

Unica Platform V12.1.9 Installation Guide



Contents

Chapter 1. Installation overview.....	4
Installation roadmap.....	4
How the installers work.....	6
Modes of installation.....	6
Sample response files.....	7
Unica Platform documentation and help.....	8
Chapter 2. Planning the Unica Platform installation.....	10
Prerequisites.....	10
Unica Platform installation worksheet.....	12
Installation order for Unica products.....	14
Chapter 3. Creating the Unica Platform data sources.....	16
Creating the JDBC connection in the web application server.....	17
Configuring the web application server for your JDBC driver.....	18
Information for creating JDBC connections.....	20
Chapter 4. Installing Unica Platform.....	27
Installing Unica Platform using the GUI mode.....	28
Creating an EAR file after running the installer.....	36
Installing Unica Platform using the console mode.....	37
Installing Unica Platform silently.....	38
Sample response files.....	39
Unica Platform components.....	40
Creating and populating the Unica Platform system tables manually.....	42
Chapter 5. Deploying Unica Platform.....	43
Configuration of JVM parameters for Unica products.....	43
Guidelines for deploying Unica Platform on WebLogic.....	51
Guidelines for deploying Unica Platform on WebSphere.....	51
Guidelines for deploying Unica Platform on JBoss.....	53
Guidelines for deploying Unica Platform on Apache Tomcat@.....	54
Google Chrome and Microsoft Edge configuration.....	56
Verifying your Unica Platform installation.....	57
Chapter 6. Deploying Unica Contact Central.....	58
Chapter 7. Deploying AI-ML.....	62
Deploying AI-ML in Microsoft Windows.....	62
Deploying AI-ML in Unix-based Operating Systems.....	67
Chapter 8. Configuring Unica Platform after deployment.....	69
Default password settings.....	69
Set the web application session timeout (optional).....	69
Additional configuration for enhanced security.....	70
Disabling the X-Powered-By flag.....	70
Configuring a restrictive cookie path.....	70
Disabling caching of Unica Campaign and Unica Platform configuration property values.....	71
Enabling single-user logging.....	71
Collecting installations logs.....	75
Chapter 9. Installing Unica Platform in a cluster.....	76
Creating a WebSphere profile for the primary node of a cluster.....	76
Creating profiles for additional nodes.....	77
Adding existing nodes to the cluster nodes.....	77
Creating a WebSphere cluster.....	78
Creating the web server in the cluster.....	78
Starting the nodes, web server, and cluster.....	78
Creating JDBC providers in a cluster.....	79
Installing Unica products in a cluster.....	79
Configuring the HTTP Server as a load balancer.....	79
Configure property to control the Scheduler in a clustered environment.....	80
Configuring Unica Platform logs in a cluster.....	80
Sharing the location where Unica Platform is installed.....	80
Setting Unica Platform home in a Java parameter.....	81
Installing Unica Platform in a Tomcat cluster.....	82
Tomcat cluster with Nginx (Load Balancer).....	82
Prerequisites.....	83
Architecture.....	83
Install Nginx.....	83
Configure Nginx.....	83
Verify Nginx.....	84
Nginx load balancer algorithms.....	84
Configure Unica with Nginx.....	87
Chapter 10. Unica Platform utilities.....	89
Setting up Unica Platform utilities on additional machines.....	91
Utilities.....	91
alertConfigTool.....	92
configTool.....	92
Clientdetails.....	97
datafilteringScriptTool.....	97
encryptPasswords.....	99
encryptTomcatDBPasswords.....	100
partitionTool.....	101
populateDb.....	103
restoreAccess.....	104
scheduler_console_client.....	105

quartzjobtool.....	107
Chapter 11. Unica Platform SQL scripts.....	109
ManagerSchema_DeleteAll.sql.....	109
ManagerSchema_PurgeDataFiltering.sql.....	109
SQL scripts for creating system tables.....	110
ManagerSchema_DropAll.sql.....	111
Chapter 12. Uninstalling Unica Platform.....	113
Index.....	

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.

Installation roadmap

Use the installation roadmap to quickly find the information that you need for installing Unica Platform.

You can use [Table 1: Unica Platform installation roadmap on page 4](#) to scan the tasks that must be completed for installing Unica Platform. The **Information** column in the following table provides links to the topics that describe the tasks for installing Unica Platform:

Table 1. Unica Platform installation roadmap

Topic	Information
Installation overview on page 4	This chapter provides the following information: <ul style="list-style-type: none">• How the installers work on page 6• Modes of installation• Unica Platform documentation and help on page 8
Planning the Unica Platform installation on page 10	This topic provides the following information: <ul style="list-style-type: none">• Prerequisites on page 10• Unica Platform installation worksheet on page 12• Installation order for Unica products on page 14
Creating the Unica Platform data sources on page 16	This topic provides the following information: <ul style="list-style-type: none">• Creating the JDBC connection in the web application server on page 17• Configuring the web application server for your JDBC driver on page 18
Installing Unica Platform on page 27	This topic provides the following information: <ul style="list-style-type: none">• Installing Unica Platform using the GUI mode on page 28• Installing Unica Platform using the console mode on page 37

Table 1. Unica Platform installation roadmap

(continued)

Topic	Information
Deploying Unica Platform on page 43	<ul style="list-style-type: none"> • Installing Unica Platform silently on page 38 • Unica Platform components on page 40 • Creating and populating the Unica Platform system tables manually on page 42
Configuring Unica Platform after deployment on page 69	<p>This topic provides the following information:</p> <ul style="list-style-type: none"> • Guidelines for deploying Unica Platform on WebLogic on page 51 • Guidelines for deploying Unica Platform on WebSphere on page 51 • Guidelines for deploying Unica Platform on Apache Tomcat on page 54 • Guidelines for deploying Unica Platform on JBoss on page 53 • Verifying your Unica Platform installation on page 57
Unica Platform utilities on page 89	<p>This topic provides the following information:</p> <ul style="list-style-type: none"> • Default password settings on page 69 <p>This topic provides the following information:</p> <ul style="list-style-type: none"> • Setting up Unica Platform utilities on additional machines on page 91 • alertConfigTool on page 92 • configTool on page 92 • Clientdetails on page 97 • datafilteringScriptTool on page 97 • encryptPasswords on page 99 • encryptTomcatDBPasswords on page 100 • partitionTool on page 101 • populateDb on page 103 • restoreAccess on page 104 • scheduler_console_client on page 105
Unica Platform SQL scripts on page 109	<p>This topic provides the following information:</p>

Table 1. Unica Platform installation roadmap

(continued)

Topic	Information
Uninstalling Unica Platform on page 113	This topic provides the steps to uninstall Platform.

- [ManagerSchema_DeleteAll.sql on page 109](#)
- [ManagerSchema_PurgeDataFiltering.sql on page 109](#)
- [SQL scripts for creating system tables on page 110](#)
- [ManagerSchema_DropAll.sql on page 111.](#)

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 Platform.

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 Platform by using the graphical user interface.



Note: To launch the installers on Windows 2019 and Windows 2022, server please set parameter - **SET JAVA_TOOL_OPTIONS="-Dos.name=Windows 7"** - on command prompt and then launch the Installers from the same command prompt.

UNIX™ X Window System mode

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

Console mode

Use the console mode to install Unica Platform 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 Platform 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.

Sample response files

You must create response files to set up a silent installation of Unica Platform. 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 2. Description of sample response files

Sample response file	Description
<code>installer.properties</code>	The sample response file for the Unica master installer.
<code>installer_product initials and product version number.properties</code>	The sample response file for the Unica Platform installer. For example, <code>installer_ucn.n.n.n.properties</code> is the response file of the Unica Campaign installer, where <code>n.n.n.n</code> is the version number. For example, <code>installer_umpn.n.n.n.properties</code> is the response file of the Platform installer, where <code>n.n.n.n</code> is the version number.

Table 2. Description of sample response files

(continued)

Sample response file	Description
	For example, <code>installer_uln.n.n.n.properties</code> is the response file of the Leads installer, where <i>n.n.n.n</i> is the version number.
<code>installer_report pack initials, product initials, and version number.properties</code>	The sample response file for the reports pack installer.
	For example, <code>installer_urpcn.n.n.n.properties</code> is the response file of the Unica Campaign reports pack installer, where <i>n.n.n.n</i> is the version number
	For example, <code>installer_urpl.properties</code> is the response file of the Leads reports pack installer.

Table 3. Description of sample response files

Sample response file	Description
<code>installer.properties</code>	The sample response file for the Unica master installer.
<code>installer_product initials and product version number.properties</code>	The sample response file for the Unica Platform installer.
	For example, <code>installer_ucn.n.n.n.properties</code> is the response file of the Unica Campaign installer, where <i>n.n.n.n</i> is the version number.
	For example, <code>installer_umpn.n.n.n.properties</code> is the response file of the Unica Platform installer, where <i>n.n.n.n</i> is the version number.
	For example, <code>installer_uln.n.n.n.properties</code> is the response file of the Leads installer, where <i>n.n.n.n</i> is the version number.

Unica Platform documentation and help

Unica Platform provides documentation and help for users, administrators, and developers.

Table 4. Get up and running

Task	Documentation
View a list of new features, known issues, and workarounds	<i>Unica Platform Release Notes®</i>

Table 4. Get up and running (continued)

Task	Documentation
Learn about the structure of the Unica Platform database	<i>Unica Platform System Tables Guide</i>
Install or upgrade Unica Platform and deploy the Unica Platform web application	One of the following guides: <ul style="list-style-type: none"> • <i>Unica Platform Installation Guide</i> • <i>Unica Platform Upgrade Guide</i>
Implement the Cognos® reports provided with Unica	<i>Unica Reports Installation and Configuration Guide</i>

Table 5. Configure and use Unica Platform

Task	Documentation
<ul style="list-style-type: none"> • Adjust configuration and security settings for products • Integrate with external systems such as LDAP and web access control • Implement single sign-on with diverse applications using SAML 2.0-based federated authentication or single sign-on • Run utilities to perform maintenance on products • Configure and use audit event tracking • Schedule runs of Unica objects 	<i>Unica Platform Administrator's Guide</i>

Chapter 2. Planning the Unica Platform installation

When you plan your Unica Platform 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, `rwrx-rx-x`.

The following additional permissions are necessary for UNIX™:

- The user account that installs Campaign 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, `rwrx-rx-x`.



Note: For versions 12.0.0 and later, to execute Optimize sessions, users are required to apply for licenses. For more details, contact the HCL Support or Sales team.

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: Ensure that you do not select the database type "Informix" during installation of versions 12.0.0.x and 12.1.0.0, since it is not functional.



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.



Note: If you install or upgrade Unica product (with non-root user) on a machine which also has OneDB installed, then before running the installer you will need to rename (`/var/.com.zerog.registry.xml`) file to some other name. Once Unica upgrade installer is completed this file can be restored back to original state.

Unica Platform installation worksheet

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

Use the following table to gather information about the database that contains your Unica Platform system tables:

Table 6. Information about the database

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

Refer the following information

Field	Notes®
Database type	
Database name	
Database account user name	
Database account password	

Field	Notes®
JNDI name	ContactCentra IDS
ODBC name	

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:

- Database type
- 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, `mycompany.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 mis-match in domain name entries, you may 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 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 7. 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	<ol style="list-style-type: none"> 1. Unica Platform 2. Unica Campaign and Unica Optimize <p> 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.</p>
Unica Interact	<ol style="list-style-type: none"> 1. Unica Platform 2. Unica Campaign 3. Unica Interact Design Time environment 4. Unica Interact Run Time environment 5. Unica Interact Extreme Scale Server <p>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:</p>

Table 7. Installation or upgrade order for Unica products

(continued)

For this product or combination:	Install or upgrade in this order:
	<ol style="list-style-type: none"> 1. Unica Platform 2. Unica Campaign 3. Unica Interact Design Time environment <p>If you want to install or upgrade the Interact runtime environment only, then install or upgrade the Interact runtime environment in the following order:</p> <ol style="list-style-type: none"> 1. Unica Platform 2. Unica Interact Run Time environment <p>If you want to install the Unica Interact Extreme Scale Server only, then install the Unica Interact Extreme Scale Server in the following order:</p> <ol style="list-style-type: none"> 1. Unica Platform 2. Unica Interact Run Time environment 3. Unica Interact Extreme Scale Server
Unica Plan	<ol style="list-style-type: none"> 1. Unica Platform 2. Unica Plan <p> 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.</p>
Unica Interact Advanced Patterns	<ol style="list-style-type: none"> 1. Unica Platform 2. Unica Campaign 3. Unica Interact 4. Unica Interact Advanced Patterns
IBM SPSS Modeler Advantage Enterprise Marketing Management Edition	<ol style="list-style-type: none"> 1. IBM SPSS Modeler Advantage Enterprise Marketing Management Edition

Chapter 3. Creating the Unica Platform data sources

You must create Unica Platform data sources before you can install Unica Platform.

Complete the following steps to prepare the data sources for Unica Platform:

1. Create a database or a database schema for the Unica Platform system tables.

The following table provides information about vendor-specific guidelines for creating a database or a database schema for the Unica Platform system tables.

Table 8. Guidelines for creating data sources



Database vendor	Guideline
Oracle	Enable the auto commit feature for the environment open. See the Oracle documentation for instructions.
DB2®	Set the database page size to at least 16k (32k if you need to support Unicode). See the DB2® documentation for instructions.
MariaDB	Lower_case_table_names is 1 so that table names are considered 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.> e.g. Set this to 25,92,000 (seconds) in case the set up can be inactive for 30days max_connections=<The maximum number of simultaneous client connections.>
SQL Server	Use either SQL Server authentication only, or both SQL Server and Windows™ authentication, because the Platform requires SQL Server authentication. If required, change the database configuration so that your database authentication includes SQL Server. Also be sure that TCP/IP is enabled in your SQL Server.



Note: If you plan to enable locales that use multi-byte characters (for example, Chinese, Korean, and Japanese), ensure that the database is created to support them.

2. Create a system user account.

The system user account must have the following rights:

- CREATE TABLES
- CREATE VIEWS (for reporting)

- CREATE SEQUENCE (Oracle only)
 - CREATE INDICES
 - ALTER TABLE
 - INSERT
 - UPDATE
 - DELETE
3. Create ODBC or native connections.
 4. Configure the web application server for your JDBC driver.
 5. Create JDBC connections in the web application server.

Creating the JDBC connection in the web application server

About this task

The following guidelines are also applicable for Contact Central, if user selects Contact Central as a component during installation.

The Unica Platform web application must be able to communicate with its system table database using a JDBC connection.

You must create this JDBC connection in the web application server where you plan to deploy Unica Platform.

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

- The installer creates the data source using `UnicaPlatformDS` 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.
- Also, if you are using Tomcat, you must add the JDBC driver to your classpath manually even if you allow the installer to create the data source.
- Automatic data source creation during installation is not supported for the JBoss Application Server.

Follow these guidelines if you decide to create the data source manually.

- In WebSphere®, set the classpath for your database driver during this process.
- When the Unica Platform system tables are created in a different schema from the default schema of the database login user, you must specify that non-default schema name in the JDBC connection used to access the system tables.
- In Tomcat, set the classpath for your database driver during this process.
- In JBOSS, set the classpath for your database driver by adding the module for the JDBC driver and register the SQL JDBC driver.
- You must use `UnicaPlatformDS` as the JNDI name. This name is required, and is noted in the [Unica Platform installation worksheet on page 12](#).

Configuring the web application server for your JDBC driver

The web application server where you plan to deploy Unica Platform 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.

WebSphere

About this task

If you are using WebSphere and if you allow the installer to create the data source when you install the product, then you can skip the step in the following procedure about setting the classpath. Otherwise, if you want to allow the installer to create the data source, then perform the following step.

To enable automatic data source creation during installation, on the **Datasource Creation** panel, select the **Create Datasource** checkbox and provide information about your application server WebSphere profile.

WebLogic

About this task

If you are using WebLogic, perform the complete steps.

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.

Use the following guidelines after you obtain the JDBC driver.

- If the driver does not exist on the server where you plan to deploy Unica Platform, 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 Unica Platform.
2. Add the full path to the driver, including the file name, to the class path of the web application server where you plan to deploy Unica Platform.

Use the following guidelines.

