot view or change data in another partition.
- Users cannot navigate the Unica Campaign file system above the root directory of their partition from the browse dialog boxes in Unica Campaign. For example, if two partitions, named partition1 and partition2 exist, and you belong to the group that is associated with partition1, you cannot navigate the directory structure for partition2 from a dialog box.

Setting up multiple partitions for Unica Optimize

Use multiple partitions in Unica Optimize so that different groups of users have access to different sets of Unica Optimize and Unica Campaign data.

About this task

With 12.0 release onwards - Unica Optimize is installed along with Unica Campaign. Default parition1 directory created for Campaign is used for Optimize. Additional partition created for Unica Campaign is available for Optimize. Since V 12.0 release, Optimize is coming by default with Campaign application. Once user creates partition it supports the partition for Optimize as well. In other words, if you require partition for Optimze, then you require to create partition for Campaign as well.

Complete the following steps to set up multiple partitions for Unica Optimize:

1. Create your Unica Campaign partitions.
2. Copy the contents of the old ContactOptimization/partitions/partition2 directory, along with all of its (empty) subdirectories, into the directory you created for the new partition.

   For example, copy the contents of the partition2 directory into the

   Campaign_Home/partitions/partition2 directory

3. Run the SQL scripts to create and populate the Unica Optimize system tables in the database that is configured for the new partition.

   For more information about running the SQL scripts, see the "Creating and populating the Unica Optimize system tables manually" topic.

4. Repeat Steps 2 through 4 for each partition you must create. Use the most recently created directory as your starting point.

   Note: By default, partitions are installed under the partitions directory.

Results

Chapter 10. Configuring multiple partitions in Unica Deliver

Configure multiple partitions in Unica Deliver to isolate and secure data for different groups of Unica Deliver users. As each partition has its own set of configuration properties, you can customize Unica Deliver for each group of users.

Installing Unica Deliver creates a default partition for Unica Deliver in the Unica Platform. You can configure more partitions for Unica Deliver. Each partition that you create for Unica Deliver operates in conjunction with a partition created for Unica Campaign.

> **Note:** To configure multiple partitions in Unica Deliver you must configure corresponding partitions in Unica Campaign.

Adding new partitions for Unica Deliver requires making changes in the Unica Platform configurations for Unica Deliver and Unica Campaign.

> **Important:** After you change the Unica Deliver and Unica Campaign configurations, you must restart the web application server that hosts Unica Campaign and restart the Response and Contact Tracker (RCT). You might also need to restart the Unica Campaign listener.

Be sure to back up your existing configurations before you make changes.

Partitions for Unica Deliver: Overview

Creating partitions for Unica Deliver allows you to isolate and secure data for different groups of users. Each partition appears to users as a separate instance of Unica Deliver, with no indication that other partitions exist on the same system. Because each partition has its own set of configuration properties, you can customize Unica Deliver for each group of users.

Users within each partition can access only those functions, data, and customer tables configured for the partition. For example, if you create partitions named `partition1` and `partition2`, Unica Deliver users working in `partition1` can select email recipients from customer tables configured in `partition1`, but not in `partition2`. HCL does not recommend creating multiple partitions if users need to share data.

When you work with multiple partitions, you should understand characteristics that are unique to partitions for Unica Deliver and how partitions for Unica Deliver relate to partitions for Unica Campaign. You should also be familiar with the workflow for creating and configuring multiple partitions for Unica Deliver.

Characteristics of partitions for Unica Deliver

Note the following when you create and configure new partitions for Unica Deliver.

- You create partitions for Unica Deliver differently than you do for Unica Campaign.

  You create new partitions for Unica Deliver using a partition template available in the Deliver configuration properties in Unica Platform.

- The name of each Unica Deliver partition must exactly match the name of its corresponding Unica Campaign partition.

- Each partition that you create for Unica Deliver must be able to connect to HCL Unica.

  You must request a separate HCL Unica account for each partition. will provide a user name and password connected to the account. You must configure a Unica Platform data source that can provide these access credentials automatically when Unica Deliver connects to HCL Unica.

  For more information about how to request an account, see the *Unica Deliver Startup and Administrator's Guide*.

Relationship to partitions in Unica Campaign

Each partition in Unica Deliver operates in conjunction with a specific partition created in the Unica Platform for Unica Campaign. The Unica Campaign partition provides the following.

- Unica Campaign schema that contains the Deliver system tables
- File structure for Unica Campaign in the partition, including directories used by Unica Deliver to create and process recipient lists
- Configuration properties associated with creating recipient lists and enabling Unica Deliver within the partition

Because Unica Deliver operates with Unica Campaign within a specific partition, the Unica Deliver and Unica Campaign partition structures must specify the same name. The partition names must match exactly.

Roadmap for configuring multiple partitions in Unica Deliver

To create a partition in Unica Deliver, use the exact same name as the partition in Unica Campaign that exists in the Unica Platform configurations.

Before you create a new partition for Unica Deliver, confirm that you have satisfied all the Unica Deliver prerequisites for the partition in Unica Campaign and Unica Deliver.

Complete the following steps to create a new partition for Unica Deliver.

Creating a new partition for Unica Deliver

Installing Unica Deliver creates a default partition for Unica Deliver in Unica Platform. You can create multiple partitions for Unica Deliver to isolate and secure data for different groups of users.

Before you begin

Before you create and configure partitions for Unica Deliver, you must meet the following requirements for Unica Deliver and Unica Campaign:

- Before you create multiple partitions in Unica Deliver, complete the following tasks for Unica Deliver:
    - Contact Support to request an account and credentials for each partition. Each partition requires a separate Unica Journey account and access credentials. For more information, *Unica Deliver Startup and Administrator's Guide.*
    - Create a system user that can access the Unica Deliver system tables that you will create in the Unica Campaign schema for the partition.

      You can update the system user that you created for the Unica Campaign partition so that it can also access the Unica Deliver system tables.

- Before you create multiple partitions in Unica Deliver, complete the following tasks in Unica Campaign:
    - Create a partition in Unica Campaign to operate with the partition that you are creating for Unica Deliver. Note the name of the partition.
    - Create Unica Campaign system tables in the Unica Campaign partition.
    - Configure a system user to access the system tables in the partition.

      If you have multiple partitions, each partition must have its own system user. The system user cannot be the same across partitions.

About this task

The installer registers the Unica Deliver configuration properties and a default partition during initial installation. The default partition includes a template that you can copy to create additional partitions.

Complete the following actions to create a new partition for Unica Deliver:

1. Navigate to `Deliver > partitions > (partition)` and duplicate the partition template.
2. Name the new partition.

   Note: Unica Deliver does not support deleting a partition after you have created it.

Example

Identifying the partition template

The Configuration page displays the Unica Deliver partition template in the navigation tree for the default partition. You can identify the partition template in the tree because its label is in italics and enclosed in parentheses.

What to do next

Naming the new partition

The following restrictions apply when you name a new partition.

- The name must be unique among categories that are siblings in the tree (that is, among categories that share the same parent category).
- The partition name may not start with a period. Also, the following characters are not allowed in partition names.

! " " ' # $ % & ( ) * + : ; ,

^ < > + ? @ [ ] { } / \ ` ~
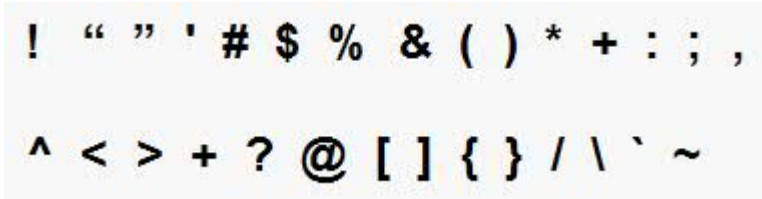
Note: Because Unica Deliver operates with Unica Campaign within a specific partition, the partitions for Unica Deliver and Unica Campaign must specify the same partition name.

Preparing Unica Deliver system tables for the partition

For each partition that you create for Unica Deliver, you must create, populate and configure the Unica Deliver system tables that are used by the partition in the Unica Campaign schema.

Complete the following actions to prepare the Unica Deliver system tables for a partition:

1. Create the Unica Deliver system tables.
   In your database client, run the SQL scripts for *creating* system tables against the Unica Campaign database.
2. Populate the tables that you created.
   Use the database client to run the scripts for *populating* tables against the Unica Campaign database. For more information about the SQL scripts, see the reference table for script names and locations at .
3. Set the following configuration properties in the Deliver configuration for the partition, specifying the user name and platform data source that you configured for the partition's Unica Campaign system user:
   - ° `Deliver > partitions > partition [n] < dataSources > systemTables > asmUserForDBCredentials`
   - ° `Deliver > partitions > partition [n] < dataSources > systemTables > amDataSourceForDBCredentials`

   Unica Deliver accesses the partition's system tables through a system user configured in Unica Platform. A Unica Platform data source added to the system user provides the required access credentials. Because the Unica Deliver system tables exist in the Unica Campaign schema for the partition, you can use the system user that you created to access the Unica Campaign schema to access the partition's Unica Deliver system tables.
4. Update the following properties in the partition's 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`

See the Unica Platform online help for each property to learn more about setting the configuration properties. For additional information about these configuration properties and configuring Unica Deliver, see the *Unica Deliver Startup and Administrator's Guide*.

Creating and populating the Unica Deliver system tables manually

For Unica Deliver, you must create additional system tables in the Unica Campaign schema and then populate the tables with initial data. The Unica Campaign installer automatically creates and populates the Unica Deliver system tables in the Unica Campaign schema if you select the option to create system tables automatically. However, if you do not select the option, you must create and populate the Unica Deliver system tables manually.

Use your database client to run the appropriate script against the Unica Campaign database.

> **Note:** The Unica Deliver installation directory is a subfolder within the Unica Campaign directory.

Scripts for creating Unica Deliver tables

Campaign provides the `ace_op_systab` scripts for creating Unica Deliver tables in your local environment.

If your Unica Campaign system tables are configured for Unicode, find the appropriate script in the `ddl/unicode` directory under your Unica Deliver installation.

