Unica Plan V12.1.7 Installation Guide

Contents

Chapter 1. Installation overview

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

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

Installation Roadmap

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

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

Table 1. Unica Plan installation roadmap

Topic	Description
Installation overview on page 1	This topic provides the following information: • How the installers work on page 3 • Modes of installation on page 3. • Unica Plan documentation and help on page 4.
Planning the Unica Plan installation on page 6	This topic provides the following information: • Prerequisites on page 6 • Unica Plan data source information worksheet on page 8. • Installation order for Unica products on page 8
Preparing the data sources for Unica Plan on page 12	This topic provides the following information: • Creating the Unica Plan system table database or schema on page 12 • Configuring the web application server for your JDBC driver on page 13 • Creating the JDBC connections in the web application server on page 14
Installing Unica Plan on page 22	• Installing Unica Plan by using the GUI mode on page 23. • Installing Unica Plan using the console mode on page 29

Table 1. Unica Plan installation roadmap

(continued)

Topic	Description
	• Installing Unica Plan using the console mode on page 29 • Installation prompt windows on page 34 • Creating an EAR file after running the installer on page 32 • JAVA environment variables on page 33 • Installation prompt windows on page 34
Configuring Unica Plan before deployment on page 36	This topic provides the following information: • Registering Unica Plan manually on page 36 • Creating and populating the Unica Plan system tables on page 37 • Setting environment variables on page 39
Overview on page 40	This topic provides the following information: • Deploying Unica Plan on WebSphere on page 45 • Deploying Unica Plan on WebLogic on page 50 • Deploying Unica Plan on JBoss on page 53 • Deploying Unica Plan on Apache Tomcat on page 55
Configuring Unica Plan after deployment on page 58	This topic provides the following information: • Granting the asm_admin user access to Unica Plan on page 59 • Configuring the markup option on page 59 • Configuring the email settings on page 62 • Configuring the integration with Unica Campaign on page 62 • Verifying your installation on page 59
Installing reports on page 65	This topic provides the following information: • Next steps for reporting on page 65
Installing Unica Plan in a cluster on page 66	This topic provides the following information: • WebSphere guidelines on page 66 • WebLogic guidelines on page 68

Table 1. Unica Plan installation roadmap

(continued)

Topic	Description
	• Configuring the shared folder properties on page 70 • Configure ehcache on page 71
Uninstalling Unica Plan on page 74	This topic provides the information about how to uninstall Unica Unica Plan.
configTool on page 75	Read more about the Config tool utilities in Unica Plan.

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 Plan .

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 mode

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

UNIX™ X Window System mode

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

Console mode

Use the console mode to install Unica Plan 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 Plan 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.

Unica Plan documentation and help

The following table describes the various tasks in the installation of Unica Plan.

The **Documentation** column contains names of documents where you can find more information about the tasks.

Table 2. Get up and running

Task	Documentation
View a list of new features, known issues, and workaround	Unica Plan Release Notes
Install or upgrade Unica Plan and deploy the Unica Plan web application	One of the following guides: • Unica *Plan Installation Guide* • Unica *Plan Upgrade Guide*

The following tables describes administrative tasks in Plan. The **Documentation** column contains names of documents where you can find more information about the tasks.

Table 3. Configure and use Unica Plan

Task	Documentation
• Set up and configure the system for users • Adjust security settings • Map tables, define offer templates, and custom attributes • Run utilities and perform maintenance	*Unica Plan Administrator's Guide*

Table 3. Configure and use Unica Plan

(continued)

Task	Documentation
• Create and manage marketing plans, programs, and projects • Analyze progress and results	*Unica Plan User's Guide*

The following tables contains information about Online Help and obtaining PDFs for Unica Plan. The **Instructions** column describes how to open Online Help and access documentation for the Unica Plan.

Table 4. Getting help

Task	Instructions
Open online help	1. Choose **Help > Help for this page** to open a context-sensitive help topic. 2. Click the **Show Navigation** icon in the help window to display the full help. You must have web access to view context-sensitive online help. Contact HCL Support for more information about accessing and installing Knowledge Center locally for offline documentation.
Obtain PDFs	Use either of the following methods: • Choose **Help > Product Documentation** to access Plan PDFs. • Choose **Help > All Unica Suite Documentation** to access all available documentation.

Chapter 2. Planning the Unica Plan installation

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

Prerequisites

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

System requirements

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

Network domain requirements

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

JVM requirements

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

Knowledge requirements

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

Internet browser settings

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

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

Access permissions

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

- Administration access for all necessary databases

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

- Read and write access to the relevant directory and sub-directories for the operating system account that you use to run the web application server and Unica components.
- Write permission for all files that you must edit.

- Write permission for all directories where you must save a file, such as the installation directory and backup directory if you are upgrading.
- Appropriate read, write, and execute permissions to run the installer.

Verify that you have the administrative password for your web application server.

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

The following additional permissions are necessary for UNIX™:

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

Points to consider before you install Unica Plan

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

JAVA_HOME environment variable

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

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

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

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

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

export JAVA_HOME= (leave empty)

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

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

Unica Platform requirements

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

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

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

Unica Plan data source information worksheet

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

Table 5. Data source information worksheet

Item	Value
Data source type	SQL Server / IBM DB2 / Oracle / MariaDB / OneDB / PostgreSQL. **Note:** PostgreSQL support is not available if you are upgrading to 12.1.7. It is only available for clean installation of 12.1.7.
Data source name	Name of Plan data source.
Data source account user name	Database username.
Data source account password	Database password.
JNDI name	`plands`
Path to the JDBC driver	Path for JDBC driver file.

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 6. Installation or upgrade order for Unica products

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

Table 6. Installation or upgrade order for Unica products

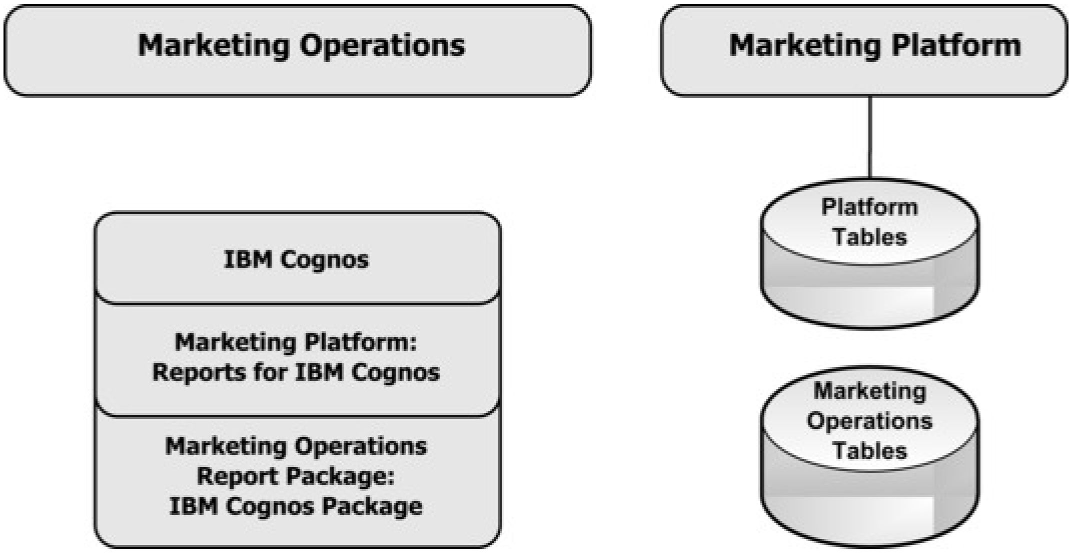
(continued)

For this product or combination:	Install or upgrade in this order:
	Note: If you are integrating Unica Plan with Unica Campaign, you must also install Unica Campaign. The order of installation for those two products does not matter.
Unica Interact Advanced Patterns	1. Unica Platform 2. Unica Campaign 3. Unica Interact 4. Unica Interact Advanced Patterns
IBM SPSS Modeler Advantage Enterprise Marketing Management Edition	1. IBM SPSS Modeler Advantage Enterprise Marketing Management Edition

Where to install Unica Plan and the Unica Platform

The following diagram provides a brief overview of where to install Unica Plan. It illustrates the most basic functional installation.

You may require a more complex, disparate installation to meet your security and performance requirements.


Marketing Operations
Marketing Platform
IBM Cognos
Marketing Platform: Reports for IBM Cognos
Marketing Operations Report Package: IBM Cognos Package
Platform Tables
Marketing Operations Tables

Unica Plan: For best performance, install Unica Plan on its own machine, where no other Unica products are installed, or on a machine that it shares with only the Unica Platform.

The Unica Plan system tables should be **on a separate machine**.

Unica Plan reports package: The reports package for Unica Plan contains the IBM® Cognos® package only. (Other applications also have reporting schemas to configure, but Unica Plan does not.) Install the reports package on the IBM® Cognos® system.

Unica Platform: 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.

Chapter 3. Preparing the data sources for Unica Plan

You can use the Unica Plan worksheet to enter the information that is required when you install Unica Plan.