- For all supported versions of WebLogic, set the class path in the setDomainEnv script in the *WebLogic_domain_directory/bin* directory where environment variables are configured. Your driver entry must be the first entry in the class path list of values, before any existing values, to ensure that the web application server uses the correct driver. For example:

UNIX™

```
CLASSPATH="/opt/drivers/ojdbc8.jar":
${PRE_CLASSPATH}${CLASSPATHSEP}${WEBLOGIC_CLASSPATH}
${CLASSPATHSEP}${POST_CLASSPATH}${CLASSPATHSEP}${WLP_POST_CLASSPATH}"
export CLASSPATH
```

```
product="Campaign DAOP DistMkt Deliver Interact Leads Plan Optimize Platform AttribMod IntHist">
```

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 class path when you set up the JDBC providers for Unica Platform.
3. Make a note of the database driver class path in the Unica Platform installation worksheet, as you must enter the path when you run the installer.
 4. 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.

JBoss

About this task

If you are using JBoss, 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.

Use the following guidelines after you obtain the JDBC driver.

- If the driver does not exist on the server where you plan to deploy Unica Platform, 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 Unica Platform.
2. Add the full path to the driver, including the file name, to the class path of the web application server where you plan to deploy Unica Platform.

Use the following guidelines.

- 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)`
3. Make a note of the database driver class path in the Unica Platform installation worksheet, as you must enter the path when you run the installer.

4. 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.

Apache Tomcat

About this task

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.

Use the following guidelines after you obtain the JDBC driver.

- If the driver does not exist on the server where you plan to deploy Platform, 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 Unica Platform.
2. Add the full path to the driver, including the file name, to the class path of the (<Tomcat_Installed Location>/lib) web application server where you plan to deploy Platform.
 3. Make a note of the database driver class path in the Unica Platform installation worksheet, as you must enter the path when you run the installer.
 4. 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.

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>`

PostgreSQL

- Database type: User_define
- Implementation class name = `org.postgresql.ds.PGConnectionPoolDataSource`
- Driver: `postgresql-42.5.4.jar`

JNDI custom properties

- Component-managed authentication alias = Select the authentication details for user
- Mapping-configuration alias = WSLogin
- portNumber = 5432
- databaseName = <DATABASE NAME>
- serverName = the host name of the PostgreSQL server
- user = the user name of the PostgreSQL server
- password = the password associated with the user name

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, go to the **Custom Properties** for the data source, and add, modify properties as follows:

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

Add the following custom property:

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>`

Add the following custom property:

Name: `webSphereDefaultIsolationLevel`

Value: `2`

Datatype: `Integer`

PostgreSQL

- Database type: `User_define`
- Implementation class name = `org.postgresql.ds.PGConnectionPoolDataSource`
- Driver: `postgresql-42.5.4.jar`

JNDI custom properties

- Component-managed authentication alias = Select the authentication details for user
- Mapping-configuration alias = `WSLogin`
- `portNumber` = `5432`
- `databaseName` = `<DATABASE NAME>`
- `serverName` = the host name of the PostgreSQL server
- `user` = the user name of the PostgreSQL server
- `password` = the password associated with the user name

JBoss

Specify the native library path of the database driver JAR file on your server. For example: `db2jcc4.jar/ojdbc8.jar/sqljdbc4.jar`.

Use the following values if your application server is JBoss:

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>,valid-connection-checker-class-name=org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker`

For example `./subsystem=datasources/data-source=UnicaPlatformDS:add(jndi-name="java:/UnicaPlatformDS",connection-url="jdbc:sqlserver://localhost:1433;databaseName=plat11",driver-name=sql,user-name=sa,password=test1234,valid-connection-checker-class-name="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>`

For example

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>`

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`

PostgreSQL

- Database type: User_define
- Implementation class name = org.postgresql.ds.PGConnectionPoolDataSource
- Driver: postgresql-42.5.4.jar

JNDI custom properties

- Component-managed authentication alias = Select the authentication details for user
- Mapping-configuration alias = WSLogin
- portNumber = 5432
- databaseName = <DATABASE NAME>
- serverName = the host name of the PostgreSQL server
- user = the user name of the PostgreSQL server
- password = the password associated with the user name

Tomcat

Specify the native library path of the database driver JAR file on your server. For example: db2jcc4.jar/ojdbc7.jar/sqjjdbc4.jar.



Note: If you want to encrypt the password then run the utility encryptTomcatDBPasswords.sh from <PLATFORM_HOME>/tools/bin. To know more about utility, visit Unica Platform Admin Guide > Chapter 20. Unica Platform utilities and SQL scripts > Unica Platform Utilities > encryptTomcatDBPassword.

Use the following values if your application server is Tomcat:

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 type: javax.sql.DataSource
- Driver URL: jdbc:sqlserver://<your_db_host>[\
\<named_instance>]:<your_db_port>;databaseName=<your_db_name>

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>`

MariaDB

- Driver: MariaDB JDBC Driver
- 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>`

PostgreSQL

- Database type: User_define
- Implementation class name = `org.postgresql.ds.PGConnectionPoolDataSource`
- Driver: `postgresql-42.5.4.jar`

JNDI custom properties

- Component-managed authentication alias = Select the authentication details for user
- Mapping-configuration alias = WSLogin
- `portNumber` = 5432
- `databaseName` = <DATABASE NAME>
- `serverName` = the host name of the PostgreSQL server
- `user` = the user name of the PostgreSQL server
- `password` = the password associated with the user name

Chapter 4. Installing Unica Platform

You must run the Unica installer to start the installation of Unica Platform. The Unica installer starts the Unica Platform 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 Platform installer starts, you must enter the required information for Unica Platform.

After installing Unica Platform, 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 Platform, make sure that the available temporary space on the computer where you install Unica Platform is more than three times the size of the Unica Platform installer.



Note: To launch the installers on Windows 2019 and Windows 2022, server please set parameter - **SET JAVA_TOOL_OPTIONS="-Dos.name=Windows 7"** - on command prompt and then launch the Installers from the same command prompt.

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 Campaign and Optimize.



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


Table 9. Installation files

Operating system	Installation file
Windows™: GUI and console mode	<p><i>Product_N.N.N.N_win64.exe</i>, where <i>Product</i> is the name of your product, <i>N.N.N.N</i> is the version number of your product, and Windows™ 64-bit is the operating system on which the file must be installed.</p> <p><i>Product_N.N.N.N_win.exe</i>,</p> <p>where <i>Product</i> is the name of your product, <i>N.N.N.N</i> is the version number of your product, and Windows™ 64-bit is the operating system on which the file must be installed.</p>
UNIX™: X Window System mode	<p><i>Product_N.N.N.N_linux.bin</i>, where <i>Product</i> is the name of your product, and <i>N.N.N.N</i> is the version number of your product.</p> <p><i>Product_N.N.N.N_linux.bin</i>, where <i>Product</i> is the name of your product, and <i>N.N.N.N</i> is the version number of your product.</p> <p><i>Product_N.N.N.N_RHEL64.bin</i>, where <i>Product</i> is the name of your product, and <i>N.N.N.N</i> is the version number of your product.</p>
UNIX™: Console mode	<p><i>Product_N.N.N.N.bin</i>, where <i>Product</i> is the name of your product, and <i>N.N.N.N</i> is the version number of your product. This file can be used for installation on all UNIX™ operating systems.</p>
SUSE Linux	<p><i>Product_N.N.N.N_linuxsuse64.bin</i>, where <i>Product</i> is the name of your product, and <i>N.N.N.N</i> is the version number of your product. This file can be used for installation on all ™SUSE Linux operating systems.</p>

Installing Unica Platform using the GUI mode

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

Before you begin

 **Important:** Before you use the GUI mode to install Unica Platform, make sure that the available temporary space on the computer where you install Unica Platform is more than three times the size of the Unica Platform installer.

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

Complete the following actions to install Unica Platform 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 10. Unica installer GUI

Window	Description
Introduction	<p>This is the first window of the Unica installer. You can open the Unica Platform 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.</p> <p>Click Next to go to the next window.</p>
Response Files Destination	<p>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 the GUI mode</p> <p>Click Choose to browse to a location where you want to store the response files.</p> <p>Click Next to go to the next window.</p>
HCL Unica products	<p>In the Install Set list, select Custom to select the products that you want to install.</p> <p>In the Install Set area, you can see all the products whose installers are in the same directory on your computer.</p> <p>In the Description field, you can view the description of the product that you select in the Install Set area.</p>

Window	Description
Installation Directory	<p>Click Next to go to the next window.</p> <p>In the Specify the installation directory field, click Choose to browse to the directory where you want to install your product.</p> <p>If you want to install the product in the folder where the installers are stored, click Restore Default Folder.</p> <p>Click Next to go to the next window.</p>
Select Application Server	<p>Select one of the following application servers for the installation:</p> <ul style="list-style-type: none"> ◦ IBM® WebSphere® ◦ Oracle WebLogic ◦ JBoss ◦ Apache Tomcat® <p>Click Next to go to the next window.</p>
Platform Database Type	<p>Select the appropriate Unica Platform database type.</p> <p>Click Next to go to the next window.</p>
Platform Database Connection	<p>Enter the following information about your database:</p> <ul style="list-style-type: none"> ◦ Database host name ◦ Database port ◦ Database name or System ID (SID) ◦ Database user name ◦ Database password <p>Click Next to go to the next window.</p>
Platform Database Connection (continued)	<p>Review and confirm the JDBC connection.</p> <p>Click Next to go to the next window. The URL can be customized with additional parameters if needed.</p>
Preinstallation Summary	<p>Review and confirm the values that you added during the installation process.</p> <p>Click Install to start the installation process.</p> <p>The Unica Platform installer opens.</p>

4. Use the information in the following table to navigate the Unica Platform installer.

Table 11. Unica Platform installer GUI

Window	Description
Introduction	<p>This is the first window of the Unica Platform installer. You can open the Unica Platform installation and upgrade guides from this window.</p> <p>Click Next to go to the next window.</p>
Software License Agreement	<p>Carefully read the agreement. Use Print to print the agreement. Click Next after you accept the agreement.</p>
Country for Install	<p>If you are installing Unica Platform in one of the countries that are listed in this window, click Yes.</p> <p>If you are installing Unica Platform in one of the countries that are not listed in this window, click No.</p> <p>Click Next to go to the next window.</p>
Installation Directory	<p>Click Choose to browse to the directory where you want to install your product or accept the default value.</p> <p>Click Next to go to the next window.</p>
Platform Components	<p>In the Install Set list, select Custom to select the components that you want to install.</p> <p>In the Install Set area, you can see all the Unica Platform components.</p> <p>You can select the following components:</p> <ul style="list-style-type: none"> ◦ Unica Platform utilities ◦ Unica Platform web application ◦ Reports for Cognos® 11 BI <p>If you are using Cognos reports, you must select this option for a new or upgrade installation. This option installs authentication-related files and the latest GlobalStyleSheet.css file for Cognos reports.</p> <p>Install Reports for IBM Cognos 11 BI in the same location where Cognos 11 BI is installed.</p>


Window	Description
	<ul style="list-style-type: none"> ◦ Unica Insights reports: Select to install Unica Insights reports for analytics <p>Unica Insights is integrated with the Unica suite and is able to provide the different reports for each module to the user. The reports are pulled up by using the BIRT technology.</p> <ul style="list-style-type: none"> ◦ Asset Picker: Select this option to install Asset viewer tool. ◦ Contact Central: Installs Contact Central that provides the ability to define communication window as per the rules defined by different Regions and also centralized place to define a capacity specific to channels. Users can define channel capacity, frequency across Unica products. On selecting Contact Central, you are prompted to provide data source connection details and application server details on separate screens. Contact Central can have a different database as that of Platform. Make sure to provide correct database credentials. <p>The steps to create the JDBC connection in the web application server and configure the web application server for your JDBC driver remains same as Platform. See the Creating the Unica Platform data sources on page 16 for details.</p> <p>Ensure that you provide JNDI name for Contact Central as "ContactCentralDS". The supported application servers are WAS and Tomcat.</p> <ul style="list-style-type: none"> ◦ AI-ML: Installs the component AI-ML. The AI-ML component uses machine learning algorithms to suggest segments from the audience tables. It is a centralized component and will be available for all Unica products when necessary. In version 12.1.7, the AI-ML component works only for suggestion of segments in Segment Central. AI-ML feature is not compatible with RHEL 7, SuSE 12 SP3, and AIX operating systems. For more details related to supported software, see the <i>HCL Unica Supported Environments</i> guide. Installing the AI-ML component adds a new node to the Settings > Configuration page. The new node is named ML. For more information on configuring ML, see Configuration Properties Guide.

Platform Connection Settings

Select one of the following connections types:

- HTTP
- HTTPS

Enter the following information:

Window	Description
	<ul style="list-style-type: none"> ◦ Network domain name, such as example.com ◦ Host name ◦ Port number <p> 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 the applications in the suite.</p> <p>Click Next to go to the next window.</p>
<p>Platform Database Setup</p>	<p>Select one of the following options for setting up the Unica Platform database:</p> <ul style="list-style-type: none"> ◦ Automatic database setup ◦ Manual database setup <p>If you select Manual database setup, use the Run Platform Configuration check box if you want to run the Unica Platform configuration.</p> <p>If you select Manual database setup, you must populate the Unica Platform system tables after the installation is complete.</p> <p>Click Next to go to the next window.</p>
<p>Platform Utilities Settings</p>	<p>Enter the following information if you plan to use the Unica Platform command-line tools:</p> <ul style="list-style-type: none"> ◦ JDBC driver class ◦ JDBC connection URL ◦ JDBC driver classpath <p>Click Next to go to the next window.</p>
<p>Platform Utilities Settings (continued)</p>	<p>Click Choose to specify the directory where Java™ is installed.</p> <p>Click Next to go to the next window.</p>
<p>Datasource creation</p>	<p>Refer the information as given for Platform. The installer can create the JDBC data source in your web application server (WebSphere, WebLogic, JBoss or Tomcat). This data source is required to enable the Unica Platform web application to connect to the Unica Platform 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.</p> <p>If you select the Create Platform Datasource check box, the installer creates the data source using information you provide..</p>

Window	Description
Application server information (WebSphere)	<ul style="list-style-type: none"> ◦ Your application server must be running. ◦ Enter any name for the datasource. <p>The installer creates the JNDI name automatically (UnicaPlatformDS).</p> <ul style="list-style-type: none"> ◦ Enter the directory in which the application server is installed. <p>For WebSphere, this is the directory that contains the <code>profiles</code> directory. You must restart WebSphere after the installer finishes and before you use this data source.</p> <p>For WebLogic, this is the directory that contains the <code>common</code> directory.</p> <p>Click Next to go to the next window.</p>
Application Server Information (JBOSS)	<p>Applies only if your application server is WebSphere.</p> <ul style="list-style-type: none"> ◦ Enter the application server profile where you plan to deploy Platform. ◦ Enter the name of the server in the profile. ◦ If security is enabled in WebSphere, enter the administrator user ID and password. <p>Click Next to go to the next window.</p> <p>Automatic Datasource creation for Jboss is not supported</p>
Application server information (WebLogic)	<p>Applies only if your application server is WebLogic.</p> <p>Enter the domain server name and the administrator user ID and password. If SSL is enabled, enter the HTTPS port for the domain.</p> <p>Note that you must add the database driver classpath to WebLogic in the WebLogic administration console.</p> <p>Click Next to go to the next window.</p>
Application server information (Tomcat)	<p>Applies only if your application server is Tomcat</p> <p>Enter the Tomcat installation and Tomcat instance path.</p> <p>Click Next to go the next window.</p>
Default locale	<p>Select a default locale for your installation. English is selected by default.</p> <p>Click Next to go to the next window.</p>

Window	Description
IBM Cognos 11 BI Location	<p>You can see this window if you selected to install the Cognos® reports in the Platform Components window.</p> <p>Click Choose to specify the directory where Cognos® 11 BI is installed.</p> <p>Click Next to go to the next window.</p>
Preinstallation Summary	<p>Review and confirm the values that you added during the installation process.</p> <p>Click Install to start the installation process.</p> <p>The Unica Platform installer opens.</p>
Installation Complete	<p>Click Done to close the Unica Platform installer and go back to the suite installer.</p>

5. Follow the instructions on the Unica installer to finish installing Unica Platform.
Use the information in the following table to take the appropriate actions on each window in the Unica installer.