If your Unica Campaign system tables are not configured for Unicode, use the non-Unicode scripts in the `ddl` directory under your Unica Deliver installation. Use the following table to see the scripts that you must use:

Table 25. Scripts for creating Unica Deliver tables

Data source type	Script name
IBM® DB2®	ace_op_systab_db2.sql The user tablespace and system temporary tablespace where the system tables reside must each have a page size of 16K or greater.
Microsoft™ SQL Server	ace_op_systab_sqlsvr.sql
MariaDB	ace_op_systab_mariadb.sql
Oracle	ace_op_systab_ora.sql

Scripts for populating Unica Deliver tables

HCL provides the `ace_op_populate_systab` scripts for populating the Unica Deliver tables in your local environment.

The populate scripts are stored in the `ddl` directory of your Unica Deliver installation. HCL provides only one version of the populate scripts because you can use them for either Unicode or non-Unicode tables.

> **Note:** The Unica Deliver installation directory is a subfolder within the Unica Campaign directory.

Use the following table to see the scripts that you must use:

Table 26. Scripts for populating Unica Deliver tables

Data source type	Script name
IBM® DB2®	ace_op_populate_systab_db2.sql
MariaDB	ace_op_populate_systab_mariadb.sql
Microsoft™ SQL Server	ace_op_populate_systab_sqlsvr.sql
Oracle	ace_op_populate_systab_ora.sql

Configuring a system user to access HCL Unica

Unica Deliver components must be able to access the HCL Unica without requiring manual entry of login credentials. To establish automatic login, define a system user in Unica 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 Unica Campaign system user creates a single user that can automatically access HCL Unica and the Unica Deliver system tables in the Unica Campaign schema.

The credentials required to access HCL Unica are the user name and password that HCL provides for your hosted messaging account.

For specific information about how to configure a system user to communicate with HCL Unica, 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

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

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

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

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

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

Enabling Unica Deliver in Unica Campaign for the new partition

To allow users of the new Unica Deliver partition to access the Unica Deliver features in Unica Campaign, enable Unica Deliver in the Unica Campaign partition by updating the `deliverInstalled` configuration property for the corresponding Unica Campaign partition.

About this task

For example, the Unica Deliver mailing tab does not appear in the Unica Campaign interface until you enable Unica Deliver in the Unica Campaign configuration.

You enable Unica Deliver in the partition by updating the `deliverInstalled` configuration property for the corresponding Unica Campaign partition.

In the Unica Platform configurations, navigate to **Campaign | partitions | partition[n] | server | internal** and set the `deliverInstalled` property to `yes`.

Specifying the Recipient List Uploader location for Unica Deliver

For each partition where you enable Unica Deliver, specify the location of the Recipient List Uploader (RLU). The RLU uploads the output list tables data and associated metadata to the remote services hosted by HCL.

About this task

During initial installation, the Unica Campaign installer automatically adds the location of the RLU to the configuration for default partition (partition1). However, when you add new partitions to your environment, you must manually configure all new partitions to reference the correct location. Because there is only one RLU for each Unica Deliver installation, all partitions access the same plugin file in the local file system of the machine that hosts the Unica Campaign web application.

1. In the configuration for **partition1** in your Unica Campaign installation, navigate to `Campaign > partitions > partition1 > Deliver > deliverPluginJarFile`.

    The value for this property is the full path to the plugin file (`deliverplugin.jar`) that operates as the RLU.

For example: `C:\Unica\Deliver\plugin\deliverplugin.jar`

2. Copy the value for the `DeliverPluginJarFile` property.

3. Navigate to the `DeliverPluginJarFile` for the new partition and enter the path you copied from **partition1**.

All partitions must use the same location for the RLU.

Restarting system components after configuring Unica Deliver

After you change the Unica Deliver and Unica Campaign configurations, you must restart the Unica Campaign web application server, the Response and Contact Tracker (RCT), and the Unica Campaign listener.

1. Restart the web application server for Unica Campaign.

   For instructions, see the documentation for your web application server.

   To verify that the server has started, log in to your Unica installation, access Unica Campaign, and confirm that you can open an existing mailing.

2. Restart the Response and Contact Tracker (RCT).
   Example

   To restart the RCT manually, run the `rct` script in the `bin` directory under your Unica Deliver installation, as follows: `rct start`.

   To stop the RCT manually run the `rct` script in the `bin` directory under your Unica Deliver installation, as follows: `rct stop`.

   If RCT is configured to run as a service, restart the RCT service. The first time that you restart the RCT as a service, you must manually restart the RCT afterward.

   For more information, see Unica Deliver Response and Contact Tracker (RCT) script on page 158.

3. Restart the Unica Campaign listener:
   Example
   ◦ On Windows™, run the `cmpServer.bat` file in the `bin` directory under your Unica Campaign installation.
   ◦ On UNIX™, run the following command as root: `./rc.unica_ac start`

Results

Testing Unica Deliver partition configuration and connections

Run the RLU script provided by Unica Deliver to verify the partition configuration and its connection to HCL Unica. You must also confirm if you can access the mailing interface from the partition.

Before you begin

⚠️ **Important:** Before you begin, if you have changed the Unica Campaign or Unica Deliver configurations, confirm that you have restarted the web application server that hosts Unica Campaign and that you have restarted the Response and Contact Tracker.

About this task

For more information on how to test the partition, see the *Unica Deliver Startup and Administrator's Guide*.

Chapter 11. Unica Platform utilities

This section provides an overview of the Unica Platform utilities, including some details that apply to all the utilities and which are not included in the individual utility descriptions.

Location of utilities

Unica Platform utilities are located in the `tools/bin` directory under your Unica Platform installation.

List and descriptions of utilities

The Unica Platform provides the following utilities.

- Clientdetails - generates a key for a client application like Unica Journey to authenticate with one Unica Platform instance.
- - registers alerts and configurations for Unica products
- - imports, exports, and deletes configuration settings, including product registrations
- - creates data filters
- - encrypts and stores passwords
- - encrypt the database passwords that the Tomcat Application server uses internally
- - creates database entries for partitions
- - populates the Unica Platform database
- - Update scheduler jobs created in version 11.1 and older versions
- - restores a user with the platformAdminRole role
- - lists or starts Unica Scheduler jobs that are configured to listen for a trigger.
- insightsdbutil - Installer places report design files which possesses database connection tokens. You must update them for your system database. You must run `insightsdbutil.sh/bat` utility to update the same. See the Unica Insights Installation and Configuration Guide for more details.

Prerequisites for running Unica Platform utilities

The following are prerequisites for running all Unica Platform utilities.

- Run all utilities from the directory where they are located (by default, the `tools/bin` directory under your Unica Platform installation).
- On UNIX™, the best practice is to run the utilities with the same user account that runs the application server on which Unica Platform is deployed. If you run a utility with a different user account, adjust the permissions on the `platform.log` file to allow that user account to write to it. If you do not adjust permissions, the utility is not able to write to the log file and you might see some error messages, although the tool should still function correctly.

Authentication of utilities

Utilities such as `configTool` and other Unica back end utilities are designed to be used by system administrators and require physical access to the host servers for them to be invoked. For this reason, authentication for these utilities has been designed to be independent of the UI authentication mechanism. Access to these utilities is available to users with Unica Platform administrator privileges. Access to these utilities is expected to be locally defined in Unica Platform and authenticated against the same.

Troubleshooting connection issues

All of the Unica Platform utilities except `encryptPasswords` interact with the Unica Platform system tables. To connect to the system table database, these utilities use the following connection information, which is set by the installer using information provided when the Unica Platform was installed. This information is stored in the `jdbc.properties` file, located in the `tools/bin` directory under your Unica Platform installation.

- JDBC driver name
- JDBC connection URL (which includes the host, port, and database name)
- Data source login
- Data source password (encrypted)

In addition, these utilities rely on the `JAVA_HOME` environment variable, set either in the `setenv` script located in the `tools/bin` directory of your Unica Platform installation, or on the command line. The Unica Platform installer should have set this variable automatically in the `setenv` script, but it is a good practice to verify that the `JAVA_HOME` variable is set if you have a problem running a utility. The JDK must be the Sun version (not, for example, the JRockit JDK available with WebLogic).

Special characters

Characters that are designated as reserved characters in the operating system must be escaped. Consult your operating system documentation for a list of reserved characters and how to escape them.

Standard options in Unica Platform utilities

The following options are available in all Unica Platform utilities.

`-l logLevel`

Set the level of log information displayed in the console. Options are `high`, `medium`, and `low`. The default is `low`.

`-L`

Set the locale for console messages. The default locale is en_US. The available option values are determined by the languages into which the Unica Platform has been translated. Specify the locale using the ICU locale ID according to ISO 639-1 and ISO 3166.

`-h`

Display a brief usage message in the console.

```
-m
```

Display the manual page for this utility in the console.

```
-v
```

Display more execution details in the console.

alertConfigTool

Notification types are specific to the various Unica products. Use the `alertConfigTool` utility to register the notification types when the installer has not done this automatically during installation or upgrade.

Syntax

```
alertConfigTool -i -f importFile
```

Commands

`-i -f importFile`

Import alert and notification types from a specified XML file.

Example

- Import alert and notification types from a file named `Platform_alerts_configuration.xml` located in the `tools\bin` directory under the Unica Platform installation.

  ```
  alertConfigTool -i -f Platform_alerts_configuration.xml
  ```

configTool

The properties and values on the **Configuration** page are stored in the Unica Platform system tables. You can use the `configTool` utility to import and export configuration settings to and from the system tables.

When to use configTool

You might want to use `configTool` for the following reasons.

- To import partition and data source templates that are supplied with Unica Campaign, which you can then modify and duplicate by using the **Configuration** page.
- To register (import configuration properties for) Unica products, if the product installer is unable to add the properties to the database automatically.
- To export an XML version of configuration settings for backup or to import into a different installation of Unica.
- To delete categories that do not have the **Delete Category** link. You do this by using `configTool` to export your configuration, then manually deleting the XML that creates the category and by using `configTool` to import the edited XML.

⚠️ **Important:** This utility modifies the `usm_configuration` and `usm_configuration_values` tables in the Unica Platform system table database, which contains the configuration properties and their values. For best results, either create backup copies of these tables, or export your existing configurations by using `configTool` and back up the resulting file so you have a way to restore your configuration if you make an error when you use `configTool` to import.