Print the Unica Plan data source information worksheet on page 8 located at the end of this chapter. Then, as you complete each task in this chapter, fill in the information in your checklist. Having this information written down makes it easier for you to provide the database connection information when you run the installer later in the installation process.

Creating the Unica Plan system table database or schema

Create the Unica Plan system table database or schema with the help of your database administrator. After you create the database, complete the data source information worksheet for future reference.

About this task

Complete the following steps to create Unica Plan system tables database or schema:

1. Work with a database administrator to create the required database for Unica Plan.
2. Have the database administrator create an account for you to specify for a system user later in the installation process.

   This account must have CREATE, SELECT, INSERT, UPDATE, DELETE, and DROP rights for both tables and views, as appropriate. In addition:

   - The database must use UTF-8 encoding.
   - If you are using SQL Server, ensure that TCP/IP is enabled.
   - If you are using DB2®, ensure that the buffer pool for the table space is at least 32K.
   - For Maria DB , lower_case_table_names is 1 so that table names are considered case insensitive
3. Print and complete the Unica Plan data source information worksheet on page 8. You use the information later in the installation process.

IBM® DB2® database tablespaces

The DB2® database manages Database-managed Space (DMS) tablespaces. A tablespace is a storage location that contains the data sets where the DB2® tables are stored. Create tablespaces for the IBM® DB2® database before you install Plan.

You can specify the following types of tablespaces for the application data:

- Online transaction Processing (OLTP) data tablespace. Use this tablespace to store the transactional data from the application.
- OLTP index tablespace. Use this tablespace to store the indexes that are created to access the OLTP data tables.
- Discussion Support System (DSS) data tablespace. Use this tablespace to store the OLTP data that is loaded into the DSS schema. The DSS schema organizes data from the OLTP to make it easier to generate reports of the application's activity.
- DSS index tablespace. Use this tablespace to store the indexes that are created to access the DSS data tables.

Make note of the names of the tablespaces and the name of the server where you create the database. These tablespaces must exist before you install Plan. You must allocate at least 100 MB of space each for the OLTP data and DSS data tablespaces, and at least 50 MB each for the OLTP index and DSS index tablespaces.

If you do not want to maintain all four tablespaces, you can specify the same tablespace for multiple fields in the installation program. For example, you can supply the same tablespace for both the OLTP data and OLTP index tables, or a single tablespace for all four types of tables. Allocate sufficient space for all the tables that are written to the tablespace.

In addition, you must provide a temporary (TMP) tablespace of at least 10 MB.

The database user that you specify when you install Plan must be associated with the tablespaces, or have permissions to administer the tablespaces. The Leads application must be able to create schema in these tablespaces and write initial data to them.

Configuring the web application server for your JDBC driver

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

About this task

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

If you are using WebLogic, you must perform this entire procedure.

Note: If you are installing Unica Plan on the same machine where you installed Unica Platform, this task is already complete. Continue to Creating the JDBC connections in the web application server on page 14.

1. Obtain the latest Type 4 JDBC drivers and any required associated files (for example, Oracle requires several associated files) for the database you plan to use. See the *Recommended Software Environments and Minimum System Requirements* guide for details.

   You should always use the latest vendor-provided Type 4 driver.

   - If the driver does not exist on the machine where Unica Plan will be installed, obtain it and copy it to any location on the Unica Plan machine.
   - If you obtain the driver from a machine where the database client is installed, verify that the version is the latest supplied by the database vendor. See your consultant for a list of supported JDBC drivers.
2. Include the full path to the driver in the `CLASSPATH` of the web application server where you plan to deploy Unica Plan, as follows.

- For all supported versions of WebLogic, add the `jar` files to the CLASSPATH variable in `DOMAIN_DIR\bin\setDomainEnv.cmd`. Yours must be the first entry of the `CLASSPATH` values to ensure that the web application server uses the correct driver. For example, if you are using SQL Server you might set the path as follows.

  ```
  set CLASSPATH=c:\SQLDRIVER\sqljdbc.jar;%PRE_CLASSPATH%;%WEBLOGIC_CLASSPATH%;
  %POST_CLASSPATH%;%WLP_POST_CLASSPATH%
  ```
- For all supported versions of WebSphere®, set the `CLASSPATH` in the Administration Console.

Creating the JDBC connections in the web application server

The Unica Plan web application uses JDBC connections to communicate with the system table database and with the Unica Platform system table database.

About this task

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

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

- The installer creates the data source using `plands` 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.

In Tomcat®, set the classpath for your database driver during this process.

Note: You must manually encrypt password using `encryptTomcatDBPasswords.sh` utility available under Platform_HOME/tools/bin.

In JBoss®, set the classpath for your database driver by adding the module for the JDBC driver and register the SQL JDBC driver.

Important: For the connection to the database holding the Unica Plan system tables, you must use `plands` as the Java™ Naming and Directory Interface (JNDI) name. This value is the required JNDI name.

Important: For the connection to the database holding the Unica Platform system tables, you must use `UnicaPlatformDS` as the JNDI name. This is the required JNDI name. If you are deploying Unica Plan and Unica Platform in the same JVM, this connection should already exist.

If you expect to have many concurrent users in Unica Plan, you may need to increase the number of connections in your web server. For best results, set your web server to allow 50 connections.

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

- Driver: org.mariadb.jdbc.Driver
- Default port: 3306
- Driver class: `org.mariadb.jdbc.MariaDbDataSource`
- Driver URL: `jdbc:mysql://hostname:port/databaseName`

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.

Add the following custom property:

Name: `webSphereDefaultIsolationLevel`

Value: `blank` or `2`

Datatype: `Integer`

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`

MariaDB

- Driver: `MariaDBDriver` (this is the mariadb jar registered with mariadb)
- Default port: `3306`
- Driver class: `org.mariadb.jdbc.MariaDbDataSource`
- Driver URL: `jdbc:mariadb://<your_db_host>:<your_db_port>/<your_db_name>jdbc:db2://<your_db_host>:<your_db_port>/<your_db_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

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: 8232
- Driver class: `oracle.jdbc.OracleDriver`
- Driver URL: `jdbc:oracle:thin:@`*`<your_db_host>:<your_db_port>:<your_db_service_name>`* `valid-connection-checker-class-name=valid-connection-checker-class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker"`

For example:/subsystem=datasources/data-source=UnicaPlatformDS:add(jndiname=" java:/UnicaPlatformDS",connection-url="jdbc:oracle:// localhost:8232;databaseName=plat11",driver-name=oracle,username= plat,password=test1234,valid-connection-checker-class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker")

DB2

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

For example:/subsystem=datasources/data-source=UnicaPlatformDS:add(jndiname=" java:/UnicaPlatformDS",connection-url="jdbc:db2:// localhost:50000;databaseName=plat11",driver-name=db2,username= plat,password=test1234,valid-connection-checker-class-name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionChecker"

MariaDB

- Driver: `MariaDBDriver` (this is the mariadb jar registered with mariadb)
- Default port: `3306`
- Driver class: `org.mariadb.jdbc.MariaDbDataSource`
- Driver URL: `jdbc:mariadb://<your_db_host>:<your_db_port>/<your_db_name>jdbc:db2://<your_db_host>:<your_db_port>/<your_db_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

Tomcat

Specify the native library path of the database driver JAR file on your server. For example: `db2jcc4.jar/ojdbc8.jar/sqljdbc4.jar` to create a datasource under Tomcat Interact Instance, add folder hierarchy as `<Tomcat_Home>/conf/Catalina/localhost`, create a xml file with the name `interact.xml` under localhost folder. Add data source pointing to Runtime, Test, Production and learning database in interact.xml, refer platform or Campaign xml file which is created through installers under there respective Tomcat instance directory. You will need to create these Data source entry in Campaign Tomcat instance too.

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

Set **removeAbandoned=true** and **removeAbandonedTimeout=300** (5 minutes) for `plands` datasource configuration in `<catalina home>/conf/Catalina/localhost/plan.xml` file within the section that starts with the parameter **Resource name="plands"**.

For example:

```
<Resource name="plands" auth="Application" type="javax.sql.DataSource"
 factory="org.apache.tomcat.jdbc.pool.DataSourceFactory" maxTotal="100" maxIdle="30" maxWaitMillis
 ="10000" username="abc" password="abc" driverClassName="org.mariadb.jdbc.Driver" removeAbandoned="true"
 removeAbandonedTimeout="300" url="jdbc:mysql://lp2-ap-51728467.prod.hclpnp.com:3306/pln121" />
```

Note: If the application executes SQL queries that requires more than `300` seconds of execution time, change the value of **removeAbandonedTimeout**.

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: `MariaDBDriver` (this is the mariadb jar registered with mariadb)
- Default port: `3306`
- Driver class: `org.mariadb.jdbc.MariaDbDataSource`
- Driver URL: `jdbc:mariadb://<your_db_host>:<your_db_port>/<your_db_name>jdbc:db2://<your_db_host>:<your_db_port>/<your_db_name>`

Note: You must update the URL for Platform data source in plan.xml manually.

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 Plan

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

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

Note: The Unica installer requires information about Unica Platform.

After installing Campaign, 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.

After installing Opportunity Detect, you can create an EAR file if necessary.

Important: Before you install Plan, make sure that the available temporary space on the computer where you install Campaign is more than three times the size of the Plan installer.