Table 12. Unica installer GUI

Window	Description
Deployment EAR file	<p>Specify whether you want to create an enterprise archive (EAR) file to deploy your Unica products.</p> <p>Click Next to go to the next window.</p>
Package EAR file	<p>You can see this window if you select Create an EAR file for deployment in the Deployment EAR file window.</p> <p>Select the applications that you want to package in the EAR file.</p>
EAR file details	<p>Enter the following information for your EAR file:</p> <ul style="list-style-type: none"> ◦ Enterprise application identifier ◦ Display name ◦ Description ◦ EAR file path

Window	Description
EAR file details (continued)	<p>Select Yes or No to create an additional EAR file. If you select Yes, you must enter the details for the new EAR file.</p> <p>Click Next to complete the installation of your product.</p>
Deployment EAR file	<p>Specify whether you want to create another EAR file to deploy your Unica products.</p> <p>Click Next to go to the next window.</p>
Installation Complete	<p>This window provides the locations of the log files that are created during installation. You can view the log files when you exit from the installer.</p> <p>Click Previous if you want to change any installation details.</p> <p>Click Done to close the suite installer.</p>

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 Platform using the console mode

The console mode allows you to install Unica Platform 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 Platform, 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 Platform installer.
2. Complete one of the following actions to run the Unica installer.

Choose from:

- For Windows™, enter the following command:

```
hcl_unica_installer_full_name -i console
```

For example, `HCL_Unica_Installer_12.1.0.0_win.exe -i console`

- For UNIX™, invoke the `hcl_unica_installer_full_name.sh` file.

For example: `HCL_Unica_Installer_12.1.0.0.sh`

3. Run the Unica installer file.

Example

For example: `HCL_Unica_Installer_12.1.0.0.sh`

4. Follow the directions that are displayed in the command-line prompt. Use the following guidelines when you have to 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.

5. The Unica installer launches the Unica Platform installers during the installation process. Follow the instructions in the command-line prompt window of the Unica Platform installers.
6. After you enter `quit` in the Unica Platform installers 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 Unica Platform.



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 Platform silently

Use the unattended or silent mode to install Unica Platform multiple times.

Before you begin

Before you install Unica Platform, make sure that you have configured the following elements:

- An application server profile
- A database schema

About this task

When you install Unica Platform 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](#).
- 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_full_name -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_full_name_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 Platform. 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 13. Description of sample response files

Sample response file	Description
<code>installer.properties</code>	The sample response file for the Unica master installer.
<code>installer_product initials and product version number.properties</code>	<p>The sample response file for the Unica Platform installer.</p> <p>For example, <code>installer_ucn.n.n.n.properties</code> is the response file of the Unica Campaign installer, where <i>n.n.n.n</i> is the version number.</p> <p>For example, <code>installer_umpn.n.n.n.properties</code> is the response file of the Unica Platform installer, where <i>n.n.n.n</i> is the version number.</p> <p>For example, <code>installer_uln.n.n.n.properties</code> is the response file of the Leads installer, where <i>n.n.n.n</i> is the version number.</p>
<code>installer_report pack initials, product initials, and version number.properties</code>	<p>The sample response file for the reports pack installer.</p> <p>For example, <code>installer_urpcn.n.n.n.properties</code> is the response file of the Unica Campaign reports pack installer, where <i>n.n.n.n</i> is the version number.</p>

Unica Platform components

The Unica Platform application contains the common navigation, reporting, user administration, security, scheduling, and configuration management features. For each Unica environment, you must install and deploy the Unica Platform once.

To use the Unica Platform utilities on additional computers, you must install the utilities and the web application on the additional computers. This is needed because the utilities use the jar files in the web application. However, when you install the Unica Platform to use the utilities, you do not have to deploy the Unica Platform again, and you do not have to create additional Unica Platform system tables.

The following table describes the components that you can select when you install Unica Platform:

Table 14. Platform components

Component	Description
Unica Platform utilities	Command line tools that allow you to work with the Unica Platform system table database from the command line to import and export configurations, create partitions and data filters, and

Table 14. Platform components

(continued)


Component	Description
Unica Platform web application	restore the platform_admin user. Install this on every machine where you want to be able to use Unica Platform utilities. The web application that supplies the common user interface, security, and configuration management for Unica. Install this on the machine where you plan to deploy the Unica Platform.
Reports for IBM® Cognos® 10 BI	Reports integration components for IBM® Cognos®. Install this component only on the Cognos® system.
Unica Insights Reports	Reports integration component for Unica Insights project. Post installation, the administrator must configure it. Unica Insights is integrated with the Unica suite and is able to provide the different reports for each module to the user. The reports are pulled up by using the BIRT technology. Post installation, the administrator must configure it. The configuration details are available in the Unica Insights Reports Installation and Configuration Guide.
Asset Picker	Installs Asset Picker that you can use to link your content artifacts to offers.
Contact Central	Installs Contact Central that provides ability to define communication window as per the rules defined by different regions and also centralized place to define a capacity specific to channels. On selecting Contact Central, you are prompted to provide data source connection details and application server details on separate screens. The steps to create the JDBC connection in the web application server and configure the web application server for your JDBC driver remains same as Platform. See the Creating the Unica Platform data sources on page 16 for details. Ensure that you provide JNDI name for Contact Central as "ContactCentralDS". The supported application servers are WAS and Tomcat.
AI-ML	 Note: By default only automatic database setup is available for Contact Central. Post installation, the administrator must configure it. The configuration details are available in the Unica Contact Central Administration Guide. AI-ML: Installs the component AI-ML. The AI-ML component uses machine learning algorithms to suggest segments from the audience tables. It is a centralized component and will be available for all Unica products when necessary. In version 12.1.7, the AI-ML component works only for suggestion of segments in Segment Central. AI-ML feature is not compatible with RHEL 7, SuSE

Table 14. Platform components

(continued)

Component	Description
	12 SP3, and AIX operating systems. For more details related to supported software, see the <i>HCL Unica Supported Environments</i> guide. Installing the AI-ML component adds a new node to the Settings > Configuration page. The new node is named ML . For more information on configuring ML, see Configuration Properties Guide .

Creating and populating the Unica Platform system tables manually

When you install Unica Platform, you can select the option where the installer creates the Unica Platform system tables automatically. Alternatively, you can manually create the system tables.

Complete the following tasks to create and populate the system tables manually:

1. Run the installer as described in [Installing Unica Platform using the GUI mode on page 28](#), but with the following differences in your choices when it launches the Unica Platform installer.
 - Select **Manual database setup**.
 - Deselect the **Run Platform configuration** checkbox.
2. After the installer finishes, create the system tables manually by running the following SQL scripts appropriate for your database type against your Unica Platform system table database, as described in [SQL scripts for creating system tables on page 110](#).

Run the scripts in this order.

◦ `ManagerSchema_DBType.sql`

If you plan to support multi-byte characters (for example, Chinese, Japanese, or Korean) and your database is DB2®, use the `ManagerSchema_DB2_unicode.sql` script.

◦ `ManagerSchema__DBType_CeateFKConstraints.sql`

◦ `active_portlets.sql`

◦ `quartz_DBType.sql`

◦ `active_portlets.sql`

◦ `notification_rules.sql` (or `notification_rules_Oracle.sql` in case of oracle db)

3. Run the `populateDB` utility to populate the system tables with default users and roles.

Use this command:

```
populateDb -n Manager
```

See [populateDb on page 103](#) for complete details on using this utility.

Chapter 5. Deploying Unica Platform

You must follow a set of guidelines when you deploy Unica Platform in your web application server. There is a different set of guidelines for deploying Unica Platform on WebLogic and on WebSphere®.

When you ran the HCL installer, you completed one of the following actions:

- You included Unica Platform in an EAR file.
- You created the WAR file of Unica Platform (`unica.war`).

If you included other products in an EAR file, you must follow the deployment guidelines that are provided in the individual installation guides for the products that are included in the EAR file.

We assume that you know how to work with your web application server. Consult your web application server documentation for details such as navigation in the **Administration** console.

Important: HCL Unica products are license based, users are required to configure required licenses with HCL Unica products to start using them.

The license consumption will only be measured on Production setup.

Read carefully the HCL Unica Licensing Guide to understand how to configure, register license server and follow license consumption.

Configuration of JVM parameters for Unica products

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

Table 15. Unica Products and the JVM Parameters to be Configured

Unica Product Name	JVM Parameters
Unica Campaign	<ul style="list-style-type: none">• <code>-DUNICA_PLATFORM_CACHE_ENABLED=true</code> - Enables platform cache functionality.• <code>-DLICENSE_ENABLED=false</code> - Disables license functionality.• <code>-Dcampaign.log4j.async=true</code> - Campaign application to enable asynchronous logging. Asynchronous logging allows Log4j to log messages in a non-blocking manner, which can improve performance in high-throughput applications by offloading the logging work to a separate thread or a pool of threads. This reduces the overhead in the main execution path, especially in scenarios with heavy logging.

Table 15. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	<ul style="list-style-type: none"> • <code>-Dcom.sun.management.jmxremote</code> - Enables JMX remote management. • <code>-Dcom.sun.management.jmxremote.port=1007</code> - Specifies the port on which the JMX server will listen for connections. • <code>-Dcom.sun.management.jmxremote.ssl=false</code> - Disables SSL for JMX connections. • <code>-Dcom.sun.management.jmxremote.authenticate=false</code> - Disables authentication for JMX connections (not recommended for production). • <code>-Dcom.sun.management.jmxremote.password.file=../conf/jmxremote.password</code> - Points to a file that stores passwords for JMX authentication. • <code>-Dcom.sun.management.jmxremote.access.file=../conf/jmxremote.access</code> - Specifies the file that defines access roles for JMX connections. • <code>-DUNICA_GOTO_CREATEEARFILE=TRUE</code> - This parameter is used by installer for creating EAR file post installation. • <code>-Dfile.encoding=UTF-8</code> - Sets the default file encoding for the JVM to UTF-8, ensuring consistent text encoding for file I/O operations. • <code>-Dclient.encoding.override=UTF-8</code> - Overrides the encoding for client-related operations to UTF-8, enforcing consistent character encoding in client-server communications. e.g : If application communicates with external systems or clients, this property could ensure that UTF-8 is used consistently for encoding/decoding data between the client and server. • <code>-Djboss.as.management.blocking.timeout=3600</code> - This option sets the blocking timeout for JBoss Application Server management operations to 3600 seconds (1 hour). It controls how long management operations (e.g.,

Table 15. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	<p>deployments, configuration changes) can block before timing out.</p> <p>These JVM options configure various aspects of the ACOOptAdmin tool, including SSL settings, headless mode, and ETL (Extract, Transform, Load) functionalities related to different Campaign components.</p> <p>SET SSL_OPTIONS=-Djavax.net.ssl.keyStoreType="JKS" - Specifies the type of key store to use for SSL connections. "JKS" (Java KeyStore) is the default format for storing cryptographic keys and certificates in Java</p> <ul style="list-style-type: none"> • - <code>Djavax.net.ssl.trustStore="C:\security\myTrustStore.jks"</code> • -Djavax.net.ssl.trustStorePassword=myPassword <p>Configuring the ACOOptAdmin tool (ACOOptAdmin.sh (UNIX) or ACOOptAdmin.bat (Windows) file)</p> <ul style="list-style-type: none"> • -Djavax.net.ssl.keyStoreType=keyStoreTypeValue - Specifies the type of key store for SSL. • -Djavax.net.ssl.keyStore=<keyStoreValue> - Points to the location of the key store file containing private keys and certificates. • - <code>Djavax.net.ssl.keyStorePassword=<keyStorePasswordValue></code> - Sets the password to access the key store. • -Djavax.net.ssl.trustStore=<trustStoreValue> - Specifies the location of the trust store containing trusted certificates. • - <code>Djavax.net.ssl.trustStorePassword=<trustStorePasswordValue></code> • -Djava.awt.headless=true - Enables headless mode, which allows the application to run without a graphical user interface (useful for running on servers without a display). • -Dcampaign.deliveretl.disabled=true - Disables the Campaign's Deliver ETL process. Without this jvm setting

Table 15. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	<p>-Dcampaign.deliveretl.disabled=true , the Deliver ETL process remains active.</p> <ul style="list-style-type: none"> -Dcampaign.interactetl.disabled=true - Disables the Campaign's Interact ETL process. Without this jvm setting -Dcampaign.interactetl.disabled=true , the Interact ETL process remains active. -Dcampaign.journeyetl.disabled=false - Enables the Campaign's Journey ETL process. -Dcampaign.journeyetl.disabled=true - Disables the Campaign's Journey ETL process Without this jvm setting -Dcampaign.journeyetl.disabled=true and journey also installed then Journey ETL process remains active. <p>EMAIL_TLS_ENABLED - Enables TLS (Transport Layer Security) for secure email communications.</p> <p>EMAIL_SSL_TRUST_ENABLED - Enables SSL (Secure Sockets Layer) for trusting SSL certificates in email communications.</p> <p>SCH_ASYNC_EXECUTION_ENABLED - Enables asynchronous execution for scheduled tasks or jobs.</p>
Unica Audience Central	<ul style="list-style-type: none"> -DAUDIENCE_CENTRAL_HOME=<Install_Base_DIR>/AudCentral
Unica Contact Central	<ul style="list-style-type: none"> CONTACTCENTRAL_HOME UNICA_PLATFORM_HOME
Unica Content Integration	<ul style="list-style-type: none"> -DASSET_PICKER_HOME -DUNICA_PLATFORM_HOME
Unica Deliver	<ul style="list-style-type: none"> -Dcampaign.deliveretl.disabled=true
Unica Interact	<p>Interact design time</p> <p>The following are the JVM parameters for Interact design time:</p>

Table 15. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	<ul style="list-style-type: none"> • INTERACTDT_HOME • Interact.UsernameToAlwaysDeployFor • com.unicacorp.interact.deliver.templateTimeout • DeploymentServletParameterEncoding • com.unicacorp.interact.flexoffers.defaultDateFormat • 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 • com.unicacorp.interact.playback.maxFilteredIdsCount • com.unicacorp.interact.playback.APITimeoutInSecs <p>Interact run time</p> <p>The following are the JVM parameters for Interact run time:</p> <ul style="list-style-type: none"> • 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.offerserving.maxOfferAllocationInMemoryPerInstance • interact.offerserving.maxDistributionPerIntervalPerInstanceFactor • interact.ignitePort • com.unicacorp.interact.chDupeCheckLimit • com.unicacorp.interact.rhDupeCheckLimit • com.unicacorp.interact.chSuppressDupe • com.unicacorp.interact.rhSuppressDupe


Table 15. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	<ul style="list-style-type: none"> • <code>com.unicacorp.interact.testclient.nullValue</code> • <code>interact.ehcache.config</code> • <code>interact.api.dateFormat</code> • <code>com.hcl.interact.testrun.rowlimit</code> • <code>Interact.DisableExceptionStackTracesInMacros</code> • <code>com.unicacorp.interact.enableDetailStats</code> • <code>com.unica.interact.deployment.timeoutInSecs</code> • <code>com.ibm.interact.instance.name</code> • <code>com.unicacorp.interact.invalidPaths</code> • <code>interact.XSessResponseConsumerManager.generateOnlyOneResponse</code> • <code>tryToPreserveInexactFloatValues</code> • <code>com.unicacorp.interact.propertyRefreshInterval</code> • <code>com.unicacorp.interact.scheduledTasksProcessInterval</code> • <code>com.unicacorp.interact.eventpatterns.parallelism</code> • <code>com.unicacorp.interact.eventpatterns.restartRetries</code> • <code>com.unicacorp.interact.eventpatterns.evaluateTimeoutMilli</code> • <code>com.unicacorp.interact.eventpatterns.restartRetryDelayInSec</code> • <code>Interact.advisoryMessageEncodingOverrides</code> • <code>com.unica.interact.api.insertSessionIDAsCookie</code> • <code>com.unica.interact.api.SessionIDCookieName</code> • <code>InteractMsgCode</code> • <code>com.ibm.interact.triggeredmessage.enableJMSConsumer</code> • <code>com.unicacorp.interact.maxStringLengthInFormatMacro</code> • <code>ContinueEvaluatingBranchAndAdvOptTreatmentLogicDespiteExceptionMessageList</code> • <code>DisableDecisionProcessBoxAndAdvOptTreatmentLogging</code> <p>Valid values are <code>True</code> or <code>False</code>. The default value is <code>False</code>. If you set the value to <code>False</code>, when evaluating the expressions of each branch, if the system encounters any errors, it logs the errors.</p> <ul style="list-style-type: none"> • <code>Interact.DPB.MuffleErrorsForMutuallyExclusiveBranchesWithNullValues</code> <p>Valid values are <code>True</code> and <code>False</code>. The default value is <code>False</code>. If you set the value to <code>False</code> and if the value of</p>