Syntax

```
configTool -d -p "elementPath" [-o]

configTool -i -p "parent ElementPath" -f importFile [-o]

configTool -x -p "elementPath" -f exportFile

configTool -vp -p "elementPath" -f importFile [-d]

configTool -r productName -f registrationFile [-o] configTool -u productName
```

Commands

-d -p "*elementPath*" [o]

Delete configuration properties and their settings, specifying a path in the configuration property hierarchy.

The element path must use the internal names of categories and properties. You can obtain them by going to the **Configuration** page, selecting the wanted category or property, and looking at the path that is displayed in parentheses in the right pane. Delimit a path in the configuration property hierarchy by using the `|` character, and surround the path with double quotation marks.

Note the following.

- Only categories and properties within an application can be deleted by using this command, not whole applications. Use the `-u` command to unregister a whole application.
- To delete categories that do not have the **Delete Category** link on the **Configuration** page, use the `-o` option.

When you use `-d` with the `-vp` command, the configTool deletes any child nodes in the path you specify if those nodes are not included in the XML file you specify.

-i -p "*parentElementPath*" -f *importFile* [o]

Import configuration properties and their settings from a specified XML file.

To import, you specify a path to the parent element under which you want to import your categories. The `configTool` utility imports properties under the category you specify in the path.

You can add categories at any level below the top level, but you cannot add a category at same level as the top category.

The parent element path must use the internal names of categories and properties. You can obtain them by going to the **Configuration** page, selecting the required category or property, and looking at the path that is displayed in parentheses in the right pane. Delimit a path in the configuration property hierarchy by using the `|` character and surround the path with double quotation marks.

You can specify an import file location relative to the `tools/bin` directory or you can specify a full directory path. If you specify a relative path or no path, `configTool` first looks for the file relative to the `tools/bin` directory.

By default, this command does not overwrite an existing category, but you can use the `-o` option to force an overwrite.

`-x -p "elementPath" -f exportFile`

Export configuration properties and their settings to an XML file with a specified name.

You can export all configuration properties or limit the export to a specific category by specifying a path in the configuration property hierarchy.

The element path must use the internal names of categories and properties, which you can obtain by going to the **Configuration** page, selecting the wanted category or property, and looking at the path that is displayed in parentheses in the right pane. Delimit a path in the configuration property hierarchy by using the `|` character and surround the path with double quotation marks.

You can specify an export file location relative to the current directory or you can specify a full directory path. If the file specification does not contain a separator (/ on UNIX™, / or \ on Windows™), `configTool` writes the file to the `tools/bin` directory under your Unica Platform installation. If you do not provide the `xml` extension, `configTool` adds it.

`-vp -p "elementPath" -f importFile [-d]`

This command is used mainly in manual upgrades, to import configuration properties. If you applied a fix pack that contains a new configuration property, and you then upgrade, importing a configuration file as part of a manual upgrade process can override values that were set when the fix pack was applied. The `-vp` command ensures that the import does not override previously set configuration values.

> ⚠️ **Important:** After you use the `configTool` utility with the -vp option, you must restart the web application server on which Unica Platform is deployed so the changes are applied.

When you use `-d` with the `-vp` command, the configTool deletes any child nodes in the path you specify if those nodes are not included in the XML file you specify.

`-r productName -f registrationFile`

Register the application. The registration file location can be relative to the `tools/bin` directory or can be a full path. By default, this command does not overwrite an existing configuration, but you can use the `-o` option to force an overwrite. The *productName* parameter must be one of those names that are listed above.

Note the following.

- When you use the `-r` command, the registration file must have `<application>` as the first tag in the XML.

  Other files can be provided with your product that you can use to insert configuration properties into the Unica Platform database. For these files, use the `-i` command. Only the file that has the `<application>` tag as the first tag can be used with the `-r` command.

- The registration file for the Unica Platform is named `Manager_config.xml`, and the first tag is `<Suite>`. To register this file on a new installation, use the `populateDb` utility, or rerun the Unica Platform installer as described in the *Unica Platform Installation Guide*.

- After the initial installation, to re-register products other than the Unica Platform, use `configTool` with the `-r` command and `-o` to overwrite the existing properties.

The `configTool` utility uses product names as parameters with the commands that register and unregister products. With the 8.5.0 release of Unica, many product names changed. However, the names that are recognized by `configTool` did not change. The valid product names for use with `configTool` are listed below, along with the current names of the products.

Table 27. Product names for configTool registration and unregistration

Product name	Name used in co nfigTool
Unica Platform	`Manager`
Unica Campaign	`Campaign`
Unica Collaborate	`Collaborate`
Unica Deliver	`Deliver`
Unica Journey	`Journey`
Unica Insights	`UnicaInsights`
Unica Content Integration	`assetPicker`
Unica Offer	`Offer`
Unica Interact	`interact`
Unica Optimize	`Optimize`
Unica Plan	`Plan`
Opportunity Detect	`Detect`
IBM SPSS Modeler Advantage Enterprise Marketing Management Edition	`SPSS`
Digital Analytics	`Coremetrics`

`-u` *productName*

Unregister an application that is specified by *productName*. You do not have to include a path to the product category; the product name is sufficient, and it is required. The process removes all properties and configuration settings for the product.

Options

`-o`

When used with `-i` or `-r`, it overwrites an existing category or product registration (node).

When used with `-d`, you can delete a category (node) that does not have the **Delete Category** link on the **Configuration** page.

Examples

- Import configuration settings from a file named `Product_config.xml` in the `conf` directory under the Unica Platform installation.

  ```
  configTool -i -p "Affinium" -f Product_config.xml
  ```

- Import one of the supplied Unica Campaign data source templates into the default Unica Campaign partition, partition1. The example assumes that you placed the Oracle data source template, `OracleTemplate.xml`, in the `tools/bin` directory under the Unica Platform installation.

  ```
  configTool -i -p "Affinium|Campaign|partitions|partition1|dataSources" -f OracleTemplate.xml
  ```

- Export all configuration settings to a file named `myConfig.xml` in the `D:\backups` directory.

  ```
  configTool -x -f D:\backups\myConfig.xml
  ```

- Export an existing Unica Campaign partition (complete with data source entries), save it to a file named `partitionTemplate.xml`, and store it in the default `tools/bin` directory under the Unica Platform installation.

  ```
  configTool -x -p "Affinium|Campaign|partitions|partition1" -f partitionTemplate.xml
  ```

- Manually register an application named productName, by using a file named `app_config.xml` in the default `tools/bin` directory under the Unica Platform installation, and force it to overwrite an existing registration of this application.

  ```
  configTool -r product Name -f app_config.xml -o
  ```

- Unregister an application named productName.

  ```
  configTool -u productName
  ```

- Run the following command to enable encodeCSV feature:

  ```
  configTool -vp -p "Affinium|Plan|umoConfiguration" -f Plan_Home\conf\Plan_encodeProperty_11.1.xml
  ```

- Register Unica Interact Settings as configuration menu under `AffiniumWebApps\Campaign\interact\conf\interact_setup_navigation.xml` using

  ```
  configTool.bat -v -i -p "Affinium|suite|uiNavigation|settingsMenu" -f "interact_setup_navigation.xml"
  ```

quartzjobtool

Scheduler jobs created in version 11.1 or older versions require to be updated to run on version 12.0. Use the quartzjobtool utility to update the scheduler jobs when the installer has not done this automatically during installation or upgrade. This tool reads environment variables from the `setenv_quartz` script. The Unica Platform installer should have set this variable automatically, but it is a good practice to verify that the `JAVA_HOME` variable is set if you have a problem running a utility. The JDK must be the Sun version (not, for example, the JRockit JDK available with WebLogic).

Syntax

quartzjobtool

Use the quartzjobtool to update scheduler jobs. This is a required step. If this upgrade tool is not run, any existing scheduled job will fail to start. The quartzjobtool is in the tools\bin directory under Unica Platform installation. Run this utility from the `tools\bin` directory.

Example command (Windows): quartzjobtool.bat

Example command (Unix): ./quartzjobtool.sh

Example

Update scheduler jobs quartzjobtool

datafilteringScriptTool

The `datafilteringScriptTool` utility reads an XML file to populate the data filtering tables in the Unica Platform system table database.

Depending on how you write the XML, you can use this utility in two ways.

- Using one set of XML elements, you can auto-generate data filters based on unique combinations of field values (one data filter for each unique combination).
- Using a slightly different set of XML elements, you can specify each data filter that the utility creates.

See *Unica Platform the Administrator's Guide* for information about creating the XML.

When to use datafilteringScriptTool

You must use `datafilteringScriptTool` when you create new data filters.

Prerequisites

The Unica Platform must be deployed and running.

Using datafilteringScriptTool with SSL

When the Unica Platform is deployed using one-way SSL you must modify the datafilteringScriptTool script to add the SSL options that perform handshaking. To modify the script, you must have the following information.

- Truststore file name and path
- Truststore password

In a text editor, open the datafilteringScriptTool script (`.bat` or `.sh`) and find the lines that look like this (examples are Windows™ version).

```
:callexec

"%JAVA_HOME%\bin\java" -DUNICA_PLATFORM_HOME="%UNICA_PLATFORM_HOME%"

com.unica.management.client.datafiltering.tool.DataFilteringScriptTool %*
```

Edit these lines to look like this (new text is in **bold**). Substitute your truststore path and file name and truststore password for `myTrustStore.jks` and `myPassword`.

```
:callexec

SET SSL_OPTIONS=-Djavax.net.ssl.keyStoreType="JKS"

-Djavax.net.ssl.trustStore="C:\security\myTrustStore.jks"

-Djavax.net.ssl.trustStorePassword=myPassword

"%JAVA_HOME%\bin\java" -DUNICA_PLATFORM_HOME="%UNICA_PLATFORM_HOME%" %SSL_OPTIONS%

com.unica.management.client.datafiltering.tool.DataFilteringScriptTool %*
```

Syntax

```
datafilteringScriptTool -r pathfile
```

Commands

`-r path_file`

Import data filter specifications from a specified XML file. If the file is not located in the `tools/bin` directory under your installation, provide a path and enclose the `path_file` parameter in double quotation marks.

Example

- Use a file named `collaborateDataFilters.xml`, located in the `C:\unica\xml` directory, to populate the data filter system tables.

  ```
  datafilteringScriptTool -r "C:\unica\xml\collaborateDataFilters.xml"
  ```

encryptPasswords

The `encryptPasswords` utility is used to encrypt and store either of two passwords that Unica Platform uses internally.