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 7. Installation files

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

Table 7. Installation files

(continued)

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

Installing Unica Plan by using the GUI mode

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

Before you begin

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

Important: On Microsoft Windows 2019 server and Microsoft Windows 2022 server, to launch the installers, set the parameter `SET JAVA_TOOL_OPTIONS="-Dos.name=Windows 7"` at the command prompt and launch the installers from the same command prompt.

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

To install Plan by using the GUI mode (for Windows™) or the X Window System mode (for UNIX™):

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

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

Table 8. Unica installer GUI

Window	Description
Introduction	This is the first window of the Unica suite installer. You can open the Plan installation and upgrade guides from this window. Click **Next** to go to the next window.
Response Files Destination	Click the **Generate Response File** check box if you want to generate response files for your product. Response files store the information that is necessary for the installation of your product. You can use response files for an unattended installation of your product. Click **Choose** to browse to a location where you want to store the response files. Click **Next** to go to the next window.
HCL Unica Products	In the **Install Set** list, select **Custom** to select the products that you want to install. In the **Install Set** area, you can see all the products whose installation files are in the same directory on your computer. In the **Description** field, you can view the description of the product that you select in the **Install Set** area. Click **Next** to go to the next window.
Installation Directory	Click **Choose** to browse to the directory where you want to install Unica. Click **Next** to go to the next window.
Select Application Server	Select one of the following application servers for the installation: • IBM® WebSphere® • Oracle WebLogic • JBoss • Tomcat Click **Next** to go to the next window.
Platform Database Type	Select the appropriate Unica Platform database type. Click **Next** to go to the next window.
Platform Database Connection	Enter the following information about your database: • Database host name • Database port • Database name or System ID (SID)

Window	Description
	• Database user name • Database password Click **Next** to go to the next window. **Important:** If Unica products are installed in a distributed environment, you must use the machine name rather than an IP address in the navigation URL for all of the applications in the suite. Also, if you are on a clustered environment and choose to use ports that are different from the default ports `80` or `443` for your deployment, do not use a port number in the value of this property.
Platform Database Connection (continued)	Review and confirm the JDBC connection. Click **Next** to go to the next window.
Preinstallation Summary	Review and confirm the values that you added during the installation process. Click **Install** to start the installation process. The Unica Platform installer opens. If a previous instance of Unica Platform exists, the instance is upgraded to the current version. If a previous instance of Unica Platform does not exist, Unica Platform is installed.

4. Follow the instructions on the Unica Platform installer to install or upgrade Unica Platform. See Unica Platform Installation Guide for more information.

   Note: If you want to use Unica Content Integration, for configuring third-party CMS for use with Unica Plan, install the Asset Picker component when installing Unica Platform. For more information, see the topic **Unica Platform components** in the *Unica Platform Installation Guide*.

5. In the **Installation Complete** window, click **Done**.

   Result

   The Unica Platform installation is complete, and the Unica Plan installer opens.
6. Use the information in the following table to navigate the Unica Plan installer. In the **Platform Database Connection** window, enter all the required information and click **Next** to start the Plan installer.

Table 9. Unica Plan installer GUI

Window	Description
Introduction	This is the first window of the Plan installer. You can open the Unica Plan installation and upgrade guides from this window. Click **Next** to go to the next window.

Window	Description
Software License Agreement	Carefully read the agreement. Use **Print** to print the agreement. Click **Next** after you accept the agreement.
Installation Directory	Click **Choose** to browse to the directory where you want to install Plan. Click **Next** to go to the next window.
Plan Components	Select the components that you want to install. Click **Next** to go to the next window.
Plan Database Setup	Select one of the following options for setting up the Plan database: • Automatic database setup • Manual database setup **Note:** • If you select **Automatic database setup**, the logs will be available in the `Plan_stdout.log` file. • If you select **Manual database setup**, see Creating and populating the Unica Plan system tables on page 37 for additional details. Click **Next** to go to the next window.
Plan Database type	Select the appropriate database type. Click **Next** to go to the next window.
Plan Database connection	Enter the following details for the Plan database: • Database host name • Database port • Database system ID (SID) • Database user name • Password **Important:** If Unica products are installed in a distributed environment, you must use the machine name rather than an IP address in the navigation URL for all of the applications in the suite. Also, if you are on a clustered environment and choose to use ports that are different from the default ports `80` or `443` for your deployment, do not use a port number in the value of this property. Click **Next** to go to the next window.
Plan JDBC connection	Review and confirm the JDBC connection.

Window	Description
	Click **Next** to go to the next window.
Plan Connection Settings	Enter the following connection settings: • Network domain name • Host name • Port number Select the **Use secure connection** check box if necessary. Click **Next** to go to the next window.
Datasource creation	The installer can create the JDBC data source in your web application server (WebSphere or WebLogic). This data source is required to enable the Plan web application to connect to the Plan system tables. You have the option to skip this step in the installer and create the data source in the administration console of your web application server after installation is complete. If you select the **Create Plan Datasource** check box, the installer creates the data source using information you provide.. • Your application server must be running. • Enter any name for the datasource. The installer creates the JNDI name automatically (plands). • Enter the directory in which the application server is installed. For WebSphere, this is the directory that contains the `profiles` directory. You must restart WebSphere after the installer finishes and before you use this data source. For WebLogic, this is the directory that contains the `common` directory. Click **Next** to go to the next window.
Application server information (WebSphere)	Applies only if your application server is WebSphere. • Enter the application server profile where you plan to deploy Plan. • Enter the name of the server in the profile. • If security is enabled in WebSphere, enter the administrator user ID and password. Click **Next** to go to the next window.
Application server information (WebLogic)	Applies only if your application server is WebLogic. Enter the domain server name and the administrator user ID and password. If SSL is enabled, enter the HTTP port for the domain. Note that you must add the database driver classpath to WebLogic in the WebLogic administration console. Click **Next** to go to the next window.

Window	Description
Application server information (WebLogic)	Applies only if your application server is WebLogic. Enter the domain server name and the administrator user ID and password. If SSL is enabled, enter the HTTP port for the domain. Note that you must add the database driver classpath to WebLogic in the WebLogic administration console. Click **Next** to go to the next window.
Application server information (JBoss)	Applies only if your application server is JBoss. Enter the domain server name and the administrator user ID and password. If SSL is enabled, enter the HTTP port for the domain. Note that you must add the database driver classpath to JBoss. Click **Next** to go to the next window.
Application server information (Tomcat)	Applies only if your application server is Tomcat. Enter the domain server name and the administrator user ID and password. If SSL is enabled, enter the HTTP port for the domain. Note that you must add the database driver classpath to Tomcat. Click **Next** to go to the next window.
Supported Locales	This window displays all the locales supported by Plan.
Default Locale	Select a default locale for your installation. English is selected by default. Click **Next** to go to the next window.
Pre-Installation Summary	Review and confirm the values that you added during the installation process. Click **Install** to start the installation process. The Plan installer opens.
Installation Complete	Click **Done** to close the Unica Platform installer and go back to the Unica installer.

7. In the **Installation Complete** window, click **Done** to exit the Plan installer and go back to the Unica installer.
8. Follow the instructions on the Unica installer to finish installing Plan.
   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
Deployment EAR file	Specify whether you want to create an enterprise archive (EAR) file to deploy your Unica products.

Window	Description
	Click **Next** to go to the next window.
Package EAR file	You can see this window if you select **Create an EAR file for deployment** in the **Deployment EAR file** window. Select the applications that you want to package in the EAR file.
EAR file details	Enter the following information for your EAR file: • Enterprise application identifier • Display name • Description • EAR file path
EAR file details (continued)	Select **Yes** or **No** to create an additional EAR file. If you select **Yes**, you must enter the details for the new EAR file. Click **Next** to complete the installation of your product.
Deployment EAR file	Specify whether you want to create another EAR file to deploy your Unica products. Click **Next** to go to the next window.
Installation Complete	This window provides the locations of the log files that are created during installation. Click **Previous** if you want to change any installation details. Click **Done** to close the Unica installer.

Installing Unica Plan using the console mode

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

Choose from:

- For Windows™, enter the following command:

  `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_Software_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 Optimize
   4 [ ] Unica Collaborate

   If you want to install Distributed Marketing, 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 Optimize
   4 [X] Unica Collaborate

   Note: Do not clear the option for Unica Platform unless you have already installed it.
5. The Unica installer launches theUnica Interact installers during the installation process. Follow the instructions in the command-line prompt window of the Interact installers.
6. After you enter `quit` in the Unica Interact 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 Interact.

   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 Plan in a silent mode

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

Before you begin

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

- An application server profile
- A database schema

About this task

When you install Unica Plan 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_Software_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_Software_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 Plan. You can use sample response files to create your response files. The sample response files are included with the installers in the `ResponseFiles` compressed archive.