Table 15. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	<p data-bbox="824 317 1448 548">-DDisableDecisionProcessBoxAndAdvOptTreatmentLogging is also set to <code>False</code>, during the evaluation of the expressions of each branch, of a Decision process box whose branches are mutually exclusive, if the system encounters any errors, it logs the errors. Otherwise, the system logs the errors as debug.</p> <ul data-bbox="808 575 1435 1894" style="list-style-type: none"> <li data-bbox="808 575 1062 596">• <code>TwoDigitYearStartDate</code> <li data-bbox="808 615 1170 636">• <code>Interact.enableTwoDigitYearFix</code> <li data-bbox="808 655 1170 676">• <code>com.ibm.interact.evpattel.conf</code> <li data-bbox="808 695 1341 716">• <code>com.unicacorp.interact.minTreatmentsPerThread</code> <li data-bbox="808 735 1317 756">• <code>com.unicacorp.interact.maxTreatmentPoolSize</code> <li data-bbox="808 774 1227 795">• <code>CircuitBreaker.processTimeoutMillis</code> <li data-bbox="808 814 1341 835">• <code>com.unicacorp.interact.event.asyncTimeoutMSec</code> <li data-bbox="808 854 1292 875">• <code>com.unicacorp.interact.event.actionTimeout</code> <li data-bbox="808 894 1065 915">• <code>Interact.HTML.Enabled</code> <li data-bbox="808 934 1235 955">• <code>Interact.HTML.MaxRequestDurationInMs</code> <li data-bbox="808 974 1211 995">• <code>Interact.HTML.RecordIndividualAPIs</code> <li data-bbox="808 1014 1292 1035">• <code>Interact.HTML.MaxStartSessionDurationInMs</code> <li data-bbox="808 1054 1260 1075">• <code>Interact.HTML.MaxGetOffersDurationInMs</code> <li data-bbox="808 1094 1260 1115">• <code>Interact.HTML.MaxPostEventDurationInMs</code> <li data-bbox="808 1134 1268 1155">• <code>Interact.HTML.MaxGetProfileDurationInMs</code> <li data-bbox="808 1173 1227 1194">• <code>Interact.HTML.LogErrorsEveryNthTime</code> <li data-bbox="808 1213 1211 1234">• <code>Interact.HTML.UseMillisecondTimers</code> <li data-bbox="808 1253 1040 1274">• <code>Interact.HTML.Debug</code> <li data-bbox="808 1293 1435 1314">• <code>com.unicacorp.interact.suppressWarningOnAnonymousUser</code> <li data-bbox="808 1333 1390 1354">• <code>com.hcl.interact.eventpatterns.printPatternAction</code> <li data-bbox="808 1373 1317 1394">• <code>com.hcl.interact.eventpatterns.eagerPersist</code> <li data-bbox="808 1413 1341 1434">• <code>com.ibm.interact.triggeredmessage.addPerfData</code> <li data-bbox="808 1453 1390 1474">• <code>com.unicacorp.interact.learning.disableAggregator</code> <li data-bbox="808 1493 1365 1514">• <code>com.unicacorp.interact.learning.disableDeletion</code> <li data-bbox="808 1533 1349 1554">• <code>com.unicacorp.interact.learning.ignoreInterval</code> <li data-bbox="808 1572 1284 1593">• <code>interact.services.loader.saveLoaderFiles</code> <li data-bbox="808 1612 1373 1633">• <code>ConvertEveryNULLAttributeValueToAJEPNullConstant</code> <li data-bbox="808 1652 1000 1673">• <code>includeJoinInfo</code> <li data-bbox="808 1692 1365 1713">• <code>com.unicacorp.interact.deployment.reloadTimeout</code> <li data-bbox="808 1732 1292 1753">• <code>com.ibm.interact.lockTimeWarningThreshold</code> <li data-bbox="808 1772 1284 1793">• <code>com.unicacorp.interact.cache.maxWaitTime</code> <li data-bbox="808 1812 1146 1833">• <code>DEFAULT_PERSISTENCE_PROVIDER</code> <li data-bbox="808 1852 1349 1873">• <code>com.unicacorp.interact.playback.purgeBatchSize</code>

Table 15. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	 Note: We have removed the JVM parameter <code>interact.runtime.instance.name</code> . Use the JVM parameter <code>com.ibm.interact.instance.name</code> to set the name of the Interact run-time instance.
Unica Journey	
Unica Plan	<ul style="list-style-type: none"> • <code>-Dplan.log4j.config=<plan_home>/conf/plan_log4j.xml</code> • <code>-Dplan.log4j.async=true</code> • <code>-Dplan.home=<plan_home></code> • <code>-Dplan.slow.query.threshold=<value in milliseconds></code> (optional) • <code>-Dplan.slow.query.show.stack.trace=true</code> (optional) • <code>-Dcampaign.https.port=<campaign's https port></code> <p>If Plan communicates with Campaign over <code>https</code>, and if Campaign is configured to run on the <code>https</code> port other than 8443, add this system property to the JVM arguments of the Plan application's JVM.</p>
Unica Platform	<ul style="list-style-type: none"> • <code>-DENABLE_PLATFORM_LOG4J_XML_LOGGING=TRUE</code> • <code>-DENABLE_PERSISTENT_NAMEID_FORMAT=TRUE</code> • <code>-DUNICA_PLATFORM_HOME=<platform_home_directory_path></code> • <code>-DmaxSessions=<number of active sessions allowed></code>- This is optional JVM parameter. By default, there is no restriction on concurrent/active sessions for the same user.
Unica Centralized Offer Management	<ul style="list-style-type: none"> • <code>-DOFFER_HOME=<Install_Base_DIR>/CentralizedOffer</code> • <code>-DUNICA_PLATFORM_CACHE_ENABLED=false</code> • <code>-DUNICA_PLATFORM_LOCAL_CACHE_ENABLED=false</code> • <code>-Doffer.isDebugEnabled=false</code> • <code>-Dlog4j2.formatMsgNoLookups=true</code>
Unica Segment Central	<ul style="list-style-type: none"> • <code>-DSEGMENT_CENTRAL_HOME=<Install_Base_DIR>/SegCentral</code>

Guidelines for deploying Unica Platform on WebLogic

You must follow a set of guidelines when you deploy Unica Platform on the WebLogic application.

Use the following guidelines when you deploy the Unica Platform products on any supported version of WebLogic:

- Unica products customize the Java™ virtual machine (JVM) that is used by WebLogic. If you encounter errors related to JVM, you can create a WebLogic instance that is dedicated to Unica products.
- Open the `startWebLogic.cmd` file and verify that the SDK that is selected for the WebLogic domain that you are using is the Sun SDK for the **JAVA_VENDOR** variable.

The **JAVA_VENDOR** variable must be set to `sun` (**JAVA_VENDOR=Sun**). If the **JAVA_VENDOR** variable is set to **JAVA_VENDOR**, it means that JRockit is selected. You must change the selected SDK, because JRockit is not supported. Refer to the BEA WebLogic documentation to change the selected SDK.

- Deploy Unica Platform as a web application.
- If you are configuring WebLogic to use the IIS plug-in, review the BEA WebLogic documentation.
- Complete the following tasks if your installation must support non-ASCII characters, for example for Portuguese or for locales that require multi-byte characters:
 1. Edit the `setDomainEnv` script in the `bin` directory under your WebLogic domain directory to add the following Java options to `JAVA_VENDOR`:

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

In case you are deploying a non-production setup, add

```
-DENABLE_NON_PROD_MODE=true
```

In case of a production setup, this Java option must be removed or set to `false`.

2. In the **WebLogic** console, click the **Domain** link on the home page.
 3. In the **Web Applications** tab, select the **Archived Real Path Enabled** check box.
 4. Restart WebLogic.
 5. Deploy and start the EAR file or the `unica.war` file.
- If deploying in a production environment, set the JVM memory heap size parameters to 1024 by adding the following line to the `setDomainEnv` script:

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

- For DB2, set `progressiveStreaming = 2 UnicaPlatformDS` datasource configuration.

Guidelines for deploying Unica Platform on WebSphere

You must follow a set of guidelines when you deploy Unica Platform on WebSphere.

Make sure that the version of WebSphere® meets the requirements that are described in the Recommended Software Environments and Minimum System Requirements document, including any necessary fix packs. Use the following guidelines when you deploy Unica Platform on WebSphere®:

- Set a custom property in WebSphere®.
 - **Click Servers > Server Types > Websphere Application Server > <server_name> > Web Container Settings > Web Container**
 - Under additional settings click on **Custom Properties**
 - On **Custom Properties** page, click **New** and enter `com.ibm.ws.webcontainer.invokefilterscompatibility` as the property name and **true** as the value.
 - Save the update and restart the server.
- Deploy the EAR file or `unica.war` file as an enterprise application. When you deploy the EAR file or the `unica.war` file, ensure that the JDK source level of the JSP compiler is set to Java™ 17 for SDK 1.7 and 18 for SDK 1.8, and that JSP pages are precompiled according to the following information:
 - In the form where you browse to and select the WAR file, select **Show me all installation options and parameters** so the **Select Installation Options** wizard runs.
 - In step 1 of the **Select Installation Options** wizard, select **Precompile JavaServer Pages files**.
 - In step 3 of the **Select Installation Options** wizard, ensure that the **JDK Source Level** is set to 17 for SDK 1.7 and set to 18 for SDK 1.8.

If you deployed an EAR, set the JDK Source Level for each WAR file.

- In step 8 of the **Select Installation Options** wizard, select **UnicaPlatformDS** as the matching Target Resource.
- In step 10 of the **Select Installation Options** wizard, the context root must be set to `/unica`, all lower case.
- In the **Web Container Settings > Web Container > Session Management** section of the server, enable cookies. Specify a different session cookie name for each application that is deployed. Use one of the following procedures to specify a cookie name:
 - Select the **Override session management** check box under **Session Management**.

If you deployed separate WAR files for your Unica products, in the WebSphere® console, in the **Applications > Enterprise Applications > [deployed_application] > Session Management > Enable Cookies > Cookie Name** section of the server, specify a unique session cookie name.

If you deployed EAR files for your Unica products, in the WebSphere® console, in the **Applications > Enterprise Applications > [deployed_application] > Module Management > [deployed_module] > Session Management > Enable Cookies > Cookie Name** section of the server, specify a unique session cookie name.

- If your installation must support non-ASCII characters, for example for Portuguese or for locales that require multi-byte characters, add the following arguments to **Generic JVM Arguments** at the server level:

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

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

In case you are deploying a non-production setup, add

```
-DENABLE_NON_PROD_MODE=true
```

In case of a production setup, this Java option must be removed or set to `false`.

Navigation tip: Select **Servers > Application Servers > Java and Process Management > Process Definition > Java Virtual Machine > Generic JVM Arguments**. See the WebSphere® documentation for additional details.

- In the **Applications > Enterprise Applications** section of the server, select the EAR file or WAR file that you deployed, then select **Class loading and update detection** and specify the following properties.
 - If you are deploying a WAR file:
 - For **Class loader order**, select **Classes loaded with local class loader first (parent last)**.
 - For **WAR class loader policy**, select **Single class loader for application**.
 - If you are deploying an EAR file:
 - For **Class loader order**, select **Classes loaded with local class loader first (parent last)**.
 - For **WAR class loader policy**, select **Class loader for each WAR file in application**.
- Start your deployment. If your instance of WebSphere® is configured to use a JVM version 1.7 or newer, complete the following steps to work around an issue with the time zone database.
 1. Stop WebSphere®.
 2. Download the Time Zone Update Utility for Java™ (JTZU).
 3. Follow the steps provided by the IBM® (JTZU) to update the time zone data in your JVM.
 4. Restart WebSphere®.
- In WebSphere Enterprise Applications, select **Your Application > Manage Modules > Your Application > Class Loader Order > Classes loaded with local class loader first (parent last)**.
- The recommended minimum heap size for the basic functioning of the application is 512 and the recommended maximum heap size is 1024.

Complete the following tasks to specify the heap size:

1. In WebSphere® Enterprise Applications, select **Servers > WebSphere application servers > server1 > Server Infrastructure > Java and Process Management > Process definition > Java Virtual Machine**.
2. Set the initial heap size to 512.
3. Set the maximum heap size to 1024

See the WebSphere® documentation for more information about sizing.

For DB2, set `progressiveStreaming = 2` in WebSphere console at following path: **JDBC > Data sources > UnicaPlatformDS > Custom properties**.

Guidelines for deploying Unica Platform on JBoss

You must follow a set of guidelines when you deploy Unica Platform on JBoss.

Make sure that the version of JBoss meets the requirements that are described in the HCL Enterprise Products Recommended Software Environments and Minimum System Requirements document. Use the following guidelines when you deploy Unica Platform on JBoss:

Use the following guidelines when you deploy the Unica Platform products on any supported version of JBoss:

1. Deploy the HCL EAR file or the unica.war file as an enterprise application.

For example: `deploy <Platform_Install>\unica.war`

See <https://docs.jboss.org/jbossweb/3.0.x/deployer-howto.html> for instructions on Deploying Web Server Application in JBoss.

2. Complete the following tasks if your installation must support non-ASCII characters, for example for Portuguese or for locales that require multi-byte characters:

- a. Edit the `standalone.conf` script in the bin directory under your JBOSS /bin directory to add

```
-Dfile.encoding=UTF-8

-Dclient.encoding.override=UTF-8

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

to JAVA_VENDOR.
```

In case you are deploying a non-production setup, add

```
-DENABLE_NON_PROD_MODE=true
```

In case of a production setup, this Java option must be removed or set to `false`.

- b. Restart JBoss server.

3. To ensure that the Scheduler works correctly, complete the following substeps.

- Take a backup of `<JBOSS_HOME>/standalone/configuration/standalone.xml` file.
- In `<JBOSS_HOME>/standalone/configuration/standalone.xml`, search the module name for driver.

```
<driver name="oracledriver" module="oracle.jdbc">
<xa-datasource-class>oracle.jdbc.OracleDriver</xa-datasource-class>
</driver>
```

- Add the following statement underneath `<subsystem xmlns="urn:jboss:domain:ee:4.0">` to make the module name global.

```
<global-modules>
<module name="oracle.jdbc"/>
</global-modules>
```

- Add `max-parameters="10000"` property for the following tags.
 - `<http-listener name="default"...max-parameters="10000"/>`
 - `<https-listener name="https"...max-parameters="10000"/>`
- Restart JBOSS server.

Guidelines for deploying Unica Platform on Apache Tomcat®

You must follow a set of guidelines when you deploy Unica Platform on Apache Tomcat. For Contact Central, the steps to deploy Tomcat remain same.

Make sure that the version of Apache Tomcat meets the requirements that are described in the Recommended Software Environments and Minimum System Requirements document. Use the following guidelines when you deploy Unica Platform on Apache Tomcat:

1. Deploy the HCL unica.war file as an enterprise application on Tomcat Apache server.



Note: Deployment of HCL EAR file is not supported in Tomcat.

- a. Edit the setenv.sh file for the respective product instances script in the bin directory under your Tomcat instances directory to add the following Java options:

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

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

In case you are deploying a non-production setup, add

```
-DENABLE_NON_PROD_MODE=true
```

In case of a production setup, this Java option must be removed or set to false.

- b. Restart Tomcat.

2. If deploying in a production environment, you can add JVM heap setting for that tomcat instance in app-one/bin/setenv.sh file respectively for all the instances.

For DB2, set `progressiveStreaming = 2` in the UnicaPlatformDSdatasource configuration.

3. When Unica war is deployed on Tomcat, complete the following configuration in server.xml, where unica.war is deployed.