The two passwords that the utility can encrypt are as follows.

- The password that the Unica Platform uses to access its system tables. The utility replaces an existing encrypted password (stored in the `jdbc.properties` file, located in the `tools\bin` directory under your Unica Platform installation) with a new one.
- The keystore password used by the Unica Platform when it is configured to use SSL with a certificate other than the default one supplied with the Unica Platform or the web application server. The certificate can be either a self-signed certificate or a certificate from a certificate authority.

When to use encryptPasswords

Use `encryptPasswords` as for the following reasons.

- When you change the password of the account used to access your Unica Platform system table database.
- When you have created a self-signed certificate or have obtained one from a certificate authority.

Prerequisites

- Before running `encryptPasswords` to encrypt and store a new database password, make a backup copy of the `jdbc.properties` file, located in the `tools/bin` directory under your Unica Platform installation.
- Before running `encryptPasswords` to encrypt and store the keystore password, you must have created or obtained a digital certificate and know the keystore password.

Syntax

```
encryptPasswords -d databasePassword
```

```
encryptPasswords -k keystorePassword
```

Commands

-d *databasePassword*

Encrypt the database password.

-k *keystorePassword*

Encrypt the keystore password and store it in a file named `pfile`.

Examples

- When the Unica Platform was installed, the login for the system table database account was set to `myLogin`. Now, some time after installation, you have changed the password for this account to `newPassword`. Run `encryptPasswords` as follows to encrypt and store the database password.

```
encryptPasswords -d newPassword
```

- You are configuring an Unica application to use SSL and have created or obtained a digital certificate. Run `encryptPasswords` as follows to encrypt and store the keystore password.

```
encryptPasswords -k myPassword
```

encryptTomcatDBPasswords

The `encryptTomcatDBPasswords` utility is used to encrypt the database passwords that the Tomcat Application server uses internally. It is used to encrypt database passwords used in Campaign.xml and unica.xml. This utility can encrypt the Unica application database password. The utility prints the encrypted password in the command line.

When to use encryptTomcatDBPasswords

Use `encryptTomcatDBPasswords` utility when you want to use encrypted password under Tomcat configurations. It can be used when the Campaign or Unica System DB password has expired or changed. You can use this utility and encrypt the password and this will be get replaced in Campaign.xml, unica.xml and plan.xml located at `<instanceHome>\conf\Catalina\localhost`.

Syntax

```
encryptTomcatDBPasswords -d databasePassword
```

Commands

-d *databasePassword*

Encrypt the database password.

> **Note:**
>
> This utility is available only when the user selects Tomcat as the application server while installing Unica Platform.
>
> This utility can be used only in case when the user wants use encrypted passwords instead of plain text passwords, under Tomcat configurations.

For more details, see the Tomcat documentation.

partitionTool

Partitions are associated with Unica Campaign policies and roles. These policies and roles and their partition associations are stored in the Unica Platform system tables. The `partitionTool` utility seeds the Unica Platform system tables with basic policy and role information for partitions.

When to use partitionTool

For each partition you create, you must use `partitionTool` to seed the Unica Platform system tables with basic policy and role information.

See the installation guide appropriate for your version of Unica Campaign for detailed instructions on setting up multiple partitions in Unica Campaign.

Special characters and spaces

Any partition description or user, group, or partition name that contains spaces must be enclosed in double quotation marks.

Syntax

```
partitionTool -c -s sourcePartition -n newPartitionName [-u admin_user_name] [-d partitionDescription] [-g groupName]
[-a application]
```

Commands

The following commands are available in the `partitionTool` utility.

-c

Replicates (clones) the policies and roles for an existing partition specified using the `-s` option, and uses the name specified using the `-n` option. Both of these options are required with `c`. This command does the following.

- Creates a new Unica user with the Admin role in both the Administrative Roles policy and the global policy in Unica Campaign. The partition name you specify is automatically set as this user's password.
- Creates a new Unica Platform group and makes the new Admin user a member of that group.
- Creates a new partition object.
- Replicates all the policies associated with the source partition and associates them with the new partition.
- For each replicated policy, replicates all roles associated with the policy.
- For each replicated role, maps all functions in the same way that they were mapped in the source role.
- Assigns the new Unica Platform group to the last system-defined Admin role created during role replication. If you are cloning the default partition, partition1, this role is the default Administrative Role (Admin).

Options

-d *partitionDescription*

Optional, used with `-c` only. Specifies a description that appears in the output from the `-list` command. Must be 256 characters or less. Enclose in double quotation marks if the description contains spaces.

-a *application*

Optional, used with `-c`, `-n`, `-g`, and `-u` only . Clones data from the source partition for the specified application only partition. The application must be Unica Suite application.

-g *groupName*

Optional, used with `-c` only. Specifies the name of the Unica Platform Admin group that the utility creates. The name must be unique within this instance of Unica Platform

If not defined, the name defaults to `partition_nameAdminGroup`.

-n *partitionName*

Optional with `-list`, required with `-c`. Must be 32 characters or less.

When used with `-list`, specifies the partition whose information is listed.

When used with `-c`, specifies the name of the new partition, and the partition name you specify is used as the password for the Admin user. The partition name must match the name you gave the partition in when you configured it (using the partition template on the Configuration page).

-s *sourcePartition*

Required, used with `-c` only. The name of the source partition to be replicated.

-u *adminUserName*

Optional, used with `-c` only. Specifies the user name of the Admin user for the replicated partition. The name must be unique within this instance of Unica Platform.

If not defined, the name defaults to `partitionNameAdminUser`.

The partition name is automatically set as this user's password.

Examples

- Create a partition with the following characteristics.
  - Cloned from partition1
  - Partition name is `myPartition`
  - Uses the default user name (`myPartitionAdminUser`) and password (`myPartition`)
  - Uses the default group name (`myPartitionAdminGroup`)

- ◦ Description is "ClonedFromPartition1"
    - ° `partitionTool -c -s partition1 -n myPartition -d "ClonedFromPartition1"`
- Create a partition with the following characteristics.
    - ◦ Cloned from partition1
    - ◦ Partition name is `partition2`
    - ◦ Specifies user name of `customerA` with the automatically assigned password of `partition2`
    - ◦ Specifies group name of `customerAGroup`
    - ◦ Description is "PartitionForCustomerAGroup"
    - ° `partitionTool -c -s partition1 -n partition2 -u customerA -g customerAGroup -d`
       `"PartitionForCustomerAGroup"`
- Update a partition with the following characteristics.
    - ◦ Cloned from partition1
    - ◦ Partition name is partition2
    - ◦ Specifiy admin user name and admin user group of partition2
    - ° `partitionTool -c -s partition1 -n partition2 -u partition2AdminUser -a Journey`

Note: While using `-a` option ensure to specify the group name, if the group name was specified explicitly when partition was created by utility.

```
partitionTool -c -s partition1 -n partition2 -u partition2AdminUser -g [partition2 group name] -a Journey
```

populateDb

The `populateDb` utility inserts default (seed) data in the Unica Platform system tables.

The Unica installer can populate the Unica Platform system tables with default data for Unica Platform and for Unica Campaign. However, if your company policy does not permit the installer to change the database, or if the installer is unable to connect with the Unica Platform system tables, you must insert default data in the Unica Platform system tables using this utility.

For Unica Campaign, this data includes security roles and permissions for the default partition. For Unica Platform, this data includes default users and groups, and security roles and permissions for the default partition.

Syntax

```
populateDb -n productName
```

Commands

```
-n productName
```

Insert default data into the Unica Platform system tables. Valid product names are `Manager` (for Unica Platform) and `Campaign` (for Unica Campaign).

Examples

- Insert Unica Platform default data manually.

```
populateDb -n Manager
```

- Insert Unica Campaign default data manually.

```
populateDb -n Campaign
```

restoreAccess

The `restoreAccess` utility allows you to restore access to Unica Platform if all users with PlatformAdminRole privileges have been inadvertently locked out or if all ability to log in to the Unica Platform has been lost.

When to use restoreAccess

You might want to use `restoreAccess` under the two circumstances described in this section.

PlatformAdminRole users disabled

It is possible that all users with PlatformAdminRole privileges in Unica Platform might become disabled in the system. Here is an example of how the platform_admin user account might become disabled. Suppose you have only one user with PlatformAdminRole privileges (the platform_admin user). Assume the `Maximum failed login attempts allowed property` property in the **General | Password settings** category on the Configuration page is set to `3`. Then suppose someone who is attempting to log in as platform_admin enters an incorrect password three times in a row. These failed login attempts cause the platform_admin account to become disabled in the system.

In that case, you can use `restoreAccess` to add a user with PlatformAdminRole privileges to the Unica Platform system tables without accessing the web interface.

When you run `restoreAccess` in this way, the utility creates a user with the login name and password you specify, and with PlatformAdminRole privileges.

If the user login name you specify exists in Unica Platform as an internal user, that user's password is changed.

Only a user with the login name of PlatformAdmin and with PlatformAdminRole privileges can universally administer all dashboards. So if the platform_admin user is disabled and you create a user with `restoreAccess`, you should create a user with a login of platform_admin.

Improper configuration of NTLMv2 authentication

If you implement NTLMv2 authentication with improper configuration and can no longer log in, use `restoreAccess` to restore the ability to log in.

When you run `restoreAccess` in this way, the utility changes the value of the `Platform | Security | Login method` property to `Unica Platform`. This change allows you to log in with any user account that existed before you were locked out. You can

optionally specify a new login name and password as well. You must restart the web application server on which Unica Platform is deployed if you use the `restoreAccess` utility in this way.

Password considerations

Note the following about passwords when you use `restoreAccess`.

- The `restoreAccess` utility does not support blank passwords, and does not enforce password rules.

- If you specify a user name that is in use, the utility resets the password for that user.

Syntax

```
restoreAccess -u loginName -p password
```

```
restoreAccess -r
```

Commands

`-r`

When used without the `-u loginName` option, reset the value of the `Platform | Security | Login method` property to `Unica Platform`. Requires restart of the web application server to take effect.

When used with the `-u loginName` option, create a PlatformAdminRole user.

Options

`-u loginNname`

Create a user with PlatformAdminRole privileges with the specified login name. Must be used with the `-p` option.

`-p password`

Specify the password for the user being created. Required with `-u`.