The following table provides information about sample response files:

Table 11. Description of sample response files

Sample response file	Description
`installer.properties`	The sample response file for the Unica master installer.
`installer_`*`product initials and product version number`*`.properties`	The sample response file for the Unica Plan installer. For example, `installer_ucn.n.n.n.properties` is the response file of the Unica Campaign installer, where *n.n.n.n* is the version number. For example, `installer_umpn.n.n.n.properties` is the response file of the Unica Plan installer, where *n.n.n.n* is the version number. For example, `installer_uln.n.n.n.properties` is the response file of the Leads installer, where *n.n.n.n* is the version number.
`installer_`*`report pack initials, product initials, and version number`*`.properties`	The sample response file for the reports pack installer. For example, `installer_urpcn.n.n.n.properties` is the response file of the Unica Campaign reports pack installer, where *n.n.n.n* is the version number For example, `installer_urpl.properties` is the response file of the Leads reports pack installer.

Creating an EAR file after running the installer

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

About this task

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

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

1. If this is the first time you are running the installer in console mode, make a backup copy of the installer's `.properties` file for each of your installed products.
    - Each product installer creates one or more response files with a `.properties` extension. These files are in the same directory where you placed the installers. Be sure to back up all files with the `.properties` extension, including the `installer_<product initials><product version number>.properties` files and the file for the installer itself, which is named `installer.properties`.

      For example, the properties files for Unica Platform `installer_ump12.1.0.0.properties` and the properties file for Campaign `installer_uo12.1.0.0.properties`.
    - If you plan to run the installer in unattended mode, you must back up the original `.properties` files, because when the installer runs in unattended mode, it clears these files. To create an EAR file, you need the information that the installer writes in the `.properties` files during the initial installation.
2. Open a command window and change directories to the directory that contains the installer.
3. Run the installer executable with this option:

    ```
    -DUNICA_GOTO_CREATEEARFILE=TRUE
    ```

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

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

JAVA environment variables

Java™ environment variables are global system variables that store system-wide values. To install Unica products, ensure that your machine has a Java™ Runtime Environment (JRE) version 1.8 or above.

Note: The *JAVA_HOME* environment variable is not required for installing Unica products, but if it is present, it must be the 1.8 version of the Sun JRE.

If you have a `JAVA_HOME` environment variable, and it points to an incorrect JRE, you must unset the *JAVA_HOME* variable before you run the Unica installers. To unset *JAVA_HOME* variable:

- Windows™: In a command window, enter

  ```
  set JAVA_HOME=leave empty and press return key
  ```
- UNIX™ type systems: In the terminal, enter

  ```
  export JAVA_HOME=leave empty and press return key
  ```

After the environment variable is unset, the Unica installers use the JRE bundled with the installers.

You can reset the environment variable after installation is complete.

Installation prompt windows

During the installation of Unica Plan, several prompt windows display. After you enter the required information, prompt windows display your responses and require a confirmation before you continue. Prompt windows give you the opportunity to make corrections, if needed.

For reference, an example of the prompts that display during installation on a UNIX™ server using console mode follows. Be sure to read and follow the instructions that display during your actual installation.

Use the following example to help you gather required information before you begin your installation and for reference during your installation.

Table 12. Example installation prompts and responses

Prompt	Response
`-bash-4.OS`	Initial prompt. Supply the name of the master installer file and any variables for the database setup utility that you want to use for the installation.
`Choose Locale`	Supply a number to choose one of the listed languages. To use the default locale, `2- English`, press Enter.
`Introduction`	If a previous version of the product is installed, an upgrade is initiated. If the same version of the product is installed, proceeding drops all tables and data.
`Response Files Generation`	Supply a number to choose whether to generate response files for use in an unattended installation. If you generate response files, you can supply a destination path.

`Choose Product Features`	A numbered list of features displays. Features with a check mark `[X]` are selected for installation, features with no check mark `[ ]` are not selected. To change the selections, supply numbers to switch from selected to cleared (or vice versa) by using a comma-separated list, then press Enter.
	For example, you see this list of features:
	`1-  Unica Platform`
	`2-  Unica Plan`
	To install Unica Platform only, supply `2` and press Enter.

Master (Unica Platform) installation	
`Installation Directory`	
`Select Application Server`	
`Platform Database Type`	Supply information for the Unica Platform system table database.
`Platform Database Host Name`	
`Platform Database Port`	

Table 12. Example installation prompts and responses (continued)

Prompt	Response
`Platform Database Name/System ID (SID)`	
`Platform Database User Name`	
`Platform Database Password`	
`JDBC Connection`	
`JDBC Driver Classpath`	
Product-specific (Unica Plan) installation	
`Introduction`	For each product feature you chose to install, the specific product name displays followed by the reinstallation warning.
`Installation Directory`	
`Plan Database Setup`	Supply a number to choose automatic or manual. • Automatic setup uses the same information for this feature that you supplied for the master installation. • Manual setup prompts you for each database and JDBC characteristic separately to accommodate any feature-specific differences.
`Plan Server/Host`	
`Plan Server Port`	
`Plan Domain Name`	Supply the same company domain, in all lowercase, for every feature you install. **Note:** To avoid any issues if an IP address must be changed after the product is installed, use a fully qualified domain name instead of an IP address.
`Supported Locales`	Supply a number to choose a language. You can also supply a comma-separated list to select multiple locales.
`Default Locale`	Supply a number to choose a language.
`Deployment EAR File`	Supply a number to choose whether to create an enterprise archive (EAR) file.

Chapter 5. Configuring Unica Plan before deployment

Complete the configuration tasks before you deploy the web application.

Registering Unica Plan manually

If the Unica Plan installer cannot connect with the Unica Platform system table database during the installation, your installation fails. In this case, you must register Unica Plan manually.

About this task

Even if the installers fail to connect to the system tables, the installation process proceeds. In this case, you must import the product information into the Unica Platform system tables manually.

The `configTool` utility that is mentioned in this procedure is in the `tools/bin` directory under your Unica Platform installation. For detailed instructions on using the `configTool` utility, see configTool on page 75.

Complete the following steps to register Unica Plan manually:

1. Set the environment variable by completing following actions:

   Choose from:

   - For Windows™, set an environment variable named `NAVIGATION_DIR` to the Unica Plan `conf` directory.
   - For Unix, set an environment variable named `$NAVIGATION_DIR` to the Unica Plan `conf` directory.
2. Run the `configTool` utility by using the following example commands as guidelines.

   Choose from:

   - For Windows™, use the following commands:

     configTool.bat -v -i -p "Affinium" -f "%NAVIGATION_DIR%\plan_registration.xml"

     configTool.bat -v -i -p "Affinium|suite|uiNavigation|mainMenu" -f "%NAVIGATION_DIR%\plan_navigation_operations.xml"

     Run the following command if the Financial Module is installed in Unica Plan.

     configTool.bat -v -i -p "Affinium|suite|uiNavigation|mainMenu|Analytics" -f "%NAVIGATION_DIR%\plan_navigation_analytics.xml"

     configTool.bat -v -i -p "Affinium|suite|uiNavigation|settingsMenu" -f "%NAVIGATION_DIR%\plan_navigation_settings.xml"

     configTool.bat -v -i -p "Affinium|suite|uiNavigation|alerts" -f "%NAVIGATION_DIR%\plan_alerts_registration.xml"

     configTool.bat -v -i -p "Affinium|suite|uiNavigation|mainMenu" -f "%NAVIGATION_DIR%\plan_navigation_financials.xml"

     configTool.bat -v -i -p "Affinium|suite|uiNavigation|mainMenu" -f "%NAVIGATION_DIR%\plan_navigation_assets.xml"

     configTool.bat -v -i -p "Affinium|suite|quicklinksCategory" -f "%NAVIGATION_DIR%\umo_quicklinks_registration.xml"
   - For Unix, use the `./configTool.sh` file to run `configTool` utility with the help of following commands:

./configTool.sh -v -i -p "Affinium" -f "$NAVIGATION_DIR/plan_registration.xml"

./configTool.sh -v -i -p "Affinium|suite|uiNavigation|mainMenu" -f "$NAVIGATION_DIR/plan_navigation_operations.xml"

Run the following command if the Financial Module is installed in Unica Plan.

./configTool.bat -v -i -p "Affinium|suite|uiNavigation|mainMenu|Analytics" -f "$NAVIGATION_DIR/plan_navigation_analytics.xml"

./configTool.bat -v -i -p "Affinium|suite|uiNavigation|settingsMenu" -f "$NAVIGATION_DIR/plan_navigation_settings.xml"

./configTool.bat -v -i -p "Affinium|suite|uiNavigation|alerts" -f "$NAVIGATION_DIR/plan_alerts_registration.xml"

./configTool.sh -v -i -p "Affinium|suite|uiNavigation|mainMenu" -f "$NAVIGATION_DIR/plan_navigation_financials.xml"

./configTool.sh -v -i -p "Affinium|suite|uiNavigation|mainMenu" -f "$NAVIGATION_DIR/plan_navigation_assets.xml"

./configTool.bat -v -i -p "Affinium|suite|quicklinksCategory" -f "$NAVIGATION_DIR/\umo_quicklinks_registration.xml"

Note: In case manual registration, if Unica Platform is not installed on same computer as Unica Plan, you must install the Unica Platform tools on the computer or copy the Unica Plan xml configuration files to your computer.

Creating and populating the Unica Plan system tables

If the automatic database setup fails during the installation of Unica Plan, you must create and populate Unica Plan system tables manually. To generate Unica Plan system tables, you must run the `umodbsetup` utility.