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

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

4. You must add Unica Platform data source in unica.xml along with the path of unica.war to the Unica Platform Tomcat instance. For example

```
<Context docBase="<Platform_Install_Path>/Platform/unica.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>"/>
```

```
</Context>"/>
```



Note: This is the sample xml configuration.



Note: Add prefix "`java:/comp/env/`" when defining datasource/JNDI.

After upgrading Platform on Apache Tomcat, you must encrypt the password again from the location `<Installation_Location>/platform/tools/bin` and manually make the changes in XMLs.

You can use the `encryptTomcatDBPasswords` utility and encrypt the database password which is used by the Tomcat Application server internally. It is used to encrypt database passwords used in `Campaign.xml`, `unica.xml`, and `Platform.xml`. The utility prints the encrypted password in the command line.

The command to run the utility is: `encryptTomcatDBPasswords -d databasePassword`.

Google Chrome and Microsoft Edge configuration

Recently, the browsers Google Chrome and Microsoft Edge updated a security fix and this security fix affects the access of Unica applications. We have received some issues from our customers like:

- issues with UI
- getting logged out from Unica

These issues are observed due to the change of behavior in browsers after applying the security fix. Applying the security fix automatically enables **Origin-keyed Agent Clusters by default**. If the setting **Origin-keyed Agent Clusters by default** is enabled automatically, it prevents changes in document referrer and domain values so that malicious websites cannot execute any type of impersonation. The setting **Origin-keyed Agent Clusters by default** existed earlier as well, but was not enabled by default.

If you update Google Chrome or Microsoft Edge to the latest version, you will observe the earlier mentioned issues. Because of how Unica is designed and because the Unica suite is deployed over multiple JVMs, it is essential that you disable the **Origin-keyed Agent Clusters by default** setting for Unica to function correctly and to provide a good user experience.

As a solution, we recommend that you perform the steps mentioned in the following Knowledge Base article: https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0107185.



Note: The update to Microsoft Edge browser is very recent and the Knowledge Base article is not yet updated for the Microsoft Edge browser.

The CliffsNotes solution is as follows:

1. Open your browser and navigate to one of the following URLs based on your browser:
 - **Google Chrome:** <chrome://flags/#origin-agent-cluster-default>
 - **Microsoft Edge:** <edge://flags/#origin-agent-cluster-default>
2. From the dropdown of the highlighted parameter **Origin-keyed Agent Clusters by default**, select `Disabled`.
3. At the bottom of the page, click the **Apply Changes** button.
4. Log out of Unica applications, log back in, and verify if everything is working as expected.

Verifying your Unica Platform installation

After you install and deploy Unica Platform, you must verify that there are no errors in your Unica Platform installation and deployment. You can configure your Unica Platform installation after the verification.

Complete the following tasks to verify your Unica Platform installation:

1. Access the Unica URL in a supported web browser.

If you entered a domain when you installed Unica Platform, the URL is the following, where `host` is the machine where Unica Platform is installed, `domain.com` is the domain in which the host machine resides, and `port` is the port number on which the web application server listens.

```
http://host.domain.com:port/unica
```

2. Log in using the default administrator login, which is `asm_admin` with `password` as the password.

You will be asked to change the password. You can enter the existing password, but for good security you should choose a new one.

The default home page is the dashboard, which you will configure later.

3. Under the **Settings** menu, check the Users, User Groups, and User Permissions pages to verify that the pre-configured users, groups, roles, and permissions are present, as described in the *Unica Platform Administrator's Guide*.
4. Add a new user and group and verify that data is entered into the Unica Platform system table database.
5. Under the **Settings** menu, check the Configuration page to verify that the Unica Platform configuration properties exist.

What to do next

There are additional configuration tasks, such as configuring the dashboard, setting up user access to applications, and integrating with an LDAP or web access control system (optional). See the *Unica Platform Administrator's Guide* for instructions.

Chapter 6. Deploying Unica Contact Central

Prerequisites

Unica Contact Central requires JRE 1.8 or higher to work.



Note: Unica Contact Central does not support clustered environment.

ContactCentralDS JNDI resource for Contact Central database is required.

Setting Unica Contact Central and Unica Platform home JVM argument

Add the following JVM argument to the application server, where Unica Contact Central is required to be deployed.

```
- DCONTACTCENTRAL_HOME = path_where_contact_central_is_installed
```

```
- DUNICA_PLATFORM_HOME = platform_installation_home_directory_path
```

For example: `CONTACTCENTRAL_HOME = /opt/Platform/ContactCentral`

Point CONTACTCENTRAL_HOME to the directory, where Unica Contact Central is installed. It is installed within the ContactCentral directory under UNICA_PLATFORM_HOME.

Provide write permissions to the logs directory already created inside CONTACTCENTRAL_HOME.

Deployment procedure

You must follow a set of guidelines, when you deploy Unica Contact Central on your web application server. When you run the suite installer, you completed the following action:

- You created the WAR file of Unica Contact Central (ContactCentral.war) inside UNICA_PLATFORM_HOME/ContactCentral directory. UNICA_PLATFORM_HOME refers to the Unica Platform installation location.



Note:

- It is assumed that you possess information on how to work with your web application server. For more information, see the web application server documentation.
- Contact Central deployment is not supported on WebLogic and JBoss.

Guidelines for deploying Unica Contact Central on WebSphere

You must follow a set of guidelines when you deploy Unica Contact Central on WebSphere. Ensure that the version of WebSphere meets the requirements that are described in the Recommended Software Environments and Minimum System

Requirements document, including any required fix packs. Use the following guidelines when deploying Unica Contact Central on WebSphere:

1. Set a custom property in WebSphere®.
 - **Click Servers > Server Types > Websphere Application Server > <server_name> > Web Container Settings > Web Container**
 - Under additional settings click on **Custom Properties**
 - On **Custom Properties** page, click **New** and enter `com.ibm.ws.webcontainer.invokefilterscompatibility` as the property name and **true** as the value.
 - Save the update and restart the server.
2. Deploy the ContactCentral.war file as an enterprise application. When you deploy the ContactCentral.war file, ensure that the JDK source level is set to 18 for SDK 1.8:
 - a. In the form, select the WAR file, select **Show me all installation options** and parameters so the **Select Installation Options** wizard runs.
 - b. In step 3 of the **Select Installation Options** wizard, ensure that the JDK SourceLevel is set to 18 for SDK 1.8.
 - c. In step 8 of the **Select Installation Options** wizard, select ContactCentralDS as the matching Target Resource. ContactCentralDS JNDI resource must be present for Contact Central. Guidelines for setting up ContactCentralDS remains same as setting up UnicaPlatformDS datasource for Platform. For more information on setting up UnicaPlatformDS datasource, see *Unica Platform Installation Guide*.
 - d. In step 10 of the **Select Installation Options** wizard, the context root must be set to /ContactCentral.
3. For your installation to support non-ASCII characters, for example for Portuguese or for locales that require multi-byte characters, add the following arguments to Generic JVMArguments at the server level.

-Dfile.encoding=UTF-8

-Dclient.encoding.override=UTF-8



Note: Navigation tip: Select **Servers > Application Servers > Java and ProcessManagement > Process Definition > Java Virtual Machine > Generic JVM Arguments**. See the WebSphere documentation for additional details.

4. In the **Applications > Enterprise Applications** section of the server, select the WAR file that you deployed, then
 - For Class loader order, select **Classes loaded with local class loader first (parentlast)**.
 - For WAR class loader policy, select **Single class loader for application**.
5. Start your deployment. If your instance of WebSphere is configured to use a JVM version 1.8 or higher, complete the following steps to work around an issue with the time zone database.
 - a. Stop WebSphere.
 - b. Download the Time Zone Update Utility for Java (JTZU).
 - c. Follow the steps provided by the IBM (JTZU) to update the time zone data in your JVM.
 - d. Restart WebSphere.
6. In Websphere Enterprise Applications, select your **Application > Manage Modules > Your Application > Class Loader Order > Classes** loaded with local class loader first (parent last).

7. The recommended minimum heap size for the basic functioning of the application is 512 MB and the recommended maximum heap size is 1024 MB. Complete the following tasks to specify the heap size.
 - a. In WebSphere Enterprise Applications, select **Servers > WebSphere application servers > server1 > Server Infrastructure > Java and Process Management > Process definition > Java Virtual Machine**.
 - b. Set the initial heap size to 512m.
 - c. Set the maximum heap size to 1024m.



Note: See the *WebSphere documentation* for more information about sizing.

For DB2, set `progressiveStreaming = 2` in WebSphere console at following path: JDBC>Data sources > ContactCentralDS > Custom properties.

Guidelines for deploying Unica Contact Central on Apache Tomcat®

You must follow a set of guidelines when you deploy Unica Contact Central on Apache Tomcat. Ensure that the version of Apache Tomcat meets the requirements that are described in the *Recommended Software Environments and Minimum System Requirements* document. Use the following guidelines when you deploy Unica Content Integration on Apache Tomcat.

1. Deploy the HCL ContactCentral.war file as an enterprise application on Tomcat Apache server. Complete the following tasks if your installation must support non-ASCII characters, for example for Portuguese or for locales that require multi-byte characters.
2. Edit the `setenv.sh` file for the respective product instances script in the `bin` directory under your tomcat instances directory to add `-Dfile.encoding=UTF-8Dclient.encoding.override=UTF-8`.
3. You must add Unica Platform and Contact Central data source in `ContactCentral.xml` along with the path of `ContactCentral.war` to the ContactCentral Tomcat instance. For example

```
<Context docBase="<Platform_Install_Path>\platform\ContactCentral\ContactCentral.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="ContactCentralDS" factory="com.unica.manager.tomcat.utils.TomcatDSFactory" auth="Application" type="javax.sql.DataSource"
```

```
maxActive="30" maxIdle="10" maxWait="10000"
```

```
username="<db user for Contact central schema>" password="<db password>" driverClassName="
```

```
"<db specific class name>" url="<db specific jdbc url>
```

```
</Context>"/>
```

4. Restart Tomcat.



Note: This is the sample xml configuration.

Unica Contact Central | User Role Creation

The procedure included in this section is not required for the following scenarios:

It is a fresh installation. The installation process creates this role under default partition(partition1).

It is an upgrade and only one partition exists in the current system.

In case, it is an upgrade and system contains more than one partition, you must execute the following command manually for each partition, except partition1. The command exists under <PLATFORM_HOME>\tools\bin directory inside your Platform installation directory. This creates a Contact Central user role under each partition.

On Windows

```
populateDb.bat -n ContactCentral -p <partition_name>
```

On Unix

```
populateDb.sh -n ContactCentral -p <partition_name>
```

Chapter 7. Deploying AI-ML

This feature is deprecated from 12.1.9 and is currently available only to support backward compatibility for users who have maintained their configurations using `INI` files. In Unica Segment Central, we have introduced two features: Manage orders and Data cleansing. These two features have the same behavior as the configurations made by the `INI` files. If you have made the same changes in user interface and the `INI` files, precedence is given to the changes made via user interface.

After the installation of Unica Platform is complete, if you selected the component AI-ML for installation, you must deploy the AI-ML component post installation. The deployment steps differ for Microsoft Windows and Unix-based operating systems. There are some common procedures that you must implement for both operating systems.

- [Deploying AI-ML in Microsoft Windows on page 62](#)
- [Deploying AI-ML in Unix-based Operating Systems on page 67](#)

Deploying AI-ML in Microsoft Windows

To deploy AI-ML component in Microsoft Windows, complete the following procedures:

Configuring Python files

1. Access the `Python3.11.7` folder available under `Platform/ML` directory.
2. In the `Platform/ML/startup.bat` file, replace the placeholder `[PYTHON_BASE_LOCATION]` with the absolute path to `Python3.11.7` directory.
3. In the `Platform/ML/lib` directory, access the `site-packages` folder.
4. In the `Platform/ML/startup.bat` file, replace the placeholder `[replace_with_the_sitepackages_path]` with the absolute path to `site-packages` directory.

Configuring the `conf.ini` document

About this task

The `conf.ini` file has the following format:

```
[server]
# Hostname or IP address of the machine hosting the ML service
host = 127.0.0.1

# Port number for the ML service
port = 8000

[security]
# API security token for the ML service
api-token =

# To enable TLS/SSL deployment, set this to "yes"
use-ssl = no

# Absolute path to the TLS/SSL private key
ssl-keyfile = key.pem
```

```

# TLS/SSL keyfile password. If required, you can leave the password unspecified and provide the password during
server startup.
ssl-keyfile-password =

# Absolute path to the TLS/SSL certificate
ssl-certfile = cert.pem

[others]
# Number of requests retained in segment suggestion history for each user (for optimal experience, provide a
value between 5 and 20).
segmentation.request.history.max-limit = 10

# Maximum allowed CPU capacity for ML service. Valid values are 0 and 1, where 1 represents 100% capacity.
cpu.utilization.max-limit = 0.8

# Maximum allowed memory utilization for ML service. Valid values are 0 and 1, where 1 represents 100%
capacity.
memory.utilization.max-limit = 0.8

# Maximum number of CPU cores permitted for the ML service.
cpu.jobs.max-cores = 4

# Batch size for audience segmentation. Valid values are 16, 32, 64, 128, 256, and so on. For optimal
performance, keep the batch size low.
memory.segmentation.batch-size = 64

# The Clustering algorithm that is to be used for audience segmentation. Only "hdbscan" is supported.
segmentation.algorithm = hdbscan

[hdbscan]
# Default minimum segment size, in percentage, of the total filtered audiences (0.01 represents 1% of total
filtered audiences).
minimum-cluster-ratio = 0.0001

# Maximum number of workers for parallel segmentation processing. Performance is dependent on the allocated CPU
and memory.
max-workers = 8

algorithm = best
min-samples = 5
leaf-size = 40


```

To configure the `conf.ini` file, complete the following steps:


1. From the `Platform/ML/conf` folder, access the `conf.ini` file.
2. Configure the following fields in the `[server]` section of the file:

<code>host</code>	The host name or the IP address on which you want to run the ML service.
<code>port</code>	The port on which you want to run the ML service.

3. Configure the following fields in the `[security]` section of the file:

<p><code>api-token</code></p>	<p>The API security token referenced in this context is a prerequisite for clients accessing the ML service. It is imperative to configure this token consistently, ensuring that the same token is utilized as the datasource password in Configuration > Platform > ML > navigation. Verify that the tokens match precisely in both settings, as any discrepancies will result in an error.</p>
<p><code>use-ssl</code></p>	<p>If you are using SSL (i.e HTTPS), set it to <code>yes</code> and generate the keyfile and the certificate file (use a tool of your choice to generate the private key and the certificate file). If you set <code>use-ssl</code> to <code>yes</code>, you must configure <code>ssl-keyfile</code> and <code>ssl-certfile</code>. If you are not using SSL (i.e HTTP), set it to <code>no</code>. If you set <code>use-ssl</code> to <code>no</code>, ignore <code>ssl-keyfile</code> and <code>ssl-certfile</code>.</p> <p> Note: It is strongly recommended to operate ML service over a TLS/SSL connection to enhance the security of API access. If TLS/SSL is disabled, generate a symmetric encryption key and insert it into the designated files listed below:</p> <p>Under ML service - <code>conf/security/decryption.key</code></p> <p>Under Segment Central - <code>conf/security/encryption.key</code></p> <p>This extra measure ensures the minimum necessary security for the system.</p>
<p><code>ssl-keyfile</code></p>	<p>The location of private key file generated with its name and extension. Example: <code>D:\HCL\unica\key.pem</code>.</p>
<p><code>ssl-certfile</code></p>	<p>The location of the TLS certificate generated with its name and extension. Example <code>D:\HCL\unica\cert.pem</code>.</p>

4. Configure the following fields in the `[others]` section of the file:

<code>segmentation.request.history.max-limit = 10</code>	<p>This is the recent number of suggestion requests to be shown on the suggestion screen. Increasing it will show more suggestion history for the particular user.</p> <p> Note: Suggestions more than the suggested limit (either pending, successfully done, or failed) are removed from the system and cannot be recovered.</p>
<code>cpu.utilization.max-limit = 0.8</code>	<p>Specify the desired CPU capacity for the ML service by assigning a value within the range of 0 to 1, where 1 corresponds to 100% CPU utilization.</p>
<code>memory.utilization.max-limit = 0.8</code>	<p>Specify the desired memory utilization for the ML service by assigning a value within the range of 0 to 1, where 1 corresponds to 100% RAM utilization.</p> <p> Note: We recommend that you refrain from setting the memory limit to 1. Assigning 1 to this property results in the system utilizing the entire system memory, potentially causing disruptions to other processes, leading to improper functionality or halting altogether.</p>
<code>cpu.jobs.max-cores = 4</code>	<p>Specify the maximum number of CPU cores allocated to the ML service. Additionally, controls the overall capacity across all assigned cores by configuring cpu.utilization.max-limit. It is advisable to limit this number to half or a quarter of the total available cores to optimize performance and resource utilization.</p>
<code>memory.segmentation.batch-size=64</code>	<p>Specify the data partition size in megabytes (MB). It is advisable to maintain a lower value, such as 32, 64, 128, 256, 512, and so forth. This practice facilitates the smooth processing of large datasets within limited memory space. If you encounter memory errors in the logs, consider reducing the configured value accordingly.</p>
<code>segmentation.algorithm = hdbscan</code>	<p>Presently, only <code>hdbscan</code> is supported.</p>

5. Configure the following fields in the `[hdbscan]` section of the file:

<code>minimum-cluster-ratio = 0.0001</code>	The default minimum cluster size is applied if it is not explicitly specified while requesting the segment suggestions in Segment Central. When providing a ratio value, choose a lower value for larger audience datasets, as opting for a larger value may result in increased execution time while generating segment suggestions (The value 0.01 corresponds to 1% of the audience dataset size).
<code>max-workers = 8</code>	Number of workers for the algorithm to process. More the number of workers, faster the execution. Avoid assigning a high value.
<code>algorithm = best</code>	The best algorithm (from hdbscan internals) is selected automatically. Do not change this value.
<code>min-samples = 5</code>	Minimum number of samples needed for making a cluster point. Do not change this value as high values can lead to memory error.
<code>leaf-size = 40</code>	Do not change this value.