Examples

- Create a user with PlatformAdminRole privileges. The login name is `tempUser` and the password is `tempPassword`.

  ```
  restoreAccess -u tempUser -p tempPassword
  ```

- Change the value of the login method to `Platform` and create a user with PlatformAdminRole privileges. The login name is `tempUser` and the password is `tempPassword`.

  ```
  restoreAccess -r -u tempUser -p tempPassword
  ```

scheduler_console_client

Jobs configured in the Unica Scheduler can be listed and kicked off by this utility if they are set up to listen for a trigger.

What to do if SSL is enabled

When the Unica Platform web application is configured to use SSL, the JVM used by the `scheduler_console_client` utility must use the same SSL certificate that is used by the web application server on which the Unica Platform is deployed.

Take the following steps to import the SSL certificate

- Determine the location of the JRE used by the `scheduler_console_client`.
    ◦ If `JAVA_HOME` is set as a system environment variable, the JRE it points to is the one used by the `scheduler_console_client` utility.
    ◦ If `JAVA_HOME` is not set as a system environment variable, the `scheduler_console_client` utility uses the JRE set either in the `setenv` script located in the `tools/bin` directory of your Unica Platform installation or on the command line.
- Import the SSL certificate used by the web application server on which the Unica Platform is deployed to the JRE used by `scheduler_console_client`.

    The Sun JDK includes a program called `keytool` that you can use to import the certificate. Consult the Java™ documentation for complete details on using this program or access the help by entering `-help` when you run the program.

    > **Note:** In case of upgrades, JRE shipped with Unica is overwritten so ensure that you reimport certificates in the JRE if you are using the same JRE.

- Open the `tools/bin/schedulerconsoleclient` file in a text editor and add the following properties. These differ depending on the web application server on which Unica Platform is deployed.
    ◦ For WebSphere®, add these properties to the file.

    -Djavax.net.ssl.keyStoreType=JKS

    -Djavax.net.ssl.keyStore="*Path to your key store JKS file*"

    -Djavax.net.ssl.keyStorePassword="*Your key store password*"

    -Djavax.net.ssl.trustStore="*Path to your trust store JKS file*"

    -Djavax.net.ssl.trustStorePassword="*Your trust store password*"

    -DisUseIBMSSLSocketFactory=false
    ◦ For WebLogic, add these properties to the file.

    -Djavax.net.ssl.keyStoreType="JKS"

    -Djavax.net.ssl.trustStore="*Path to your trust store JKS file*"

    -Djavax.net.ssl.trustStorePassword="*Your trust store password*"

If the certificates do not match, the Unica Platform log file contains an error such as the following.

```
Caused by: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to
requested target
```

Prerequisites

The Unica Platform must be installed, deployed, and running.

Syntax

```
scheduler_console_client -v -t trigger_name user_name
```

```
scheduler_console_client -s -t trigger_name user_name
```

Commands

`-v`

List the scheduler jobs configured to listen for the specified trigger.

Must be used with the `-t` option.

`-s`

Send the specified trigger.

Must be used with the `-t` option.

Options

`-t trigger_name`

The name of the trigger, as configured in the scheduler.

Example

- List jobs configured to listen for a trigger named `trigger1`.

  ```
  scheduler_console_client -v -t trigger1 myLogin
  ```
- Execute jobs configured to listen for a trigger named `trigger1`.

  ```
  scheduler_console_client -s -t trigger1 myLogin
  ```

Configuring cipher list in Unica Campaign

Prerequisite: Unica Campaign must be configured with SSL.

If Unica Campaign application and Listener are configured with SSL options as `TRUE`, then by default 98 ciphers are supported to enable the SSL communication between Unica Campaign application(Server) and listener.

To disallow weak ciphers from this default cipher list, users can use `<SSLCipherList>` tag or property in config.xml file.

To remove support of weak ciphers, users must add the following line in config.xml file. It specifies that support to default ciphers excludes `AES256-SHA, CAMELLIA256-SHA, AES128-SHA, SEED-SHA, CAMELLIA128-SHA, DES-CBC3-SHA, IDEA-CBC-SHA`.

```
<property name="SSLCipherList"><value>DEFAULT:!AES256-SHA:!CAMELLIA256-SHA:!AES128-SHA:!SEED-SHA:!CAMELLIA128-SHA:!
DES-CBC3-SHA:!IDEA-CBC-SHA</value></property>
```

This disables the above-mentioned ciphers, which are included in `<SSLCipherList>` tag of config.xml file.

If clients or users do not mention the SSLCipherList tag in config.xml file, then the default cipher list is considered and 98 ciphers are supported.

> ✏️ **Note:** The listener will not start and the following errors are generated in unica_aclsnr.log file, if users or clients disable any cipher which is required by certificate or browser.

```
Error enabling SSL connection.
```

```
SOCKET BIND port=4664: ERRNO 10048: Unknown error
```

Unica 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 Unica Deliver installation. The Unica Deliver directory is a sub-directory in the Unica 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.

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.

  ```
  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 Unica 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
```

Chapter 12. Uninstalling Unica Campaign and Unica Optimize

Run the Unica Campaign and Unica Optimize uninstaller to uninstall Unica Campaign and Unica Optimize. When you run the uninstaller, the files that were created during the installation process are removed. For example, files such as configuration files, installer registry information, and user data are removed from the computer.

About this task

When you install Unica products, an uninstaller is included in the `Uninstall_Product` directory, where *Product* is the name of your product. On Windows™, an entry is also added to the **Add or Remove Programs** list in the Control Panel.

If you manually remove the files in your installation directory instead of running the uninstaller, the result might be an incomplete installation if you later reinstall a product in the same location. After uninstalling a product, its database is not removed. The uninstaller only removes default files that are created during installation. Any file that is created or generated after installation is not removed.

In addition to the general instructions for uninstalling Unica products, follow the guidelines when you uninstall Campaign :

- If you have multiple Unica Campaign runtime installations using the same Unica Platform installation, you must remove network connectivity for the Unica Campaign runtime workstation before you run the uninstaller. Failure to do so uninstalls the configuration data for all the other Unica Campaign runtime installations from Unica Platform.
- You can safely ignore any warnings about failure to unregister with Unica Platform.
- You can export a copy of your configuration as a precaution before uninstalling Unica Campaign.
- If you uninstall the Unica Campaign design time environment, after you run the uninstaller, you may need to manually unregister Unica Campaign. Use the configtool utility to unregister `full_path_to_Interact_DT_installation_directory\interactDT\conf\interact_navigation.xml` and `full_path_to_Interact_RT_installation_directory\conf\interact_navigation.xml`.

Note: On UNIX™, the same user account that installed Unica Campaign must run the uninstaller.

1. If you have deployed the Unica Campaign web application, undeploy the web application from WebSphere® or WebLogic.
2. Stop the Unica Campaign listener.
3. To drop the Unica Optimize tables from the system table database, run the aco_systab_drop.sql script in the `ddl` directory under your Unica Campaign installation. If the `ddl` directory exists in the product installation directory, run the scripts that are provided in the `ddl` directory to drop tables from the system table database.
4. Shut down WebSphere® or WebLogic.
5. Stop the processes that are related to Campaign.
6. If the `ddl` directory exists in the product installation directory, run the scripts that are provided in the `ddl` directory to drop tables from the system table database.
7. Make sure that the IBM SPSS® Modeler Collaboration and Deployment Service (C&DS) is uninstalled.

Open the Installation Manager and verify that C&DS is uninstalled. If C&DS is not uninstalled, use the Installation Manager to uninstall C&DS.

8. Complete one of the following steps to uninstall Unica Campaign:

Choose from:

◦ Click the Unica Campaign uninstaller that exists in the `Uninstall_Product` directory. The uninstaller runs in the mode in which you installed Unica Campaign and Attribution Modeler.

◦ In a command-line window, navigate to the directory where the uninstaller exists, and run the following command to uninstall Unica Campaign and Attribution Modeler by using the console mode:

Uninstall**_Product** -i console

◦ If you are uninstalling on a UNIX™ environment, navigate to the directory where the uninstaller exists, and run the following command to uninstall Unica Campaign:

./Uninstall\ Interaction\ History.

> **Note:** UNIX™ requires spaces after `\`.

◦ In a command-line window, navigate to the directory where the uninstaller exists, and run the following command to uninstall Unica Campaign and Attribution Modeler by using the silent mode:

Uninstall**_Product** -i silent

When you uninstall Unica Campaign by using the silent mode, the uninstallation process does not present any dialogs for user interaction.

> **Note:** If you do not specify an option for uninstalling Unica Campaign, the uninstaller runs in the mode in which Unica Campaign is installed.

Chapter 13. Unica Campaign web application clustering

You can configure the Unica Campaign web application (J2EE application) in a cluster by completing additional tasks during the Unica Campaign installation.

You can integrate a Unica Campaign clustered web application environment with Interact, Plan and Deliver.

Overview

The Unica Campaign web application server components support load balancing when the web application is deployed in a clustered environment.

In Unica Campaign, certain objects, such as offers, are cached in the web application server to improve response times in the user interface. In a load balanced environment, offers created on one web application server are available to users whose sessions connect through another server if you have configured `campaign_ehcache.xml`.

The objects that are cached include: offers, offer templates and attributes, campaigns, sessions, folders, attributes, initiatives, and segments.

A clustered configuration can be used to minimize disruptions in application availability in the event of a web application server failure. Because the Unica Campaign system tables are stored in an external database, it is not necessary to replicate data to the standby server. If the primary web application server fails, the Unica Campaign web application must be started on the standby server. When started, the Unica Campaign web application retrieves all current data from the Unica Campaign system table database.

Main steps

To install Unica Campaign in a clustered web application environment, follow the instructions from chapters 2 through 8 and supplement those instructions with information from this chapter.