The `umodbsetup` utility completes one of the following tasks:

- Creates the required system tables in the Unica Plan database and populates the tables with required default data.
- Outputs the database creation and population scripts to a file that you or your DBA can then run in your own database client.

Configuring environment variables

Before you run the `umodbsetup` utility, complete the following steps to configure the environment variables properly:

1. From the `<HCL_Unica_Home>\<Plan_Home>\tools\bin` directory, locate and open the `setenv` file in a text editor.
2. Verify that the JAVA_HOME variable indicates the correct directory for your installation of Java™, and that the JDBC drivers are the first entry for the DBDRIVER_CLASSPATH variable. For more information about setting this environment variable, see JAVA environment variables on page 33.

3. Save and close the file.
4. From the *<HCL_Unica_Home>**<Plan_Home>*\tools\bin directory, locate and open the umo_jdbc.properties file.
5. Set values for the following parameters.
   - **umo_driver.classname**
   - **umo_data_source.url**
   - **umo_data_source.login**
   - **umo_data_source.password**
6. Save and close the file.

Running the database setup utility

From a command prompt or UNIX™ shell, go to the <HCL_Unica_Home>\<Plan_Home>\tools\bin directory. Run the umodbsetup utility and provide appropriate input for the parameters that are required for your situation.

For example, the following command runs a full database installation (rather than an upgrade), sets the locale to en_US, and sets the logging level to high:

```
./umodbsetup.sh -t full -L en_US -l high
```

Following is a description of all the possible variables for the utility:

Table 13. Variables for the umodbsetup.sh utility

Variable	Description
-h	Provides help for the utility.
-l	Records the output in the umo-tools.log file from the actions of the umodbsetup utility . This file is in the *<HCL_Unica_Home>**<Plan_Home>*\tools\logs directory. This variable specifies the logging level. You can set the logging level to high, medium, or low.
-L	Sets the default Locale for the installation. For example, use -L de_DE for a German installation. The valid entries for locales include de_DE, en_GB, en_US, es_ES, fr_FR, it_IT, ja_JP, ko_KR, pt_BR, ru_RU, zh_TW, and zh_CN. **Note:** The locale information is case-sensitive and must be used as mentioned in the guide.
-m	Outputs the scripts to a file in the *<HCL_Unica_Home>**<Plan_Home>*\tools directory, which you can then run manually. Use this option if there is a reason you must run scripts from your own database client application. When you use this variable, the umodbsetup tool does not run the script.
-t	Type of database installation. Valid values are full and upgrade. For example, -t full

Table 13. Variables for the `umodbsetup.sh` utility (continued)

Variable	Description
-v	Verbose.
-b	For upgrades only. Identifies the base version of the database that you are attempting to upgrade. By default, the utility detects the version of the database you are upgrading. However, if an earlier attempt to upgrade the database failed in some way, the version might be the updated one, even though the upgrade failed. When you have correct the problem and run the utility again, you use this variable with the `-f` variable to specify the correct base version. For example, `-f -b 9.0.0.0`
-f	For upgrades only. Instructs the utility to use the base version that is specified by the `-b` variable, overriding the base version that may detect in the database. See the description of the `-b` variable.
-E	This option is used to encrypt existing password available in file. You can use this option along with other options like -t, -P. For example, umodbsetup.bat/sh -E
-P	This option is be used to change existing password and encrypt it. If user chooses this option, the tool prompts user to enter new password. The new password is stored in `umo_jdbc.properties` file after encrypting. Use this option independently as it prompts for new password. For example, umodbsetup.bat/sh -P

Running the database scripts manually

If you used the *-m* variable to output the scripts so you can run them from your own database client application, run the scripts now.

Do not deploy the `plan.war` file before you create and populate the system tables.

Setting environment variables

Specify the environment variables if you plan to deploy Unica Plan on a WebLogic web application server that is installed on a Windows™ machine.

About this task

On the machine where WebLogic is installed, add the following values to the Path System environment variable:

- Full path to the `bin` directory where the Sun JDK is installed.
- Full path to the `server\bin` directory where WebLogic is installed.

Chapter 6. Overview

There are general guidelines for deploying Unica Plan on WebSphere® and WebLogic.

If you created an EAR file after running the installer and your EAR file includes any other products, you must follow all the deployment guidelines detailed in the individual install guides for the products included in the EAR file in addition to those listed in this chapter.

It is assumed that you know how to work with your web application server. Consult your web application server documentation for details about things like how to use the Administration console.

Configuration of JVM parameters for Unica products

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

Table 14. Unica Products and the JVM Parameters to be Configured

Unica Product Name	JVM Parameters
Unica Campaign	• `-DUNICA_PLATFORM_CACHE_ENABLED=true` • `-DLICENSE_ENABLED=false` • `-Dcampaign.log4j.async=true` • `-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote` • `-Dcom.sun.management.jmxremote.port=1007` • `-Dcom.sun.management.jmxremote.ssl=false` • `-Dcom.sun.management.jmxremote.authenticate=false` • `-Dcom.sun.management.jmxremote.password.file=../conf/jmxremote.password` • `-Dcom.sun.management.jmxremote.access.file=../conf/jmxremote.access` • `-DUNICA_GOTO_CREATEEARFILE=TRUE` • `-Dfile.encoding=UTF-8` • `-Dclient.encoding.override=UTF-8` • `-DUNICA_GOTO_CREATEEARFILE=TRUE` • `-Dclient.encoding.override=UTF-8` • `-Djboss.as.management.blocking.timeout=3600` SET SSL_OPTIONS=-Djavax.net.ssl.keyStoreType="JKS"

Table 14. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters

	• `-Djavax.net.ssl.trustStore="C:\security\myTrustStore.jks"`
	• `-Djavax.net.ssl.trustStorePassword=myPassword`
	Configuring the `ACOOptAdmin` tool (`ACOOptAdmin.sh` (UNIX) or `ACOOptAdmin.bat` (Windows) file)
	• `-Djavax.net.ssl.keyStoreType=keyStoreTypeValue`
	• `-Djavax.net.ssl.keyStore=<keyStoreValue>`
	• `-Djavax.net.ssl.keyStorePassword=<keyStorePasswordValue>`
	• `-Djavax.net.ssl.trustStore=<trustStoreValue>`
	• `-Djavax.net.ssl.trustStorePassword=<trustStorePasswordValue>`
	• `-Djava.awt.headless=true`
	• `-Dcampaign.deliveretl.disabled=true`
	• `-Dcampaign.interactetl.disabled=true`
	• `-Dcampaign.journeyetl.disabled=false`
	• `-Dcampaign.journeyetl.disabled=true`
	`EMAIL_TLS_ENABLED`
	`EMAIL_SSL_TRUST_ENABLED`
	`SCH_ASYNCH_EXECUTION_ENABLED`

Unica Audience Central	• `-DAUDIENCE_CENTRAL_HOME=<Install_Base_DIR>/AudCentral`
Unica Contact Central	• `CONTACTCENTRAL_HOME` • `UNICA_PLATFORM_HOME`
Unica Content Integration	• `-DASSET_PICKER_HOME` • `-DUNICA_PLATFORM_HOME`
Unica Deliver	• `-Dcampaign.deliveretl.disabled=true`
Unica Interact	**Interact design time** The following are the JVM parameters for Interact design time: • `INTERACTDT_HOME` • `Interact.UsernameToAlwaysDeployFor`

Table 14. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	• `com.unicacorp.interact.deliver.templateTimeout` • `DeploymentServletParameterEncoding` • `com.unicacorp.interact.flexoffers.defaultDateTimeFormat` • `com.unicacorp.interact.flexoffers.defaultDateFormat` • `com.unicacorp.Campaign.interact.offermapping.batchsize` • `com.unicacorp.Campaign.interact.offermapping.service.synctimeout` • `com.unicacorp.interact.cacheTTL` • `com.unicacorp.interact.cacheRefreshIntervalInMin` • `com.unicacorp.interact.enableDTPerfLogging` • `com.unicacorp.interact.compressAPIResponse` • `ignoreSpecialCharacterValidator` • `Interact.CustomStringDelimiter` • `com.unicacorp.interact.playback.maxFilteredIdsCount` • `com.unicacorp.interact.playback.APITimeoutInSecs` **Interact run time** The following are the JVM parameters for Interact run time: • `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 14. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	• `com.unicacorp.interact.testclient.nullValue` • `interact.ehcache.config` • `interact.api.dateFormat` • `com.hcl.interact.testrun.rowlimit` • `Interact.DisableExceptionStackTracesInMacros` • `com.unicacorp.interact.enableDetailStats` • `com.unica.interact.deployment.timeoutInSecs` • `com.ibm.interact.instance.name` • `com.unicacorp.interact.invalidPaths` • `interact.XSessResponseConsumerManager.generateOnlyOneResponse` • `tryToPreserveInexactFloatValues` • `com.unicacorp.interact.propertyRefreshInterval` • `com.unicacorp.interact.scheduledTasksProcessInterval` • `com.unicacorp.interact.eventpatterns.parallelism` • `com.unicacorp.interact.eventpatterns.restartRetries` • `com.unicacorp.interact.eventpatterns.evaluateTimeoutMilli` • `com.unicacorp.interact.eventpatterns.restartRetryDelayInSec` • `Interact.advisoryMessageEncodingOverrides` • `com.unica.interact.api.insertSessionIDAsCookie` • `com.unica.interact.api.SessionIDCookieName` • `InteractMsgCode` • `com.ibm.interact.triggeredmessage.enableJMSConsumer` • `com.unicacorp.interact.maxStringLengthInFormatMacro` • `ContinueEvaluatingBranchAndAdvOptTreatmentLogicDespiteExMessageList` • `DisableDecisionProcessBoxAndAdvOptTreatmentLogging` Valid values are `True` or `False`. The default value is `False`. If you set the value to `False`, when evaluating the expressions of each branch, if the system encounters any errors, it logs the errors. • `Interact.DPB.MuffleErrorsForMutuallyExclusiveBranchesWithNullValues` Valid values are `True` and `False`. The default value is `False`. If you set the value to `False` and if the value of `-DDisableDecisionProcessBoxAndAdvOptTreatmentLogging`