Configuring `encoding.ini` file

About this task

Audience data may include categorical information, where the order or ranking of categories is crucial for segmentation. For example, a dataset of audiences categorized into job positions like "Junior", "Intermediate", "Senior", and "Lead." In such cases, it is essential to create segments based on the specific ranking of job positions rather than using an arbitrary order. The ML service offers a way to specify these orders in the Platform/ML/conf/encoding.ini file. If the dataset is stored in a user database table named Audiences, with job positions recorded in the job_position column, configure the encoding.ini file as shown in the following example. The ML service considers these declarations when generating segment recommendations.

```
[Audiences]
job_position = Junior, Intermediate, Senior, Lead
```



Note: This declaration does not support categories having commas in their name.

This declaration can be expanded to include other columns in the Audiences table, as well as any other database table in the same way.

```
[Audiences]
job_position = Junior, Intermediate, Senior, Lead
another_column_in_audiences = category1, category2, category3, category4

[some_other_audience_table]
column_name = category1, category2, category3
```

Starting the ML services

1. Open the command prompt.
2. Change the directory to `Platform/ML`.
3. To launch the ML service, run the command `startup.bat`.


Deploying AI-ML in Unix-based Operating Systems

To deploy AI-ML component in RHEL and SuSE, complete the following procedures:

Configuring Python files

Before you begin

Even if you have `Python-3.10.10` installed on your system, you must install `Python-3.11.7`.

1. Access the `Python-3.11.7` folder within the `Platform/ML` directory.
2. Change the directory to `Python-3.11.7` and run the following command:
 - **RHEL:** `yum update -y && yum install -y xz gcc zlib-devel openssl-devel bzip2-devel libffi-devel sqlite-devel pip`
 - **SuSE:** `zypper update -y && yum install -y xz gcc zlib-devel openssl-devel bzip2-devel libffi-devel sqlite-devel pip`
3. Run the following commands in the same order:
 - `./configure LDFLAGS='-L/usr/lib64'`
 - `make clean`
 - `make install`
4.  **Note:** Skip this step if you are not using Oracle as your user database.

This step is relevant only when working with the data stored in Oracle database. To facilitate access to the Oracle database, it is necessary to install the appropriate Oracle client libraries for Python (`cx_Oracle`). For step-by-step instructions on the installation process, see the following official page:

https://cx-oracle.readthedocs.io/en/latest/user_guide/installation.html

5. In the file `Platform/ML/startup.sh`, at line number 1, replace the placeholder `[REPLACE_WITH_THE_ABSOLUTE_PYTHON_PATH]` with the absolute path to Python 3.11.7 version. For example `Platform/ML/Python-3.11.7`.
6. In the `Platform/ML/lib` directory, access the `site-packages` folder.
7. In the `Platform/ML/startup.sh` file, replace the placeholder `[REPLACE_WITH_THE_SITEPACKAGES_PATH]` with the absolute path to `site-packages` directory.

Configuring the `conf.ini` document

About this task

This is similar to the configuration done on Microsoft Windows. See [Configuring the `conf.ini` document on page 62](#).

Configuring `encoding.ini` file

About this task

This is similar to the configuration done on Microsoft Windows. See [Configuring `encoding.ini` file on page 66](#).

Starting the ML services

1. Change the directory to Platform/ML directory.
2. To launch the ML service, run `startup.sh`.

Chapter 8. Configuring Unica Platform after deployment

For a basic installation of Unica Platform, you must configure Unica Platform after deployment if you use the Unica reporting feature, or if you want to use a password policy.

If you use the Unica reporting feature, see the *Unica Reports Installation and Configuration Guide*. If you want to use a password policy in mind, see [Default password settings on page 69](#) to determine whether you must change the default password settings.

The Unica Platform has additional properties on the **Configuration** page that perform important functions that you can optionally adjust. See the context help for the properties, or the *Unica Platform Administrator's Guide* to learn more about what they do and how to set them.

Default password settings

Unica provides default settings for using passwords. However, you can use the **General > Password settings** category on the Unica **Configuration** page to change the default settings, and create your password policies.

The default password settings apply to passwords for users that are created within Unica. The settings do not apply to the users that are imported through synchronization with an external system, such as Windows™ Active Directory, a supported LDAP directory server, or web access control server. The exception is the `Maximum failed login attempts allowed` setting, which affects both internal and external users. Also note that this property does not override any similar restriction set in an external system.

The following settings are the default password settings that Unica:

- `Maximum failed login attempts allowed` - 3
- `Password history count` - 0
- `Validity (in days)` - 30
- `Blank passwords allowed` - True
- `Allow identical user name and password` - True
- `Minimum number of numeric characters` - 0
- `Minimum number of letter characters` - 0
- `Minimum character length` - 4

See the online help for descriptions of the default settings.

Set the web application session timeout (optional)

The session timeout determines how long an inactive HTTP session can remain open before it expires.

Set the session timeout on your web application server:

- **WebSphere®:** Use the IBM 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.

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.disablePoweredBy` 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.

Disabling caching of Unica Campaign and Unica Platform configuration property values

If you encounter an issue where configuration properties are not being picked up by Unica Campaign or Unica Platform, you can disable caching as described in this procedure.

About this task

Set generic JVM arguments in the web application server where Unica Platform is deployed, as described in the following steps.

1. Set the following JVM argument, which disables the Hibernate L2 cache.

```
-DUNICA_PLATFORM_CACHE_ENABLED=false
```

By default, this value is set to `true`, which provides the best performance.

2. Set the following JVM argument, which enables local caching. This improves performance when the Hibernate cache is disabled.

```
-DUNICA_PLATFORM_LOCAL_CACHE_ENABLED=true
```

When this local caching mechanism is enabled, configurations are updated at 60 minute intervals. If you want configuration changes to be picked up immediately, you must restart the application server.

Enabling single-user logging

About this task

You can enable single-user logging by configuring logging to use the XML file and then editing the XML file.

Logging is configured using one of two files: `log4j.properties` or `log4j.xml`. By default, the `log4j.properties` file is used.

You can enable per-user logging by configuring logging to use the XML file and then editing the XML file. If Unica Platform is configured in a cluster deployment, copy the XML file to each node.

- You can enable single-user logging by editing the XML file.
- Logging is configured using `log4j.xml`, which is the default configuration file.
- If Unica Platform is configured in a cluster deployment, copy the XML file to each node.

With XML logging enabled, a thread is created that periodically checks if the XML configuration file has been created or modified. If a change or file creation is detected, then the XML file is read to configure `log4j`. The polling interval is 60 seconds.

1. Configure logging to use `log4j.xml` by setting the following JVM parameter.

```
-DENABLE_PLATFORM_LOG4J_XML_LOGGING=true
```

The value must be set to `true` to enable per-user logging.

If Unica Platform is configured in a cluster deployment, set this JVM parameter in each node of the cluster.

2. To specify the user account to be logged in per-user logging, edit the `log4j.xml` file and add the users in the filter tag. The logs for the users that are added in the filter tag.
 - You can add multiple tags in the `log4j.xml` file to create separate log files for specific users. You must add a new appender for each new user specific log file.
 - By default, the log file is created in the `Platform_Home /Platform/logs` folder and is named as `platform.log`. You can specify a different valid path and file name. You must specify the absolute or complete path to generate the log files in the respective folders.
 - If both user specific logs and logs for all users are required, add an appender tag with a new name and without the filter tag defined. The appender must have a unique name.
 - Add a corresponding entry under the root tag for this new appender.
3. If Unica Platform is configured in a cluster deployment, copy the edited XML file to each node of the cluster.

You can use a command like the one shown in the following example.

```
-DPLATFORM_LOG4J_XML_FILE=log4j_node1.xml
```

The `log4j_node1.xml` file is a copy of the `log4j.xml` file. You can use any name for the copied file. Log file is also created with this new name like `log4j_node1.log` automatically instead of default name `platform.log`.

Consider the following example where the logs are collected for the user `asm_admin` and for all other users.

```
<appender name="Console" class="org.apache.log4j.ConsoleAppender">
<param name="ImmediateFlush" value="true"/>
<layout class="org.apache.log4j.PatternLayout">
<param name="ConversionPattern" value="%-5p %c - %m%n"/>
</layout>
<filter class="com.unica.manager.logger.UserMatchFilter">
<param name="StringToMatch" value="asm_admin" />
</filter>
</appender>
<appender name="Console" class="org.apache.log4j.ConsoleAppender">
<param name="ImmediateFlush" value="true"/>
<layout class="org.apache.log4j.PatternLayout">
```



```

<param name="ConversionPattern" value="%-5p %c - %m%n"/>
</layout>
<filter class="com.unica.manager.logger.UserMatchFilter">
<param name="StringToMatch" value="asm_admin" />
</filter>
</appender>
</appender>
<!-- <logger name="com.unica.manager.configuration.ConfigurationManager">
<level value="TRACE"/>
</logger>
<logger name="com.unica.suite.scheduler.server.manager.TaskManager">
<level value="DEBUG"/>
</logger>
<logger name="org.hibernate.util.JDBCExceptionReporter">
<level value="ERROR"/>
</logger>
-->
<root>
<level value="WARN"/>
<appender-ref ref="System"/>
<appender-ref ref="Console"/>
<appender-ref ref="SystemAllUsers"/>
</root>

```

1. To specify the user account to be logged in per-user logging, edit the `log4j.xml` file and uncomment `RollingFile` tag with name `UserLogAppender`. Add the `userid` in the filter tag. The logs for the user that is added in the filter tag are saved in the file that is mentioned in that appender. Set below JVM parameter if not already set,

```
-DUNICA_PLATFORM_HOME= <platform_home_directory_path>
```

- You can add multiple tags in the `log4j.xml` file to create separate log files for specific users. You must add a new appender for each new user specific log file.
- By default, the log file is created in the `Platform_Home /Platform/logs` folder and is named as `platform.log`. You can specify a different valid path and file name. You must specify the absolute or complete path to generate the log files in the respective folders.
- If both user specific logs and logs for all users are required, add an appender tag with a new name and without the filter tag defined. The appender must have a unique name.
- Add a corresponding entry under the root tag for this new appender.

2. If Unica Platform is configured in a cluster deployment, copy the edited XML file to each node of the cluster.

You can use a command like the one shown in the following example.

```
-DPLATFORM_LOG4J_XML_FILE=log4j_node1.xml
```

The `log4j_node1.xml` file is a copy of the `log4j.xml` file. You can use any name for the copied file. Log file is also created with this new name like `log4j_node1.log` automatically instead of default name `platform.log`.

Example

Consider the following example where the logs are collected for the user `asm_admin` and for all other users.

```

<?xml version="1.0" encoding="UTF-8"?>
<Configuration packages="com.unica.manager.logger" monitorInterval="60">
  <Appenders>

    <!-- Console Log Appender -->

    <Console name="CONSOLE_LOG" target="SYSTEM_OUT" immediateFlush="true">
      <PatternLayout pattern="%-5p %c - %m%n"/>
    </Console>

    <!-- System Log Appender -->
    <!-- The following section is for logs for all the user-->
    <RollingFile name="SYS_LOG" fileName="${sys:UNICA_PLATFORM_LOG_FILE}"
filePattern="${sys:UNICA_PLATFORM_LOG_FILE}_%d{yyyy-MM-dd}-%i"
immediateFlush="true" append="true" >
      <PatternLayout pattern="%d{DATE} - %-5p - %m%n" />
    <Policies>
      <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
      <SizeBasedTriggeringPolicy size="10 MB" />
    </Policies>
    <DefaultRolloverStrategy max="10"/>
  </RollingFile>

    <!-- The following section is for user specific logs for the user asm_admin-->

    <RollingFile name="UserLogAppender" fileName="${sys:UNICA_PLATFORM_HOME}/logs/asm_admin.log"
filePattern="${sys:UNICA_PLATFORM_HOME}/logs/asm_admin.log_%d{yyyy-MM-dd}"
immediateFlush="true" append="true" >
      <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss} [%X{user}] %-5p %F.%M:%L: %m%n" />
    <Policies>
      <SizeBasedTriggeringPolicy size="10 MB" />
    </Policies>
    <DefaultRolloverStrategy max="10"/>
    <UserMatchFilter user="asm_admin" onMatch="ACCEPT" onMismatch="NEUTRAL"/>
  </RollingFile>

  </Appenders>
  <Loggers>

  <Root level="WARN" includeLocation="true">
    <AppenderRef ref="SYS_LOG"/>
    <AppenderRef ref="CONSOLE_LOG"/>
    <!-- <AppenderRef ref="UserLogAppender"/> -->
  </Root>

  <!-- <Logger name="com.unicacorp" level="INFO"> -->
  <!-- <AppenderRef ref="UserLogAppender"/> -->
  <!-- </Logger> -->

  <!-- <Logger name="com.unica" level="INFO"> -->
  <!-- <AppenderRef ref="UserLogAppender"/> -->
  <!-- </Logger> -->

```

```
</Loggers>
</Configuration>
```

Collecting installations logs

When you install any Marketing Software product, the installation logs for that particular installation are bundled and exported. Use the Installer utility to collect the logs for any of the Marketing Software product that are installed.



Note: When you install any Marketing Software product, the installation logs are bundled and exported to the `Installer_logs` folder in the installation directory.

The installer utility

To collect the installation logs for any of the installed Marketing Software products, complete the following steps:

1. Update the `Logs_properties.xml` file with the installation directory of the products for which you want to collect installation logs.
2. Run the installer from the command line with the following command line argument: `-DUNICA_EXPORT_LOGS=1`
`-DUSER_INSTALL_DIR=<<Suite Installation Dir>>`
3. In the **Zip Installation logs** pane, select **Yes** to zip the installation logs.
4. Select the products for which the log files are to be exported.

The installation logs are bundled and saved in the `Installer_Logs` folder in the User Installation directory.

```
[Logs<<timestamp>> , Logs<<timestamp>>.zip]
```

Chapter 9. Installing Unica Platform in a cluster

Unica Platform supports cluster deployment for IBM WebSphere Application Server, JBOSS, and Tomcat. Clustering is not supported on WebLogic.

Process overview

The following steps are an overview of the process for setting up a Unica Platform cluster deployment.

1. Set up WebSphere clusters on two or more servers.
2. Create a profile for the primary node in WebSphere.
3. Create WebSphere cluster nodes.
4. Create a WebSphere cluster
5. Add existing nodes to the cluster (optional)
6. Start the nodes, web server, and cluster
7. Create JDBC providers
8. Install and deploy Unica products
9. Configure the HTTP Server as a load balancer
10. Configure Unica Platform logs.