When you install Unica Campaign in a cluster, there are many ways to configure the installation. The following steps describe the basic process:

1. Run the installers on one system, typically, the administration server or whatever the equivalent is for your application server type.
2. Create and share a file directory to hold the upload files for all of the Unica Campaign installations.
3. Create the EAR or WAR file and deploy it on the cluster and web server.
4. Configure each system to share the Unica Platform system tables and the Unica Campaign system tables.
5. Configure each system to use the shared file directory.
6. Determine which server in the cluster will send notifications. Then, suppress the notification process on all of the other servers.
7. Configure `campaign_ehcache.xml` for distributed caching of templates and offer folders, as explained in the *Unica Campaign Installation Guide.*. To specify how often the cache is refreshed, you can adjust the Unica Campaign

cache configuration settings, such as `offerTemplateDataTTLSeconds`. However, this adjustment can adversely affect performance because smaller values result in more frequent cache updates. Configuration settings are described in the *Unica Campaign Administrator's Guide.*

WebSphere clustering guidelines

If you are installing Unica Campaign on a cluster in WebSphere®, complete these extra steps in addition to the steps for installing Unica Campaign on WebSphere®.

Preparing the data sources

In addition to any other instructions provided in this guide, complete the following tasks for your data sources:

- The Unica Campaign database must be on a server that is accessible to all servers in the cluster, but it does not need to be on a server in the cluster.
- When you configure the JDBC provider, specify the cluster as the scope.

Installing the products

When you run the installer, be sure that you install Unica Platform and Unica Campaign only once, on a server that is designated as the administration server for the cluster. The administration server is accessible to all servers in the Unica Campaign cluster.

You do not need to install the software on each cluster member. Instead, you install it on the administration server, create the EAR or WAR file, and then deploy the EAR or WAR file on each cluster member.

> 📝 **Note:** The IBM® WebSphere® application server that is bundled with Campaign does not support clustering. In a clustered environment, use IBM® WebSphere® Application Server Network Deployment.

Pre-deployment steps

Before you deploy Unica Campaign, complete the following task:

- Share the top-level directory in which Unica Campaign is installed. For example, if you installed Unica Campaign in `C:\CampaignCluster\UNICA\Campaign`, you would share the entire `CampaignCluster` directory.

Deployment steps

In addition to the instructions in the deployment chapter (), complete the following tasks:

1. Map modules to servers. When you set the options in the **Select Installation Options** wizard for WebSphere®, select the cluster and the web server when mapping modules to servers.
2. Extra instructions for the Generic JVM property: Configure the Generic JVM property on each server in the cluster.

   The path that you specify in the `<CAMPAIGN_HOME>` and other properties must point to the shared installation directory.

Set the following extra parameters for clusters:

◦ Set the following parameter to ensure that Deliver ETL is not triggered by all clustered nodes:

 -Dcampaign.deliveretl.disabled=true

 For the one node that should trigger ETL Unica Deliver data for all other nodes, set the parameter to **false**.

◦ Set the following parameter to ensure that Unica Interact ETL is not triggered by all clustered nodes:

 -Dcampaign.interactetl.disabled=true

 For the one node that should trigger ETL Interact data for all other nodes, set the parameter to false.

Property to enable/disable Journey CH RH in campaign.

Property name: **campaign.journeyetl.disabled**

Supported values: true or false

Default value: false

◦ **-Dcampaign.journeyetl.disabled=false**

 If system property is not present or is set to false, campaign will read CH RH messages from Kafka and dump them in CH RH tables.

◦ **-Dcampaign.journeyetl.disabled=true**

 If system property is set to true, campaign will not read the CH RH data from Kafka and no data will be dumped in Campaign Database.

Post-deployment steps

If you are using a plug-in for load balancing, complete the following configuration steps:

• For Unica Campaign to work effectively in a clustered environment, users must remain on a single node for their entire session. This option for session management and load balancing is known as session affinity. The documentation for your application server provides details about configuring your installation to use session affinity.

> ✏️ **Note:** When a node fails on a system that uses this configuration option, every user session on that node also fails. The load balancer must not switch users to another available node because user authentication applies only to a single node in Unica Campaign. Users are prompted to log in again. In some cases, an unexpected error and corresponding data loss can occur.

• Log in to Unica Campaign. Select **Settings > Configuration**, and configure the following URL parameters so that all references to the Unica Campaign server use the proxy host and port:

Campaign | navigation | serverURL

Jboss clustering guidelines

For Installing Unica Campaign on Jboss cluster refer (Unica Campaign web application clustering on page 163).

Preparing the data sources

You will need to add/create JNDI in the same JBOSS domain which you are planning to use in cluster. Please follow below steps to create JNDI.

- Add required database modules in JBOSS. For example, oracle database -

```
jboss-cli.bat --connect --controller=JBOSS_DOMAIN_MASTER:MASTER_PORT --
commands="module add --name= oracle.jdbc --resources=<Database_Driver_Path\ojdbc8.jar> --
dependencies=javax.api,javax.transaction.api"
```

- JNDI's can be added by modifying domain.xml of both master and slave.

Deployment steps

In addition to the instructions in the deployment chapter (Deploying Unica Campaign on JBoss Application Server on page 106), complete the following tasks:

1. Launch JBOSS management console.
2. Navigate to "Deployments" tab. Select the ServerGroup.
3. Click on Add button under Deployments section
4. A pop up will be shown, browse the application WAR/EAR and complete the deployment process.

Tomcat Clustering Guidelines

For Installing Unica Campaign on Tomcat cluster refer (Unica Campaign web application clustering on page 163).

📝 **Note:** EAR file is not supported for Tomcat cluster.

For Deployment Steps, you can refer (Deploying Unica Campaign on Tomcat Application Server on page 107)

Post deployment, JNDI name should be prefixed with "java:/comp/env/" for example: campaignPartition1DS should be mentioned as "java:/comp/env/campaignPartition1DS" while creating Unica Campaign UA_SYSTEM_TABLE Data Source.

Configure a load Balancer like nginx for redirecting request to all the nodes of Tomcat cluster.

Tomcat clustering guidelines

Tomcat Cluster with nginx (Load Balancer)

Nginx(pronounced "engine X") is an HTTP and reverse proxy server. It is well known for its high performance and stability.

It is pretty feature-rich and very simple to configure. Nginx hosts nearly 12.18 percent (22.2M) of active sites across all domains.

Nginx uses event-driven architecture to handle requests. When compared to a thread-per-request model, event-driven is highly scalable with a low and predictable memory footprint.

Pre-Requisites

- Tomcat server (server1, server2, sever3,…,etc.)
- Tomcat 1: HCL Unica Platform and Campaign (web & listener)
- Tomcat 2: HCL Unica Platform
- Tomcat 3: HCL Unica Platform
- Nginx load balancer (server4)
- Client browser to access Unica application

Architecture
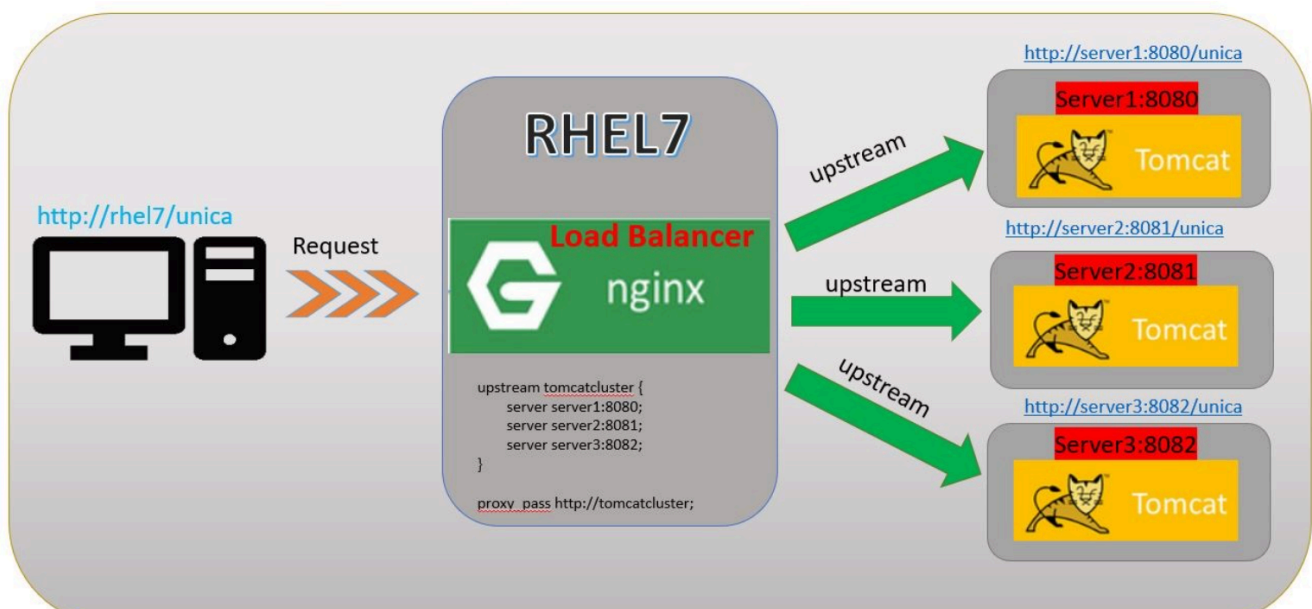
Steps to Follow

1. **Tomcat Installation - server1**

   Download and install Tomcat server and listen on 8080

2. **Tomcat Installation - server2**

   Download and install Tomcat server and listen on 8081

3. **Tomcat Installation - server3**

   Download and install Tomcat server and listen on 8082

4. **Tomcat server1:8080**

   Install and deploy both HCL Unica Platform and Campaign and make it work with 8080 port.

5. **Tomcat server2:8081**

   Install and deploy HCL Unica Platform only and make it work with 8081 port.

6. **Tomcat server3:8082**

   Install and deploy HCL Unica Platform only and make it work with 8082 port.

Note: Another way to implement clustering for unica is to have shared mount point across all the clusters and perform the installation at the shared point. This way the installation will be done just once, but will be shared across all the clusters. With this way the file system including war files and log files will be shared among the participating clusters.

Nginx installation

Download nginx & Install:

1. wget http://nginx.org/download/nginx-1.2.6.tar.gz
2. tar -xzf **nginx-1.2.6.tar.gz**
3. cd nginx-1.2.6/
4. ./configure --prefix=/opt/Tomcat/nginx (Ex:- `./configure --prefix=/opt/Tomcat/nginx --without-http_gzip_module --with-http_stub_status_module`)
5. Make
6. Make install

Nginx Configuration

After completion of installation, to configure nginx:

1. Update `<NGINX_HOME>/conf/ nginx.conf`
2. To start and stop nginx

   `<NGINX_HOME>/sbin/ngin`

   `<NGINX_HOME>/sbin/nginx -s stop`

3. Nginx Logs directory: `<NGINX_HOME>/logs/`

Test nginx

In this example, nginx is listening on 80 port, hence URL is http://rhel7:80. With the above screen, can confirm that nginx is running without any issues.