Table 14. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	is also set to `False`, 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. • `TwoDigitYearStartDate` • `Interact.enableTwoDigitYearFix` • `com.ibm.interact.evpatetl.conf` • `com.unicacorp.interact.minTreatmentsPerThread` • `com.unicacorp.interact.maxTreatmentPoolSize` • `CircuitBreaker.processTimeoutMillis` • `com.unicacorp.interact.event.asyncTimeoutMSec` • `com.unicacorp.interact.eventActionTimeout` • `Interact.HTMI.Enabled` • `Interact.HTMI.MaxRequestDurationInMs` • `Interact.HTMI.RecordIndividualAPIs` • `Interact.HTMI.MaxStartSessionDurationInMs` • `Interact.HTMI.MaxGetOffersDurationInMs` • `Interact.HTMI.MaxPostEventDurationInMs` • `Interact.HTMI.MaxGetProfileDurationInMs` • `Interact.HTMI.LogErrorsEveryNthTime` • `Interact.HTMI.UseMillisecondTimers` • `Interact.HTMI.Debug` • `com.unicacorp.interact.suppressWarningOnAnonymousUser` • `com.hcl.interact.eventpatterns.printPatternAction` • `com.hcl.interact.eventpatterns.eagerPersist` • `com.ibm.interact.triggeredmessage.addPerfData` • `com.unicacorp.interact.learning.disableAggregator` • `com.unicacorp.interact.learning.disableDeletion` • `com.unicacorp.interact.learning.ignoreInterval` • `interact.services.loader.saveLoaderFiles` • `ConvertEveryNULLAttributeValueToAJEPNullConstant` • `includeJoinInfo` • `com.unicacorp.interact.deployment.reloadTimeout` • `com.ibm.interact.lockTimeWarningThreshold` • `com.unicacorp.interact.cache.maxWaitTime` • `DEFAULT_PERSISTENCE_PROVIDER` • `com.unicacorp.interact.playback.purgeBatchSize`

Table 14. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	Note: We have removed the JVM parameter `interact.runtime.instance.name`. Use the JVM parameter `com.ibm.interact.instance.name` to set the name of the Interact run-time instance.
Unica Journey	
Unica Plan	• `-Dplan.log4j.config=<plan_home>/conf/plan_log4j.xml` • `-Dplan.log4j.async=true` • `-Dplan.home=<plan_home>` • `-Dplan.slow.query.threshold=<value in milliseconds>` (optional) • `-Dplan.slow.query.show.stack.trace=true` (optional) • `-Dcampaign.https.port=<campaign's https port>` If Plan communicates with Campaign over `https`, and if Campaign is configured to run on the `https` port other than 8443, add this system property to the JVM arguments of the Plan application's JVM.
Unica Platform	• `-DENABLE_PLATFORM_LOG4J_XML_LOGGING=TRUE` • `-DENABLE_PERSISTENT_NAMEID_FORMAT=TRUE` • `-DUNICA_PLATFORM_HOME=<platform_home_directory_path>`
Unica Centralized Offer Management	• `-DOFFER_HOME=<Install_Base_DIR>/CentralizedOffer` • `-DUNICA_PLATFORM_CACHE_ENABLED=false` • `-DUNICA_PLATFORM_LOCAL_CACHE_ENABLED=false` • `-DOffer.isDebug=false` • `-Dlog4j2.formatMsgNoLookups=true`
Unica Segment Central	• `-DSEGMENT_CENTRAL_HOME=<Install_Base_DIR>/SegCentral`

Deploying Unica Plan on WebSphere

You can deploy the Unica Plan application from a WAR file or an EAR file on the WebSphere® Application Server (WAS).

Before you begin

Consider the following points before you deploy Unica Plan on Websphere:

- Verify that your version of WebSphere® meets the requirements that are described in the *Recommended Software Environments and Minimum System Requirements* document, including any necessary fix packs or upgrades.
- You use the WebSphere® Integrated Solutions console to configure your WebSphere® Application Server. The steps that follow provide guidance for setting specific controls.
- Ensure that you add `javax.el-3.0.1-b11.jar` in WAS server lib directory. You can download `javax.el-3.0.1-b11.jar` from https://mvnrepository.com/artifact/org.glassfish/javax.el/3.0.1-b11.

Note: Different versions of WebSphere® Application Server present user interface controls in varying sequences and with alternative labels.

About this task

Complete the following steps to set up the environment for deploying Unica Plan:

1. Define a custom property: On the **Application servers > <servers> > Web container > Custom properties** form, click **New** and enter following values:
   - **Name**:`com.ibm.ws.webcontainer.invokefilterscompatibility`
   - **Value**: `true`
2. Create the JDBC provider: On the **Resources > JDBC > JDBC providers** form, click **New**. Complete the **Create a new JDBC provider** wizard, including the following fields:

   Datasource creation in the web app server can be skipped if you configure through the installer.

   a. For **Implementation type**, select **Connection pool** data source.

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

      Example

      For example, *`db2jcc4.jar / ojdbc6.jar / sqljdbc4.jar / mariadb-java-client-2.4.1 / onedb-jdbc-8.0.0.1-complete.jar`* / *postgresql-42.5.4.jar*.

      Note: PostgreSQL support is not available if you are upgrading to 12.1.7. It is only available for clean installation of 12.1.7.

   c. For MariaDB, provide Implementation class name as `org.mariadb.jdbc.MariaDbDataSource`
3. Create the data source: On the **Resources > JDBC > Data sources** form, click **New**. Complete the Create a data source wizard by completing the following actions:

   Datasource creation in the web app server can be skipped if you configure through the installer.

   a. Specify a data source name.
   b. For JNDI name, enter `plands`.
   c. Select the JDBC provider that you created in step 2.
   d. Specify the database name and server name.
   e. For **Mapping-configuration** alias select **WSLogin**.

4. Define custom properties for the data source: On the **JDBC providers > <database provider> > Data sources > Custom properties** form click **New** to add two properties:
    - **Name**: `user`
    - **Value**: <user_name>
    - **Name**: `password`
    - **Value**: <password>

    If your Unica Plan system tables are in DB2®, locate the `resultSetHoldability` property and set its value to `1`. If this property does not exist, add it.
5. Create new custom properties for MariaDB data source.

    Choose from:

    - Port: <`portnumber`>
    - databaseName: <`database name`>
    - serverName: <`server name`>
    - driverType : 4
6. Configure the JVM: On the **Application servers > <server> > Process definition > Java Virtual Machine** form, identify the Classpath then add the following entries as Generic JVM arguments, separated by a space:
    - `-Dplan.home=`*`<installation_directory>`*`\`*`<Plan_Home>`*

      where *`<Plan_Home>`* is the path to the directory where you installed Unica Plan. Typically this path is `HCL_Unica/Plan`.
    - `-Dclient.encoding.override=UTF-8`
    - Make sure that the following JVM parameters are present. If not add them:
        - For version 12.1.0, 12.1.0.1, 12.1.0.2, and 12.1.0.3, add the following parameters:
            - `-Dlog4j2.contextSelector=org.apache.logging.log4j.core.async.AsyncLoggerContextSelector`
            - `-Dlog4j2.configurationFile=file:///<Plan_home>\conf\plan_log4j.xml (in case of fresh install of 12.1)`
            - `-Dlog4j.configurationFile=file:///<Plan_home>\conf\plan_log4j.xml (in case of fresh install of 12.1)`
        - For version 12.1.0.4 or later, if you are installing, add the following parameters. If you are upgrading from version 12.1.0 or later, remove the parameters mentioned earlier on page 47, and add the following parameters:
            - `-Dlog4j.configuration=file:///<Plan_home>\conf\plan_log4j_1x.xml`
            - `-Dplan.log4j.config=<Plan_home>\conf\plan_log4j.xml`
            - `-Dplan.log4j.async=true (If not set, deafults to true)`

    Note:

    - For version 12.1.0, 12.1.0.1, 12.1.0.2, and 12.1.0.3, if you have customized the `plan_log4j.xml` and made some changes to suit your organization, then ensure that you make those settings in the `plan_log4j.xml` specified here as per the `-Dlog4j2.configurationFile` and the `-Dlog4j.configurationFile` that has been deployed.