Prerequisites

The following are prerequisites that must be met before you perform the procedures described in this section.

- The HTTP Server and the HTTP Server plugins must be installed on the server machine on which you plan to create the WebSphere Deployment Manager profile is being created. The HTTP Server is included with WebSphere at no charge.
- The HTTP Server is up and running.

Creating a WebSphere profile for the primary node of a cluster

Follow these guidelines to create a WebSphere profile for the primary node of a Unica Platform cluster.

About this task

Perform the following steps in the WebSphere profile creation wizard.

See the following video for a demonstration: https://www.youtube.com/watch?v=RREp9n2Yy_o

1. Select **Cell (Deployment manager and a federated application server)** as the type.
2. On the Web Server Definition Part 1 screen, do the following.
 - a. Select **Create a Web Server definition**.
 - b. Select **HCL HTTP Server** as the web server type.

- c. Enter the web server name.

In a later step you create a web server in the cluster using this name.

Examples in this guide use `IMPWebServer` as the web server name.

- d. Enter the web server host and port.

3. On the Web Server Definition Part 2 screen, do the following.

- a. Web server installation directory path
- b. Web server plug-in installation directory path

Creating profiles for additional nodes

Follow these guidelines to create additional nodes for a Unica Platform cluster.

About this task

Create WebSphere nodes on multiple servers. You can create the node profiles using the Profile Management tool or the `manageprofile` command.

Adding existing nodes to the cluster nodes

Optionally, you can add your existing WebSphere profiles as nodes in the cluster environment.

About this task

Follow these guidelines to add existing nodes to the cluster.

1. Start the Websphere Deployment Manager as follows.
 - a. Log in to the server where the WebSphere Deployment Manager is installed.
 - b. Navigate to the Deployment Manager profile location

Example location: `/data/webservers/IBM/WASND85/profiles/Dmgr01/bin`

- c. Execute the `startmanager` script.

2. Add the WebSphere profile node to the cluster as follows.

- a. Log in to server where the WebSphere profile is created.
- b. Navigate to the WebSphere profile location.

Example location: `/data/webservers/IBM/WASND85/profiles/AppSrv01/bin`

- c. Execute command: `./addNode.sh DEPLOYMENT_MANAGER_HOST
DEPLOYMENT_MANAGER_SOAP_PORT`

For example: `./addNode.sh example.com 8879`

You can obtain the SOAP Port can be from the Deployment Manager – AboutThisProfile.txt.

Creating a WebSphere cluster

Use the WebSphere Deployment Manager profile to create the cluster.

About this task

For information, see the following video: <https://www.youtube.com/watch?v=wK06h2kVq5w>

Creating the web server in the cluster

The HTTP Server can be configured to connect with the nodes in the cluster and serve as a load balancer.

About this task

Use the WebSphere Deployment Manager profile to create the web server in the cluster.

For information, see the following video: <https://www.youtube.com/watch?v=rfOFBRX7Kjc>

Give your web server the name you used in the step "Creating a WebSphere profile for the primary node of a cluster."

Examples in this guide use `IMPWebServer` as the web server name.

Starting the nodes, web server, and cluster

Start the nodes, HTTP Server, and cluster, in that order.

About this task

Before you start the WebSphere cluster from WebSphere console, start all of the nodes you created in previous steps, and start the HTTP Server. The HTTP Server is included with WebSphere at no charge.

1. Start the nodes on each server by navigating to the appropriate profile under `data/webservers/IBM/WASND85/profiles` under your WebSphere installation and executing the `startNode` script.
2. Start the HTTP Server as follows.

- a. Open the HTTP Server `conf` file and verify that the HTTP Server listen port is 80.

The `conf` file is located in the `conf` directory under your HTTP Server installation. The typical install location is `data/webservers/IBM/IHS`.

Look for `Listen 80` in the file.

- b. To start the server, navigate to the `data/webservers/IBM/IHS/bin` directory under your HTTP Server installation and execute this command: `./apachectl start`.

- a. Verify that the HTTP Server is started successfully by launching the URL of the server.

3. Start the cluster from WebSphere Deployment Manager as follows.

- a. Navigate to **Servers > Clusters > WebSphere application server cluster**.
- b. Select your cluster and click **Start**.

If the cluster does not start, do the following.

:

- Verify that your cluster nodes are running on each server.
- Verify that the application servers are started. You can navigate to cluster members and start each application server manually.

Creating JDBC providers in a cluster

Creating JDBC providers is slightly different in a clustered environment than in a non-clustered environment.

About this task

See the following video for information: <https://www.youtube.com/watch?v=fGYgwNvSC7c>

Installing Unica products in a cluster

Follow the installation guide for your Unica product to install and deploy the product, with the additional configuration described in this procedure.

About this task

- Install your product on the primary node, following the instructions in the product Installation Guide.
- During the installation, provide the port of the cluster manager machine as the HTTP port for the application.

If you do not provide this port, you must change the value of the product's navigation URL configuration property on the **Settings > Configuration** page.

- Share or mount the product installation directory from the primary machine to all node machines. For AIX/Linux you can mount the installation drive.
- Deploy the product web application on the WebSphere cluster.

Configuring the HTTP Server as a load balancer

Follow these steps to configure the HTTP Server as a load balancer with Unica Platform.

1. In the WebSphere administrative console, navigate to **Servers > Server Types > Web Servers**.
2. Select the server name and click **Generate Plug-in**. Then select the server name and click **Propagate Plug-in**.

Verify that the `plugin-cfg.xml` file is generated. Note the path of this file.

Open the file and verify that it contains the correct URIGroups and cluster server information. If the information is not correct, modify the file and then repeat the steps in this procedure.

3. Open the HTTP Server `conf` file and add the following two lines to the bottom of the file.

The `conf` file is located in the `conf` directory under your HTTP Server installation. The typical install location is `data/websevers/IBM/IHS`.

```
LoadModule was_ap22_module
    IBM HTTP SERVER HOME/Plugins/bin/64bits/mod_was_ap22_http.so
WebSpherePluginConfig
    IBM HTTP SERVER HOME/config/vmw009113022209/plugin-cfg.xml
```

4. Restart the cluster.

Configure property to control the Scheduler in a clustered environment

In a clustered environment, the various instances of the IBM EMM Scheduler provided by Unica Platform all run scheduling tasks, which can result in conflicts. To prevent the conflict, disable the Scheduler on all nodes except one.

About this task

In the WebSphere administration console, set the `DISABLE_SCHEDULER_CLIENT_ON_CURRENT_NODE` JVM parameter with the `-D` option in all the nodes except one. This disables the Scheduler on the application server clustered node.

For example: `-DDISABLE_SCHEDULER_CLIENT_ON_CURRENT_NODE=true`.

When you configure this JVM parameter on all nodes except one, then all the scheduled jobs are run on the application server cluster node where this parameter is not set.



Important: If the node where the Scheduler is enabled goes down, no scheduled tasks run.

Configuring Unica Platform logs in a cluster

You can generate logs for each node where you deploy Unica Platform. You can specify different logging levels for each node in your cluster.

You have two options for generating Unica Platform logs for each node in the cluster:

- Sharing the location where Unica Platform is installed.
- Setting Unica Platform home in a Java parameter.

Both methods are described in this section.

Sharing the location where Unica Platform is installed

Use this procedure to share the Unica Platform installation location on all nodes of the cluster.

Before you begin

To share the location, you must install Unica Platform on a shared drive that is accessible from all nodes.

1. In the WebSphere administration console, add the `-DPLATFORM_LOG4J_PROPERTIES_FILE` Java parameter in each node.

For example: `-DPLATFORM_LOG4J_PROPERTIES_FILE=log4j_node1.properties`

where `log4j_node1.properties` is a copy of the `log4j.properties` file.



Note: If XML logging is configured, then in the WebSphere administration console, add the `-DPLATFORM_LOG4J_XML_FILE` Java parameter in each node.

For example: `-DPLATFORM_LOG4J_XML_FILE=log4j_node1.xml` where `log4j_node1.xml` is a copy of the `log4j.xml` file.

2. Use the following command to set the log file name in each node. Ensure that the log file names are different so that it is easy to identify the file that is generated from each node.

In the WebSphere administration console, add `-Dlog4j.appender.System.File=Log_File_Name`

For example: `-Dlog4j.appender.System.File=platform_node1.log`.



Note: This property can be used in case of XML logging or properties file logging.

3. Restart the cluster.

All the log files are created in the `Platform/logs` directory under your Unica Platform installation.

Setting Unica Platform home in a Java parameter

Use this procedure to set the Unica Platform home directory in a Java parameter that points to the location where the logs are generated.

1. Use the following command to specify the Java parameter in the WebSphere administration console.

`-DUNICA_PLATFORM_HOME=path_where_log_files_are_placed`

For example: `UNICA_PLATFORM_HOME=/opt/Platform`

2. Create the following directories at the location you specified above: `conf` and `logs`
3. Provide the write permission to the `logs` directory.
4. Copy the `log4j.properties` file to the `conf` directory.

The `log4j.properties` is located in the `conf` directory under the Unica Platform installation directory.

5. Replicate the Unica Platform installation directory structure on all nodes of the cluster, as follows.

- a. Create a same directory structure on each node as far as the `PLATFORM_HOME/Platform/conf` directory.
 - b. Create the `logs` directory in the `PLATFORM_HOME/Platform` directory and provide write permission to the `logs` directory.
 - c. Copy the `log4j.properties` file in the `conf` directory.
6. Restart the cluster.

Deployment of Unica Platform on JBOSS cluster

To install Unica Campaign on Jboss cluster, see the Campaign Install Guide.

Preparing the data sources

You must add or create JNDI in the same JBOSS domain which you are planning to use in a cluster. Complete the following step to create JNDI.

1. 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"
```

JNDIs 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" in the Campaign Installation Guide, complete the following steps.

1. Launch JBOSS management console.
2. Navigate to **Deployments** tab. Select the **ServerGroup**.
3. Under **Deployments** section, click **Add**. A pop-up window appears.
4. Browse the WAR/EAR application and complete the deployment process.

Installing Unica Platform in a Tomcat cluster

Unica Platform supports an IBM WebSphere Application Server cluster deployment.

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.

Prerequisites

The following are the prerequisites.

- Tomcat server (server1, server2, server3, ..., etc.)
- Tomcat 1: Unica Platform and Campaign (web and listener)
- Tomcat 2: Unica Platform
- Tomcat 3: Unica Platform
- Nginx load balancer (server4)
- Client browser to access Unica application

Architecture

Perform the following steps.

1. Tomcat Installation - server1: Download and install Tomcat server and listener on 8080.
2. Tomcat Installation - server2: Download and install Tomcat server and listener on 8081.
3. Tomcat Installation - server3: Download and install Tomcat server and listener on 8082.
4. Tomcat server1:8080: Install and deploy both Unica Platform and Campaign and make it work with 8080 port.
5. Tomcat server2:8081: Install and deploy Unica Platform only and make it work with 8081 port.
6. Tomcat server3:8082: Install and deploy Unica Platform only and make it work with 8082 port.

Install Nginx

As an example, perform the following steps to download and install Nginx for Linux.

- `wget http://nginx.org/download/nginx-1.2.6.tar.gz`
- `tar -xzf nginx-1.2.6.tar.gz`
- `cd nginx-1.2.6/`
- `./configure --prefix=/opt/Tomcat/nginx` (example: `./configure --prefix=/opt/Tomcat/nginx --without-http_gzip_module --with-http_stub_status_module`)
- `Make`
- `Make install`

Configure Nginx

Perform the following steps to configure Nginx.

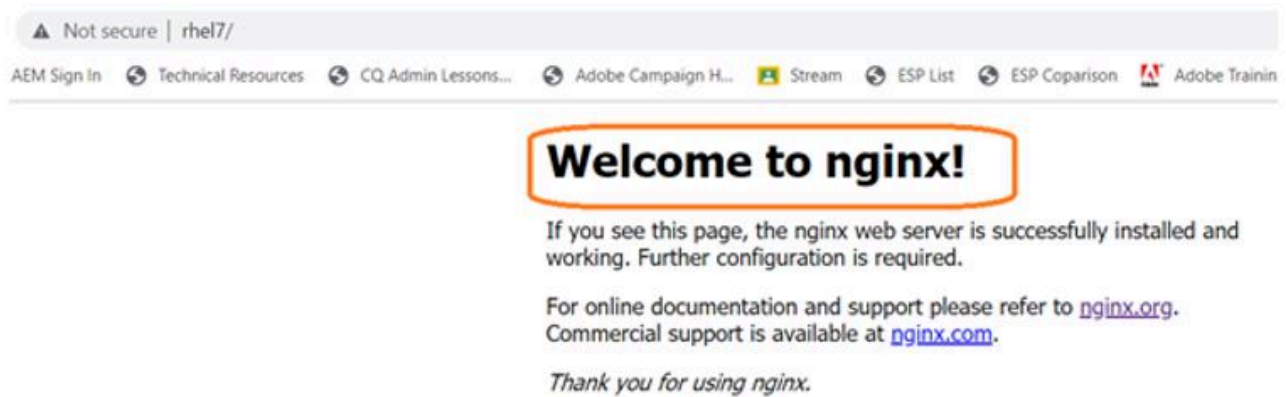
1. Update `<NGINX_HOME>/conf/nginx.conf`.
2. To start and stop nginx, execute the following commands.

- `<NGINX_HOME>/sbin/nginx`
- `<NGINX_HOME>/sbin/nginx -s stop`

3. Nginx Logs directory: `<NGINX_HOME>/logs/`

Verify Nginx

In this example, nginx is listening on 80 port, hence my URL is <http://rhel7:80>



With the above screen, you can verify that nginx is running without any issues.

Nginx load balancer algorithms

- Round-robin: Requests to the application servers are distributed in a round-robin fashion. The following is the sample configuration (<NGINX_HOME>/conf/ nginx.conf) with round robin fashion.

```
#gzip on;

upstream tomcatcluster {
    server1:8080;
    server2:8081;
    server3:8082;
}

server {
    listen      80;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        proxy_pass http://tomcatcluster;
        #root    html;
        # index  index.html index.htm;
    }

    #error_page 404          /404.html;
```

- Ip_hash: With ip-hash, the client's IP address is used as a hashing key to determine the server to be selected in a server group for the client's requests.

This method ensures that the requests from the same client are always directed to the same server except when this server is unavailable.

The following is the sample configuration (<NGINX_HOME>/conf/nginx.conf) with round robin fashion.