Nginx Load Balancer Algorithams

1. **Round-robin**

   requests to the application servers are distributed in a round-robin fashion.

   Here is the sample configuration (<NGINX_HOME>/conf/ nginx.conf) with round robin fashion

2. **Ip_hash**

   With ip-hash, the client's IP address is used as a hashing key to determine what server in a server group should be selected for the client's requests.

   This method ensures that the requests from the same client will always be directed to the same server except when this server is unavailable.

   Here is the sample configuration (<NGINX_HOME>/conf/nginx.conf) with round robin fashion

3. **Least_conn**

   With the least-connected load balancing, nginx will try not to overload a busy application server with excessive requests, distributing the new requests to a less busy server instead.

4. **Weight Distribution (weigth n)**

   With this configuration, every 5 new requests will be distributed across the application instances as the following: 3 requests will be directed to srv1, one request will go to srv2, and another one — to srv3.

   Ex:-

   ```
   Upstream tomcatcluster
    {
            Server server1:8080 weight 3;
            Server server2:8081;
            Server server4:8082;
      }
   ```

5. **Max Fails configuration**

   Max fails refers to the maximum number of failed attempts to connect to a server should occur before it is considered inactive. Fall_timeout specifies the length of that the server is considered inoperative. Once the time expires, new attempts to reach the server will start up again. The default timeout value is 10 seconds.

   Ex:-

   ```
   Upstream tomcatcluster
        {
                Server server1:8080 max_fails=3 fail_timeout=15s;
                Server server2:8081;
                Server server4:8082;
          }
   ```

Unica configuration with nginx:

Change the Platform navigation URL with nginx server host and actual context.

Configure `ehcache`

The `ehcache` is an open source Java™ distributed cache for caching, Java™ EE, and light-weight containers. Every node in the cluster can use the same `campaign_ehcache.xml` file or you can set a different `campaign_ehcache.xml` file for each node. For installations in a cluster, edit the `campaign_ehcache.xml` file so you do not have to restart the computer if you change any templates or offer folders.

> **! Important:**
>
> - If your installation is upgraded from a previous version, the `campaign_ehcache.xml` file is overwritten. Add and edit the following sections in the file.
> - You must also configure caching for user sessions (`userSessionCache`).

Use one of the following procedures to configure the `ehcache` file.

Configuring `ehcache` with Remote Method Invocation (RMI)

Unica Campaign systems that have the following topography typically use RMI:

Go to the `<HCL_Unica_HOME>\<CAMPAIGN_HOME>\conf` directory and open the `campaign_ehcache.xml` file in a text editor. Then, make the following edits:

- Uncomment the following section of the file.

  You must customize the settings for machineA and machineB to reflect your environment. Provide a pipe-separated list of all servers in the cluster and use fully qualified host names.

  ```
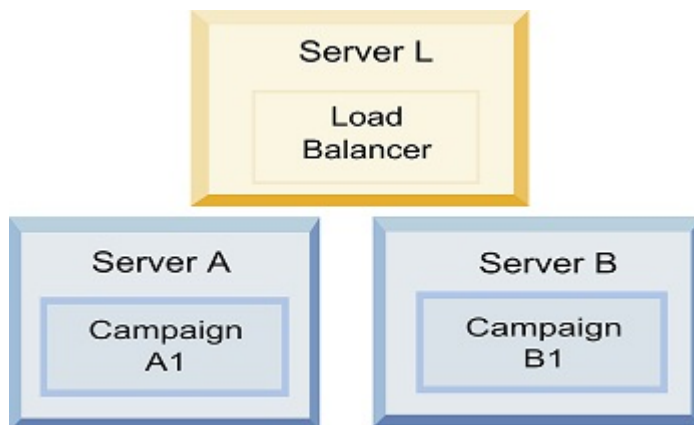  <!--<cacheManagerPeerProviderFactory
  class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"
  properties="peerDiscovery=manual, rmiUrls=//
  <machineA>:40000/campaignApplicationCache|//
  <machineB>:40000/campaignApplicationCache|//
  <machineA>:40000/userSessionCache|//
  <machineB>:40000/userSessionCache"/>
  ```

```
<machineA>:40000/mappingCache
<machineA>:40000/multipartitionOptSessionRunStatusCache
<machineA>:40000/optSessionUserSessionCache
-->
```

• Uncomment the following section of the file.

```
<!--
<cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
properties="replicateAsynchronously=true, replicatePuts=true,
replicateUpdates=true, replicateUpdatesViaCopy=true,
replicateRemovals=true"/>
<cacheEventListenerFactory
class="com.unicacorp.Campaign.cache.CampaignCacheEventListenerFactory" />
```

```
<machineA>:40000/mappingCache
<machineA>:40000/multipartitionOptSessionRunStatusCache
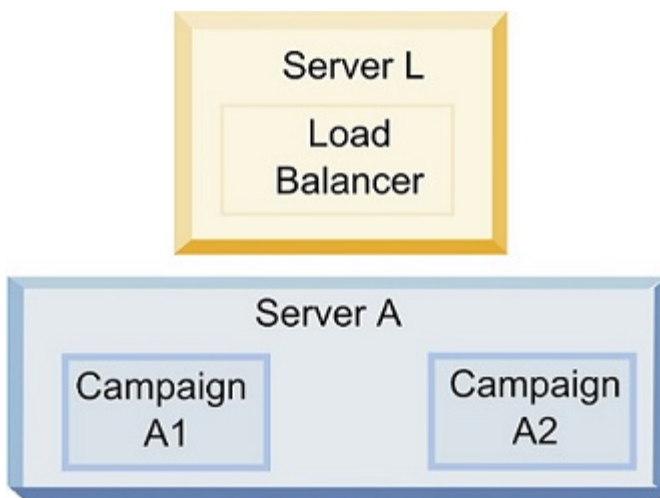<machineA>:40000/optSessionUserSessionCache


-->
```

• Uncomment the following section of the file.

```
<!--
<cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
properties="replicateAsynchronously=true, replicatePuts=true,
replicateUpdates=true, replicateUpdatesViaCopy=true,
replicateRemovals=true"/>
-->
```

Configuring `ehcache` with multicast

Unica Campaign systems that have the following topography typically use multicast:

Go to the `<UNICA_HOME>\<CAMPAIGN_HOME>\conf` directory and open the `campaign_ehcache.xml` file in a text editor. Then, make the following edits:

- Uncomment the following section of the file.

  You must customize the settings for multicastGroupAddress and multicastGroupPort to reflect your environment.

  ```
  <!--<cacheManagerPeerProviderFactory
  class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"
  properties="peerDiscovery=automatic, multicastGroupAddress=230.0.0.1,
  multicastGroupPort=4446, timeToLive=32"/>

  <cacheManagerPeerListenerFactory
  class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"/>
  -->
  ```

- Uncomment the following section of the file.

  ```
  <!--
  <cacheEventListenerFactory
  class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
  properties="replicateAsynchronously=true, replicatePuts=true,
  replicateUpdates=true, replicateUpdatesViaCopy=true,
  replicateRemovals=true"/>
  <cacheEventListenerFactory
  class="com.unicacorp.Campaign.cache.CampaignCacheEventListenerFactory" />
  -->
  ```

Chapter 14. Installing a clustered listener environment

Follow these instructions if you want to install a clustered listener configuration. A listener cluster is a set of two or more listeners that act as a unit to provide load balancing and high availability. Unica Campaign listener clusters are Active-Active; meaning each node serves requests using the load-balanced approach. Each Unica Campaign listener provides an interface between front-end clients (such as the Unica Campaign web application) and back-end analytical server processes.

1. Confirm that you have met the prerequisites listed separately in Supported listener clustering configurations on page 176. For example, it is important that a shared file location is already configured and that you have prepared separate machines for each node in the cluster.

2. Follow the instructions below to install a clustered listener configuration.

Step	Description
A. Start the installer.	Go to the folder where you saved the Unica installer and run the installer. This invokes all of the product installers that are in that location (Unica Platform, Unica Campaign).
B. Install Unica Platform if necessary.	If you have not already done so, complete the screens that install Platform, and click **Done** on the **Installation Complete** window. If you need help, see the "Install Platform using Console mode" topic in the Platform Installation Guide.
C. Install Unica Campaign in a clustered listener configuration, optionally including the first listener.	The Unica Campaign installer opens. In this installer, you must configure Unica Campaign for a clustered configuration and, optionally, install the first listener in the cluster as well. One listener in the cluster may be installed on the Unica Campaign web application server; however, each subsequent listener must be installed on its own standalone server. To install Unica Campaign in a clustered listener configuration, complete the following screens: ◦ Introduction ◦ Software License Agreement ◦ Installation Directory ◦ Components: Select the desired options and be sure to select Unica Campaign Server, which is the option to install the listener. ◦ Single or Multiple Listeners: Select **Clustered listener configuration (two or more nodes)** ◦ Listener Node Properties: Select **Run Campaign** or **Run Optimize** or both to install Campaign Analytical Server to serve Campaign or Optimize or both. ◦ Complete the following screens to install the first listener: ▪ **Shared Network File Location.** The clustered listener configuration requires that certain files and folders are shared and accessible to all of the listeners in a cluster. Use this screen to specify the path to the shared file location. Use a mapped Microsoft™ Windows™ server path (such as `Z:\Campaign_Shared`) or a

Step	Description
	mounted UNIX™ path (such as `/mnt/Campaign_Shared`). This location is known as the campaignSharedHome. ▪ **Listener Node Properties.** For each listener node that you install in a cluster, you must specify properties such as a unique name for the node, the node's network host and port, and so on. ▪ **Master Listener Priority.** The priority determines which node in the listener cluster is the master listener and which node to use in the event of failover. ▪ **Load Balancing Weight.** The weight determines how much listener traffic the node can support to share processing with the other nodes. Specify any value other than zero, because a value of zero prevents the node from handling any listener connections. From this point in the installation process, the remaining screens are essentially the same as for a single-node install. When you complete the Pre-installation Summary screen, click **Install** to complete the installation of Unica Campaign and the first listener node in the cluster. The Unica Campaign installer runs with the specified options.
D. Deploy and package the EAR file	For the first installer node in the cluster, which you might be installing on the Unica Campaign web application server, the installer displays screens to deploy and package the EAR file, the same as for a single-node installation. Continue with the process of deploying and running Unica Campaign in your web application server, and running the listener on the Unica Campaign server, as described in Configuring Unica Campaign and Unica Optimize before deployment on page 87.
E. Install the second node in the cluster.	If you have not already done so, copy the Unica master installer and the Unica Campaign installation files onto the server where your next Unica Campaign in-