- For version 12.1.0.4 or later, if you have customized the `plan_log4j.xml` and made some changes to suit your organization, then ensure that you make those settings in the `plan_log4j.xml` specified here as per the `-Dplan.log4j.config` file that has been deployed.
- For version 12.1.0, 12.1.0.1, 12.1.0.2, and 12.1.0.3, ensure that the `xml` file pointed by `log4j2.configurationFile` and `log4j.configurationFile` is the same and follows the `log4j2` syntax. The file should contain the following two lines at the top:

  ```
   <?xml version="1.0" encoding="UTF-8"?>
  <Configuration packages="com.unica.afc.logger" monitorInterval="60">
  ```
- For version 12.1.0.4 or later, ensure that the `xml` file pointed by `-Dplan.log4j.config` file follows the `log4j2` syntax. The file should contain the following two lines at the top:

  ```
   <?xml version="1.0" encoding="UTF-8"?>
  <Configuration packages="com.unica.afc.logger" monitorInterval="60">
  ```
- See the log4j2 documentation for details by referring to the link: https://logging.apache.org/log4j/2.0/manual/configuration.html
- In Windows, the file path contains a back slash. Whereas, in UNIX, it must contain a forward slash.
- The path separator in the log4j configuration file name should be set as Windows style() or UNIX (/) style based on the underlying operating system.

7. In the `<Plan_home>/conf/` location, the 12.1.7 installer creates a back up of the existing files:

File name before backup	File name after backup
`plan_log4j.xml`	`plan_log4j.xml_pre_12.1.7`
`plan_log4j_client.xml`	`plan_log4j_client.xml_pre_12.1.7`
`plan_log4j_socket_server.xml`	`plan_log4j_socket_server.xml_pre_12.1.7`

   If the backed up files contain custom settings, you must ensure that the custom settings are copied to the newly added configuration files by the 12.1.7 installer.
8. Set the JSP Compilation level in the WebSphere application server to 18.

Deploying the WAR or EAR file

When you deploy a new enterprise application, the WebSphere® Integrated Solutions Console presents a series of forms. The steps that follow provide guidance for setting specific controls on these forms. Different versions of WebSphere® can present controls in another sequence or with other labels.

About this task

Complete the following steps to deploy the WAR or EAR file:

1. Select **Applications > New Application > New Enterprise Application**.
2. On the initial form, select **Remote** file system and then **Browse** to the `plan.war` file or your EAR file.
3. On the next **Preparing for the Application Installation** window:

- Select **Detailed**.
- Select **Generate Default Bindings**.
- Select **Override existing bindings**.

4. On the **Select Installation Options** window, complete the following actions:
   - Select **Precompile JavaServer Pages files**.
   - For **Application name**, enter `plan`.
   - Select **Override class reloading settings for Web and EJB modules**.
   - For **Reload interval in seconds**, enter an integer such as `4`.
5. On the **Map Modules to Servers** window, select the **Module**. If you deployed an EAR, select every WAR file.
6. On the **Provide Options to Compile JSPs** window, select the **Web module**. If you deployed an EAR, select every WAR file.
7. Set the **JDK Source Level** to `18`.
8. On the **JSP reloading options for Web modules** form, select **JSP enable class reloading** and for **JSP reload interval in seconds** enter `5`.
9. On the **Map shared libraries** window, select the **Application** and the **Module**.
10. On the **Map shared library relationships** window, select the **Application** and the **Module**.
11. On the **Map resource references to resources** window, select the module and for **Target Resource JNDI Name** enter `plands`.
12. On the **Map context roots for Web modules** window, for **Context Root** enter `/plan`.
13. Review and save your settings.

Defining class loader policy

Class loader policy defines how the application is configured on WAS. You must change some of the default settings on WAS before you deploy Unica Plan.

About this task

Complete the following steps to define the class loader policy:

1. On the **Enterprise Applications > plan > Class loader**, select **Override class reloading settings for Web and EJB modules**.
2. For **Class loader** order, select **Classes loaded with local class loader first (parent last)**.
3. For **WAR class loader policy**, select **Single class loader for application**.
4. Click **Apply** and **Save Settings**.

Defining settings for cookies

You must define and set settings for cookies by using **Session Management** option on the **Websphere Enterprise Applications**.

About this task

Complete the following steps to define the settings for cookies:

1. Go to the **Enterprise Applications >** ***plan*** **> Session management**.
2. Select **Override session management**.
3. Select **Enable cookies**.
4. Click **Apply**, then go to the **Enterprise Applications >** ***plan*** **> Session management > Cookies**.
5. Change the Unica Plan **Cookie name** from `JSESSIONID` to `UMOSESSIONID`.
6. Click **Apply** and **Save Settings**.

Defining EAR module settings

If you deployed an EAR file, you must define settings for the individual WAR files that are contained by the EAR files.

About this task

Complete the following steps to define EAR file module settings:

1. Go to **Enterprise Applications** and select the EAR file.
2. On the **Manage Modules** form, select one of the WAR files; for example, `MktOps.war`.
3. On the **Enterprise Applications >** ***EAR*** **> Manage Modules > WAR** form, complete the following substeps:
    a. Set **Starting weight** to `10000`.
    b. For **Class loader order**, select **Classes loaded with application class loader first**.
4. On the **Enterprise Applications > EAR > Manage Modules > WAR > Session management**, select **Enable cookies**.
5. On the **Enterprise Applications > EAR > Manage Modules > WAR > Session Management > Cookies**, complete the following steps:
    a. Set the **Cookie name** to `CMPJSESSIONID`.
    b. For the **Cookie maximum age**, select **Current browser session**.
6. On the **Enterprise Applications > EAR > Manage Modules > WAR > Session management**, enter the following information:
    a. Select **Allow overflow**.
    b. For **Maximum in-memory sessions**, enter `1000`.
    c. For **Session timeout**, select **Set timeout** and enter `30`.
7. Define the same settings for each of the other WAR files, including `unica.war` and `plan.war`.

> **Note:** If the `Campaign.war` file also exists in the EAR file and you plan to integrate Unica Plan with Unica Campaign, define the same settings for the `Campaign.war` file.

Deploying Unica Plan on WebLogic

Use the following guidelines for deploying Unica Plan on WebLogic.

About this task

- Unica products customize the JVM used by WebLogic. You might create a WebLogic instance dedicated to Unica products if you encounter JVM-related errors.
- Do not install more than one Unica Plan application under the same WebLogic domain.

- Verify that the Software Developement Kit (SDK) selected for the WebLogic domain you are using is the Sun SDK by looking in the startup script (`startWebLogic.cmd`) for the JAVA_VENDOR variable. It should be set to `JAVA_VENDOR=Sun`. If it is set to `JAVA_VENDOR=BEA`, JRockit has been selected. JRockit is not supported. To change the selected SDK, refer to the WebLogic documentation.

Complete the following steps to deploy Unica Plan to WebLogic:

1. If your operating system is AIX®, extract the Unica Plan `WAR` file, delete the `xercesImpl.jar` file from `WEB_INF/lib` directory, and recreate the `WAR` file.
   If the installer combined products in an `EAR` file, you must first extract that file to obtain the `WAR` file, and you must recreate the `EAR` file.
2. If you are configuring WebLogic to use the IIS plug-in, review the WebLogic documentation to determine whether you have additional requirements.
3. From the `bin` directory under your WebLogic domain directory, locate and open the `setDomainEnv` script in a text editor. Scroll to the `JAVA_OPTIONS` property, and add the following entries. Use a space to separate entries.
   - `-Dplan.home=<HCL_Unica_Home>\<Plan_Home>`

     Where `<HCL_Unica_Home>` is the path to the top-level irectory and `<Plan_Home>` is the path to the directory where you installed Unica Plan. Typically this directory is `HCL_Unica/Plan`.
   - `-Dfile.encoding=UTF-8`
   - Make sure that the following JVM parameters are present. If not add them:
     - For version 12.1.0, 12.1.0.1, 12.1.0.2, and 12.1.0.3, add the following parameters:
       - `-Dlog4j2.contextSelector=org.apache.logging.log4j.core.async.AsyncLoggerContextSelector`
       - `-Dlog4j2.configurationFile=file:///<Plan_home>\conf\plan_log4j.xml (in case of fresh install of 12.1)`
       - `-Dlog4j.configurationFile=file:///<Plan_home>\conf\plan_log4j.xml (in case of fresh install of 12.1)`
     - For version 12.1.0.4 or later, if you are installing, add the following parameters. If you are upgrading from version 12.1.0 or later, remove the parameters mentioned earlier on page 51, and add the following parameters:
       - `-Dlog4j.configuration=file:///<Plan_home>\conf\plan_log4j_1x.xml`
       - `-Dplan.log4j.config=<Plan_home>\conf\plan_log4j.xml`
       - `-Dplan.log4j.async=true (If not set, deafults to true)`

   Note:

   - For version 12.1.0, 12.1.0.1, 12.1.0.2, and 12.1.0.3, if you have customized the `plan_log4j.xml` and made some changes to suit your organization, then ensure that you make those settings in the `plan_log4j.xml` specified here as per the `-Dlog4j2.configurationFile` and the `-Dlog4j.configurationFile` that has been deployed.
   - For version 12.1.0.4 or later, if you have customized the `plan_log4j.xml` and made some changes to suit your organization, then ensure that you make those settings in the `plan_log4j.xml` specified here as per the `-Dplan.log4j.config` file that has been deployed.