```

#gzip on;

upstream tomcatcluster {
    ip_hash();
    server1:8080;
    server2:8081;
    server3:8082;
}

server {
    listen      80;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        proxy_pass http://tomcatcluster;
        #root html;
        # index index.html index.htm;
    }
}

```

- Least_conn: With the least-connected load balancing, nginx tries not to overload a busy application server with excessive requests, distributing the new requests to a less busy server instead.
- Weight Distribution (weight n): With this configuration, every five new requests are distributed across the application instances as the following:
 - Three requests are directed to srv1.
 - One request goes to srv2.
 - Another one goes to srv3.

For example:

Upstream tomcatcluster

```

{
    Server server1:8080 weight 3;
    Server server2:8081;
    Server server4:8082;
}

```

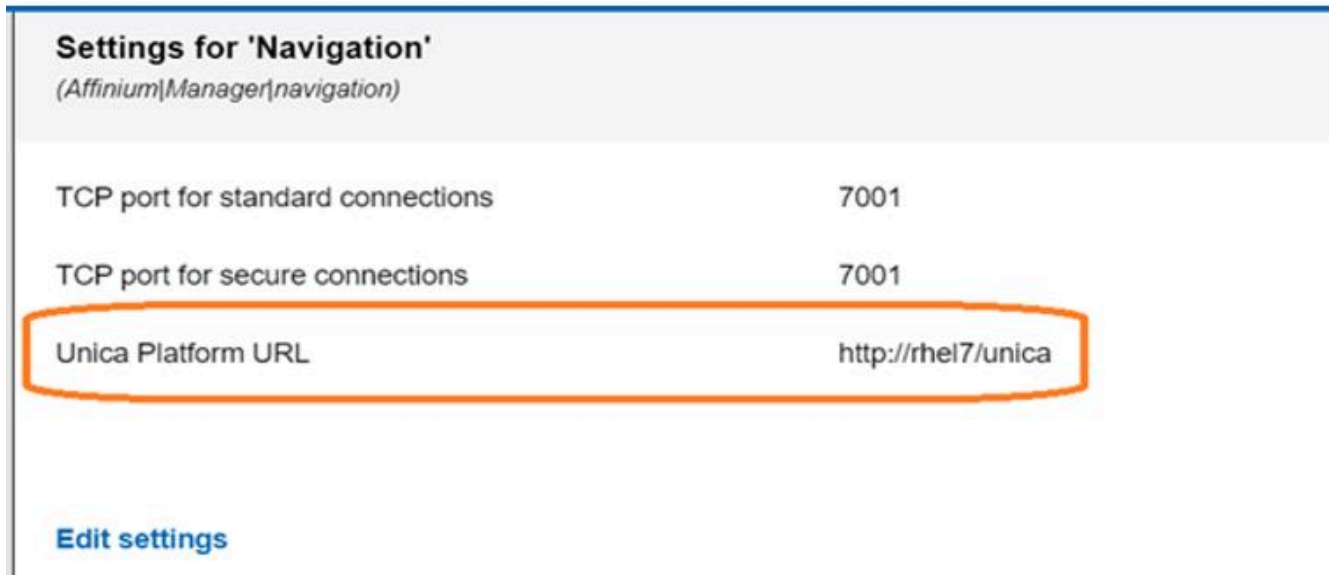
- Max Fails configuration: Max fails refers to the maximum number of failed attempts to connect to a server must occur before it is considered inactive. Fall_timeout specifies the length of that the server is considered inoperative. Once the time expires, new attempts to reach the server will start up again. The default timeout value is 10 seconds.
- For example:

Upstream tomcatcluster

```
{  
    Server server1:8080 max_fails=3 fail_timeout=15s;  
    Server server2:8081;  
    Server server4:8082;  
}
```

Configure Unica with Nginx

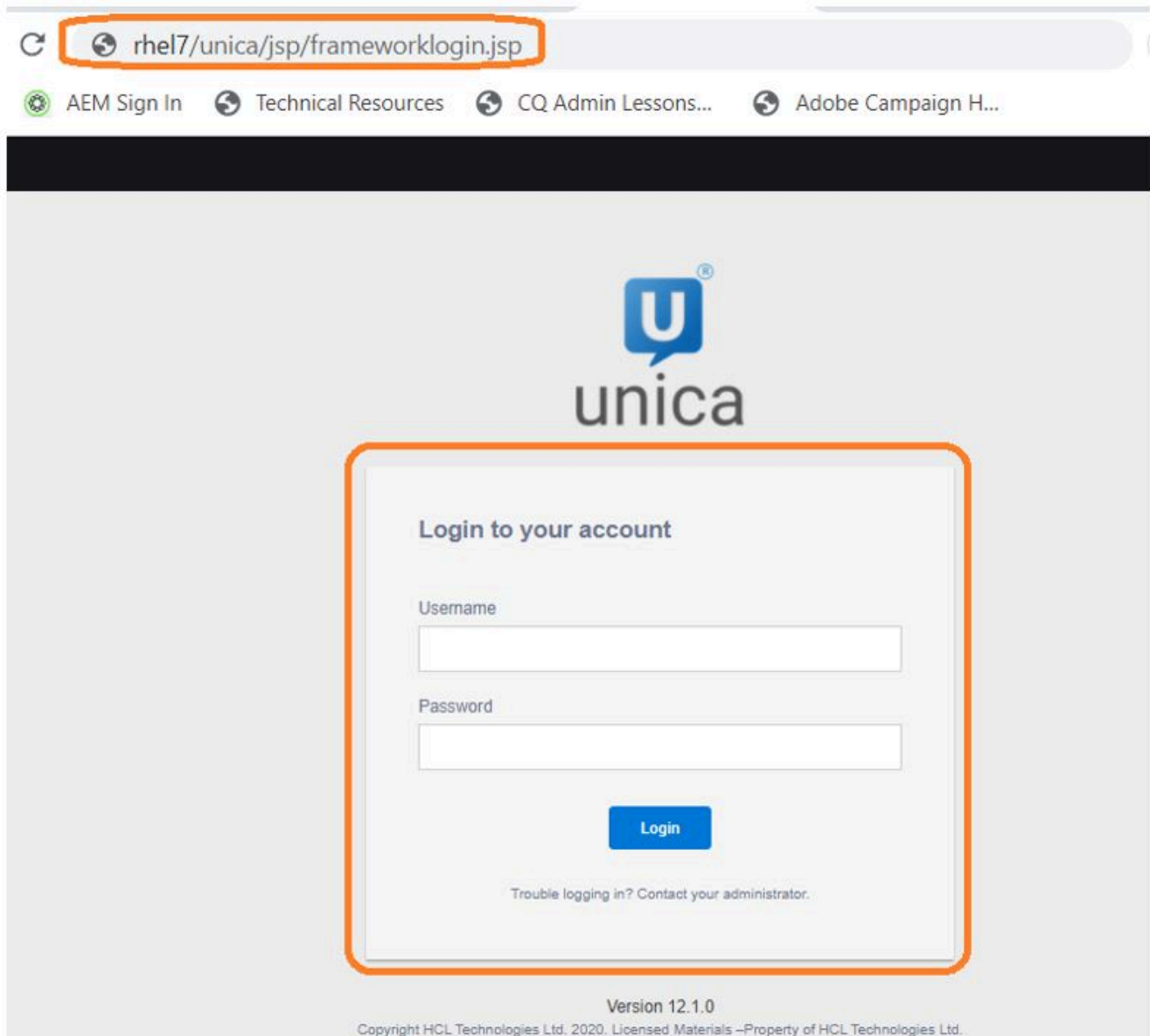
Change the Platform navigation URL with Nginx server host and actual context.



Settings for 'Navigation'
(Affinium\Manager\navigation)

TCP port for standard connections	7001
TCP port for secure connections	7001
Unica Platform URL	http://rhel7/unica

[Edit settings](#)



Reference URL

- Tomcat : <https://tomcat.apache.org/download-90.cgi>
- Nginx: <http://nginx.org/>

Chapter 10. 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 on page 97](#) - generates a key for a client application like Unica Journey to authenticate with one Unica Platform instance.
- [alertConfigTool on page 92](#) - registers alerts and configurations for Unica products
- [configTool on page 92](#) - imports, exports, and deletes configuration settings, including product registrations
- [datafilteringScriptTool on page 97](#) - creates data filters
- [encryptPasswords on page 99](#) - encrypts and stores passwords
- [encryptTomcatDBPasswords on page 100](#) - encrypt the database passwords that the Tomcat Application server uses internally
- [partitionTool on page 101](#) - creates database entries for partitions
- [populateDb on page 103](#) - populates the Unica Platform database
- [quartzjobtool on page 107](#) - Update scheduler jobs created in version 11.1 and older versions
- [restoreAccess on page 104](#) - restores a user with the platformAdminRole role
- [scheduler_console_client on page 105](#) - 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.

Setting up Unica Platform utilities on additional machines

On the machine where the Unica Platform is installed, you can run the Unica Platform utilities without any additional configuration. However, you might want to run the utilities from another machine on the network. This procedure describes the steps required to do this.

Before you begin

Ensure that the machine on which you perform this procedure meets the following prerequisites.

- The correct JDBC driver must exist on the machine or be accessible from it.
- The machine must have network access to the Unica Platform system tables.
- The Java™ runtime environment must be installed on the machine or be accessible from it.

1. Gather the following information about the Unica Platform system tables.
 - The fully qualified path for the JDBC driver file or files on your system.
 - The fully qualified path to an installation of the Java™ runtime environment.

The default value in the installer is the path to the supported version of the JRE that the installer places under your Unica installation directory. You can accept this default or specify a different path.

- Database type
 - Database host
 - Database port
 - Database name/system ID
 - Database user name
 - Database password
2. Run the Unica installer and install the Unica Platform.

Enter the database connection information that you gathered for the Unica Platform system tables. If you are not familiar with the Unica installer, see the Unica Campaign or Unica Plan installation guide.

You do not have to deploy the Unica Platform web application if you are installing the utilities only.

Utilities

This section describes the Unica Platform utilities, with functional details, syntax, and examples.

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 16. Product names for configTool registration and unregistration

Product name	Name used in configTool
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"
```

Important:

In case of full Affinium re-import we need to take below steps:

```
./configTool.sh -x -p 'Affinium' -f full_export.xml
```

```
./configTool.sh -d -p "Affinium" -o
```




```
./configTool -i -o -p "Affinium" -f full_export.xml
```

For child node:

```
./configTool.sh -i -p "Affinium|Manager" -f allcfg.xml -o
```

Clientdetails

This utility generates keys for client application, like Unica Journey to authenticate with one Platform instance.

It registers the key in Platform database and prints it on the console. The key can then be copied and pasted on the target application.

Syntax

```
clientDetails -a appName
```

Commands

```
-a appName
```

Generate the key for the specified application. Possible values for appName is Manager (for Unica Platform) and Journey (for Unica Journey)

Examples

Generate the key for Unica Journey

```
clientDetails -a Journey
```

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 Platform 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`
 - Specify 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 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 loginName

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
```

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

Chapter 11. Unica Platform SQL scripts

This section describes the SQL scripts provided with Unica Platform to perform various tasks relating to the Unica Platform system tables.

The Unica Platform SQL scripts are located in the `db` directory under your Unica Platform installation.

The scripts are designed to be run against the Unica Platform system tables, using the database client.

ManagerSchema_DeleteAll.sql

The `ManagerSchema_DeleteAll.sql` script removes all data from the Unica Platform system tables without removing the tables themselves. This script removes all users, groups, security credentials, data filters, and configuration settings from Unica Platform.

When to use ManagerSchema_DeleteAll.sql

You might want to use `ManagerSchema_DeleteAll.sql` if corrupted data prevents you from using an instance of Unica Platform.

Additional requirements

To make Unica Platform operational after running `ManagerSchema_DeleteAll.sql`, you must perform the following steps.

- Run the `populateDB` utility. The `populateDB` utility restores the default configuration properties, users, roles, and groups, but does not restore any users, roles, and groups you have created or imported after initial installation.
- Use the `configTool` utility with the `config_navigation.xml` file to import menu items.
- If you have performed any post-installation configuration, such as creating data filters or integrating with an LDAP server or web access control platform, you must perform these configurations again.
- If you want to restore previously existing data filters, run the `datafilteringScriptTool` utility using the XML originally created to specify the data filters.

ManagerSchema_PurgeDataFiltering.sql

The `ManagerSchema_PurgeDataFiltering.sql` script removes all data filtering data from the Unica Platform system tables without removing the data filter tables themselves. This script removes all data filters, data filter configurations, audiences, and data filter assignments from Unica Platform.

When to use ManagerSchema_PurgeDataFiltering.sql

You might want to use `ManagerSchema_PurgeDataFiltering.sql` if you need to remove all data filters without removing other data in the Unica Platform system tables.



Important: The `ManagerSchema_PurgeDataFiltering.sql` script does not reset the values of the two data filter properties, `Default table name` and `Default audience name`. If these values are no longer valid for the data filters you want to use, you must set the values manually on the Configuration page.

SQL scripts for creating system tables

Use the scripts described in the following table to create Unica Platform system tables manually when your company policy does not allow you to use the installer to create them automatically.

The scripts are shown in the order in which you must run them.

Table 17. Scripts for creating system tables

Datasource Type	Script Names
IBM® DB2®	<ul style="list-style-type: none"> • <code>ManagerSchema_DB2.sql</code> <p>If you plan to support multi-byte characters (for example, Chinese, Japanese, or Korean), use the <code>ManagerSchema_DB2_unicode.sql</code> script.</p> <ul style="list-style-type: none"> • <code>ManagerSchema__DB2_CeateFKConstraints.sql</code> • <code>active_portlets.sql</code> • <code>notification_rules.sql</code>
Microsoft™ SQL Server	<ul style="list-style-type: none"> • <code>ManagerSchema_SqlServer.sql</code> • <code>ManagerSchema__SqlServer_CeateFKConstraints.sql</code> • <code>active_portlets.sql</code> • <code>notification_rules.sql</code>
MariaDB	<ul style="list-style-type: none"> • <code>ManagerSchema_MariaDB.sql</code> • <code>ManagerSchema_MariaDB_StoredProcedures.sql</code> • <code>ManagerSchema_MariaDB_CreateFKConstraints.sql</code> • <code>active_portlets.sql</code> • <code>notification_rules.sql</code>
Oracle	<ul style="list-style-type: none"> • <code>ManagerSchema_Oracle.sql</code> • <code>ManagerSchema__Oracle_CeateFKConstraints.sql</code> • <code>active_portlets.sql</code> • <code>notification_rules_Oracle.sql</code>

Table 17. Scripts for creating system tables (continued)

Datasource Type	Script Names
PostgreSQL	<ul style="list-style-type: none"> • <code>ManagerSchema_PostgreSql.sql</code> • <code>ManagerSchema_PostgreSql_CreateFKConstraints.sql</code> • <code>ManagerSchema_pgsql_upg.sql</code> • <code>ManagerSchema_PostgreSQL_StoredProcedures.sql</code> • <code>ContentIntegration_pgsql.sql</code> • <code>active_portlets_PostgreSQL.sql</code>

If you plan to use the scheduler feature that enables you to configure a flowchart to run at predefined intervals, you must also create the tables that support this feature. To create the scheduler tables, run the appropriate script, as described in the following table.

Table 18. Scripts for enabling the Unica Scheduler

Data Source Type	Script Name
DB2®	<code>quartz_db2.sql</code>
Microsoft™ SQL Server	<code>quartz_sqlServer.sql</code>
Oracle	<code>quartz_oracle.sql</code>
MariaDB	<code>quartz_MariaDB.sql</code>
PostgreSQL	<code>quartz_PostgreSql.sql</code>

When to use the create system tables scripts

You must use these scripts when you install or upgrade Unica Platform if you have not allowed the installer to create the system tables automatically, or if you have used `ManagerSchema_DropAll.sql` to delete all Unica Platform system tables from your database.

ManagerSchema_DropAll.sql

The `ManagerSchema_DropAll.sql` script removes all Unica Platform system tables from a database. This script removes all tables, users, groups, security credentials, and configuration settings from Unica Platform.



Note: If you run this script against a database containing an earlier version of the Unica Platform system tables, you might receive error messages in your database client stating that constraints do not exist. You can safely ignore these messages.

When to use `ManagerSchema_DropAll.sql`

You might want to use `ManagerSchema_DropAll.sql` if you have uninstalled an instance of Unica Platform where the system tables are in a database that contains other tables you want to continue using.

Additional requirements

To make the Unica Platform operational after running this script, you must perform the following steps.

- Run the appropriate SQL script to re-create the system tables.
- Run the `populateDB` utility. Running the `populateDB` utility restores the default configuration properties, users, roles, and groups, but does not restore any users, roles, and groups you have created or imported after initial installation.
- Use the `configTool` utility with the `config_navigation.xml` file to import menu items.
- If you have performed any post-installation configuration, such as creating data filters or integrating with an LDAP server or web access control platform, you must perform these configurations again.

Chapter 12. Uninstalling Unica Platform

Run the Unica Platform uninstaller to uninstall Unica Platform. 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.



Note: On UNIX™, the same user account that installed Unica Campaign must run the uninstaller.

1. If you have deployed the Unica Platform web application, undeploy the web application from WebSphere® or WebLogic.
2. Shut down WebSphere or WebLogic.
3. Stop the processes that are related to Unica Platform.
4. 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.
5. 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.

6. Complete one of the following steps to uninstall Unica Platform:


Choose from:

- Click the Unica Platform uninstaller that exists in the `Uninstall_Product` directory. The uninstaller runs in the mode in which you installed Unica Platform.
- In a command-line window, navigate to the directory where the uninstaller exists, and run the following command to uninstall Unica Platform 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 Platform


```
./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 Platform by using the silent mode:

```
Uninstall_Product -i silent
```

When you uninstall Unica Platform 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 Platform, the uninstaller runs in the mode in which Unica Platform is installed.