Step	Description
⚠️ **Important: Each listener node must be installed on a separate machine.**	staller node will run, and launch the Unica installer. Complete the screens as described here: ◦ In the Unica installer, provide the information necessary to connect to the Unica Platform database, exactly the same way as for the first listener you installed. Each listener in the same cluster must use the same Unica installer configuration. When the Unica Campaign installer appears, complete the screens as described here: ◦ Introduction ◦ Software License Agreement ◦ Installation Directory ◦ Campaign Components: Select only **Campaign Server**, because you are only installing the listener on this system. ◦ Single or Multiple Listeners: Select **Clustered listener configuration (two or more nodes)** ◦ Complete the following screens to install the second listener: ▪ **Shared Network File Location.** The clustered listener configuration requires that certain files and folders are shared and accessible to all of the listeners in a cluster. Use this screen to specify the path to the shared file location. Use a mapped Microsoft™ Windows™ server path (such as `Z:\Campaign_Shared`) or a mounted UNIX™ path (such as `/mnt/Campaign_Shared`). This location is known as the campaignSharedHome. 📝 **Note:** The value that you enter here must be the same for every listener in the cluster. ▪ **Listener Node Properties.** For each node that you install in a cluster, you must specify properties such as a unique name for the node, the node's network host and port, and so on. ▪ **Master Listener Priority.** The priority determines which node in the listener cluster is the master listener, and which node to use in the event of failover. ▪ **Load Balancing Weight.** The weight determines how much listener traffic the node can support for sharing the processing with the other nodes. Specify any value other than zero, because a value of zero prevents the node from handling any listener connections. From this point in the installation process, the remaining screens are essentially the same as for a single-node install. When you complete the Pre-installation Summary screen, click **Install** to complete the installation of Unica Campaign and the first listener node in the cluster. The Unica Campaign installer runs with the options that you specified.

175

Step	Description
G. Install each subsequent node in the cluster.	Repeat the installation steps that you performed for the second listener node for each additional node you want to install. Remember that each node must be installed on a separate system from the other nodes. Restart the web application server and the listener on each node as you complete its installation.
H. Optionally, adjust configuration settings.	Log in to Unica Campaign to view or modify any of the configuration settings. See the *Unica Campaign Administrator's Guide* for instructions.

Results

You have now installed the Unica Campaign clustered listeners.

Supported listener clustering configurations

This topic pertains to a clustered listener configuration.

Prerequisites and requirements for Unica Campaign listener cluster configurations:

- There can be only one listener per physical host machine.
- All back-end machines for clustered listeners must be running the same operating system type.
- All back-end machines for clustered listeners must have the same version of Unica Campaign installed.
- The shared network location (campaignSharedHome) must be in place and must be accessible from each physical host machine on which you plan to install a listener node. You must set this up prior to installing the listener nodes.
- listenerType --Campaign Listener when installed in cluster mode, ListenerType plays an important role to decide which listener should execute Campaign flowcharts or Optimize session or both.
    - ListenerType - 1 - Run only Campaign flowcharts
    - ListenerType - 2 - Run only Optimize sessions
    - ListenerType - 3 - Run Campaign flowchart and Optimize session

These listenertypes are set based on the option selected while installing Campaign listener. Users can switch between these values at any time and restart the application to reflect this.

Diagram of listener clustering

This diagram illustrates a three-node listener cluster configuration.

✏️ **Note:** The following summary provides a high-level overview of the components. Details are provided in separate topics.

A cluster consists of multiple listener nodes. Each node (unica_aclsnr) is on a separate physical machine, and each node has a native ODBC connection to the Unica Campaign system database. As with a single-node configuration, each unica_aclsnr process spawns additional back-end processes for logins and flowcharts.

Each node also has a connection to a back-end user database (not shown in diagram).

In a clustered configuration, one node acts as the master listener. The job of the master listener is to perform load balancing by distributing incoming requests to each node. The Unica Campaign web application sends client requests over TCP/IP, and the load balancer component communicates with clustered nodes over TCP/IP. All nodes share a network file system, so they can access shared files. Additionally, each node has its own local tmp folder and its own set of files that are not shared.

Shared network location for clustered listeners: campaignSharedHome

A clustered listener configuration for Unica Campaign requires that certain files and folders must be shared and accessible to all listeners in a cluster. Therefore, you must have a shared file system in place.

Requirements

- The common area can be a machine or location that is accessible from all other machines in the listener cluster.
- Each listener in a cluster must have full access to the shared files and folders.
- Best practice is for all listeners to be on the same network, and for the shared home to also be on that network, to avoid latency issues.
- To avoid any single point of failure, the shared file system should use mirrored RAID or an equivalent method of redundancy.
- If you are installing a single-listener configuration, a shared file system is a best practice, in case you decide to implement listener clusters in the future.

Shared files and folders

In a clustered configuration, all listeners share the folder structure shown below. The shared location (*<campaignSharedHome>*) is specified at installation time and is configurable at **Campaign|campaignClustering|campaignSharedHome**. The shared partition contains all logs, campaigns, templates, and other files.

```
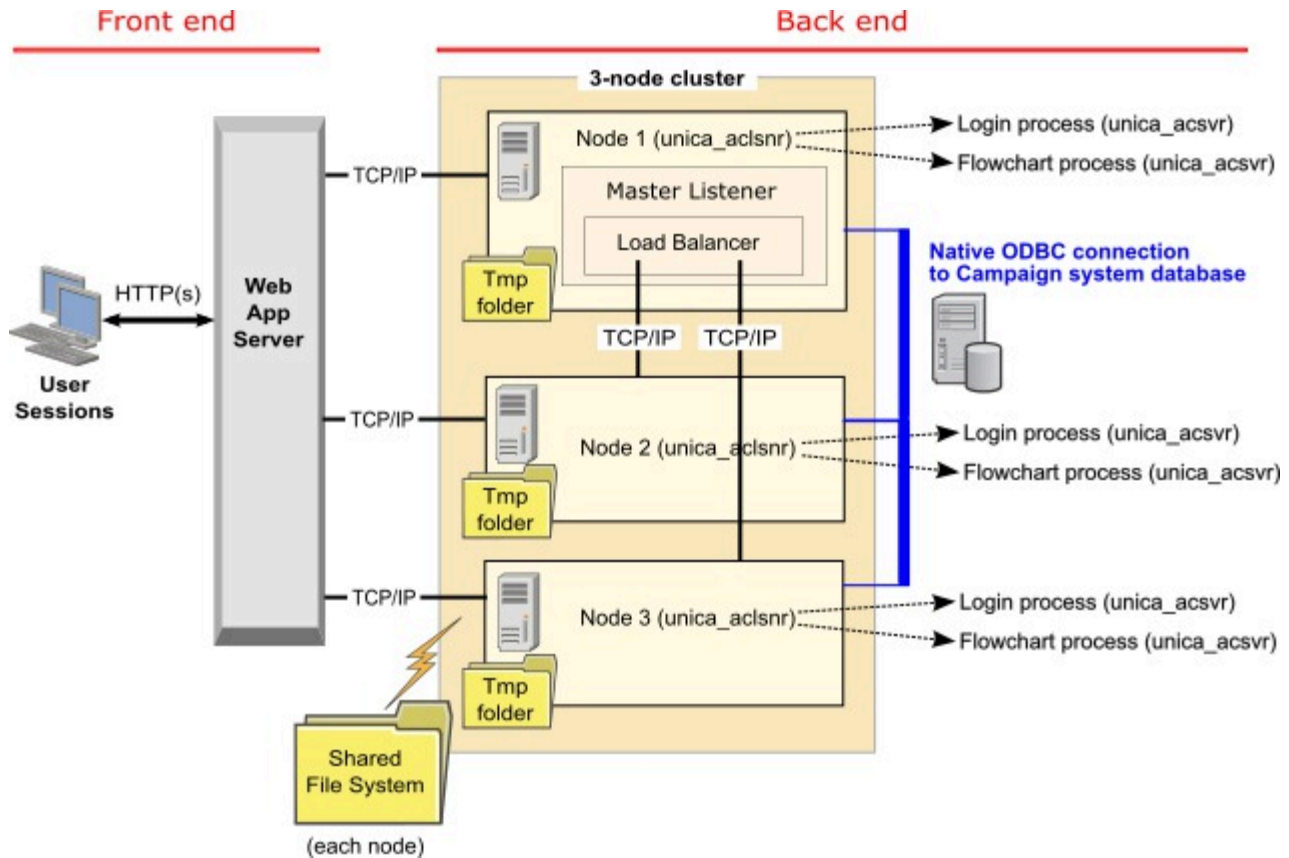campaignSharedHome
 |--->/conf
    |------> activeSessions.udb
    |------> deadSessions.udb
    |------> etc.
 |--->/logs
    |------> masterlistener.log
    |------> etc.
 |--->/partitions
    |------> partition[n]
       |----------> {similar to <Campaign_home> partition folder structure}
```

Files and folders that are not shared

Each Unica Campaign listener has its set of folders and files that are not shared, under *<Campaign_home>*. Campaign_home is an environment variable that represents the Unica Campaign application installation directory. This variable is set in cmpServer.bat (Windows™) or rc.unica_ac.sh (UNIX™). Partitions are specific to the local listener. Each local partition folder contains a `tmp` folder for temporary files during flowchart runs and a `conf` folder for the table manager cache file.

```
Campaign_home
 |--->/conf
    |------> config.xml
    |------> unica_aclsnr.pid
    |------> unica_aclsnr.udb
    |------> etc.
 |--->/logs
    |------> unica_aclsnr.log
    |------> etc.
 |--->/partitions
    |------> partition[n]
       |---------->/tmp
       |---------->/conf
       |---------->{other files specific to the partition}
```