- For version 12.1.0, 12.1.0.1, 12.1.0.2, and 12.1.0.3, ensure that the `xml` file pointed by `log4j2.configurationFile` and `log4j.configurationFile` is the same and follows the `log4j2` syntax. The file should contain the following two lines at the top:

  ```
   <?xml version="1.0" encoding="UTF-8"?>
  <Configuration packages="com.unica.afc.logger" monitorInterval="60">
  ```
- For version 12.1.0.4 or later, ensure that the `xml` file pointed by `-Dplan.log4j.config` file follows the `log4j2` syntax. The file should contain the following two lines at the top:

  ```
   <?xml version="1.0" encoding="UTF-8"?>
  <Configuration packages="com.unica.afc.logger" monitorInterval="60">
  ```
- See the log4j2 documentation for details by referring to the link: https://logging.apache.org/log4j/2.0/manual/configuration.html
- In Windows, the file path contains a back slash. Whereas, in UNIX, it must contain a forward slash.
- The path separator in the log4j configuration file name should be set as Windows style() or UNIX (/) style based on the underlying operating system.

Note:

- In Windows, the file path contains a back slash. Whereas, in UNIX, it must contain a forward slash.
- The path separator in the log4j configuration file name should be set as Windows style() or UNIX (/) style based on the underlying operating system.

4. In the `<Plan_home>/conf/` location, the 12.1.7 installer creates a back up of the existing files:

File name before backup	File name after backup
`plan_log4j.xml`	`plan_log4j.xml_pre_12.1.7`
`plan_log4j_client.xml`	`plan_log4j_client.xml_pre_12.1.7`
`plan_log4j_socket_server.xml`	`plan_log4j_socket_server.xml_pre_12.1.7`

   If the backed up files contain custom settings, you must ensure that the custom settings are copied to the newly added configuration files by the 12.1.7 installer.
5. Save and close the file.
6. Restart WebLogic.
7. Deploy Unica Plan as a web application module: select `plan.war`.
8. Start the deployed web application.

What to do next

WebLogic 12.2.1.3 blocks a few HTTP methods, for example DELETE and PATCH, which are used by REST API for the newly added Enhanced workflow. To resolve this issue, apply a WebLogic patch (if it is not applied).

For details about the patch, access https://support.oracle.com/knowledge/Middleware/2331453_1.html. You need to be a registered user to access the link. After sucessful authentication, search for the patch number `26923558`.

Note: The information provided about the patch belongs to a third-party website. HCL Unica will not be responsible if the information is inaccurate.

Deploying Unica Plan on JBoss

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

1. Perform the following steps:
    - Clear the recent items for all the users. Note: Administrator: Please communicate to all the users to clear their respective recent items prior to the migration to JBOSS.
    - Clear the directory `<Plan_home>/recentdata`.
    - Migrate to JBOSS + 12.1.7.

From the `bin` directory under your JBoss installation directory, locate and open the `standalone.conf or standalone.conf.bat` script appropriate for your operating system in a text editor. Scroll to the `JAVA_OPTIONS` property, and add the following entries. Use a space to separate entries.

- `-Dplan.home=<Plan_Home>`

  Where `<HCL_Unica_Home>` is the path to the top-level directory and `<Plan_Home>` is the path to the directory where you installed Unica Plan. Typically this directory is `HCL_Unica/Plan`.
- `-Dfile.encoding=UTF-8`
- Make sure that the following JVM parameters are present. If not add them:
    - For version 12.1.0, 12.1.0.1, 12.1.0.2, and 12.1.0.3, add the following parameters:
        - `-Dlog4j2.contextSelector=org.apache.logging.log4j.core.async.AsyncLoggerContextSelector`
        - `-Dlog4j2.configurationFile=file:///<Plan_home>\conf\plan_log4j.xml (in case of fresh install of 12.1)`
        - `-Dlog4j.configurationFile=file:///<Plan_home>\conf\plan_log4j.xml (in case of fresh install of 12.1)`
    - For version 12.1.0.4 or later, if you are installing, add the following parameters. If you are upgrading from version 12.1.0 or later, remove the parameters mentioned earlier on page 53, and add the following parameters:
        - `-Dlog4j.configuration=file:///<Plan_home>\conf\plan_log4j_1x.xml`
        - `-Dplan.log4j.config=<Plan_home>\conf\plan_log4j.xml`
        - `-Dplan.log4j.async=true (If not set, deafults to true)`

Note:

- For version 12.1.0, 12.1.0.1, 12.1.0.2, and 12.1.0.3, if you have customized the `plan_log4j.xml` and made some changes to suit your organization, then ensure that you make those settings in the `plan_log4j.xml` specified here as per the `-Dlog4j2.configurationFile` and the `-Dlog4j.configurationFile` that has been deployed.

- For version 12.1.0.4 or later, if you have customized the `plan_log4j.xml` and made some changes to suit your organization, then ensure that you make those settings in the `plan_log4j.xml` specified here as per the `-Dplan.log4j.config` file that has been deployed.
- For version 12.1.0, 12.1.0.1, 12.1.0.2, and 12.1.0.3, ensure that the `xml` file pointed by `log4j2.configurationFile` and `log4j.configurationFile` is the same and follows the `log4j2` syntax. The file should contain the following two lines at the top:

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

- For version 12.1.0.4 or later, ensure that the `xml` file pointed by `-Dplan.log4j.config` file follows the `log4j2` syntax. The file should contain the following two lines at the top:

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

- See the log4j2 documentation for details by referring to the link: https://logging.apache.org/log4j/2.0/manual/configuration.html
- In Windows, the file path contains a back slash. Whereas, in UNIX, it must contain a forward slash.
- The path separator in the log4j configuration file name should be set as Windows style() or UNIX (/) style based on the underlying operating system.

- In the `<Plan_home>/conf/` location, the 12.1.7 installer creates a back up of the existing files:

File name before backup	File name after backup
`plan_log4j.xml`	`plan_log4j.xml_pre_12.1.7`
`plan_log4j_client.xml`	`plan_log4j_client.xml_pre_12.1.7`
`plan_log4j_socket_server.xml`	`plan_log4j_socket_server.xml_pre_12.1.7`

If the backed up files contain custom settings, you must ensure that the custom settings are copied to the newly added configuration files by the 12.1.7 installer.

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

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

1. Deploy the HCL plan.war file as an enterprise application.

   For example: deploy `<Plan_home>\plan.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 to JAVA_VENDOR`.

b. Restart JBoss server.

By default JBOSS is configured to process requests with a maximum size of 10 MB. If you expect users to download large files, configure the **max-post-size** as needed.

To increase the value of this parameter, directly edit the standalone configuration or domain configuration as follows:

```
<buffer-cache name="default"/>
           <server name="default-server">
               <http-listener name="default" socket-binding="http" redirect-socket="https"
max-post-size="20971520" enable-http2="true"/>
```

For more information, see *JBOSS documentation*.

Deploying Unica Plan on Apache Tomcat®

You must follow a set of guidelines when you deploy Unica Plan on Apache Tomcat.

Set **removeAbandoned=true** and **removeAbandonedTimeout=300** (5 minutes) for `plands` datasource configuration in `<catalina home>/conf/Catalina/localhost/plan.xml` file within the section that starts with the parameter **Resource name="plands"**.

For example:

```
<Resource name="plands" auth="Application" type="javax.sql.DataSource"
 factory="org.apache.tomcat.jdbc.pool.DataSourceFactory" maxTotal="100" maxIdle="30" maxWaitMillis
 ="10000" username="abc" password="abc" driverClassName="org.mariadb.jdbc.Driver" removeAbandoned="true"
 removeAbandonedTimeout="300" url="jdbc:mysql://lp2-ap-51728467.prod.hclpnp.com:3306/pln121" />
```

Note: If the application executes SQL queries that requires more than `300` seconds of execution time, change the value of **removeAbandonedTimeout**.

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

From the `bin` directory under your Apache Tomcat installation directory, locate or create the `setenv.sh or setenv.bat` script appropriate for your operating system in a text editor, and add or modify `JAVA_OPTIONS` property, and add the following entries. Use a space to separate entries.

- `-Dplan.home=<*Plan_Home*>`

  Where `<*Plan_Home*>` is the path to the directory where you installed Unica Plan.
- `-Dfile.encoding=UTF-8`
- Make sure that the following JVM parameters are present. If not add them:

    - For version 12.1.0, 12.1.0.1, 12.1.0.2, and 12.1.0.3, add the following parameters:
        - `-Dlog4j2.contextSelector=org.apache.logging.log4j.core.async.AsyncLoggerContextSelector`
        - `-Dlog4j2.configurationFile=file:///<Plan_home>\conf\plan_log4j.xml (in case of fresh install of 12.1)`
        - `-Dlog4j.configurationFile=file:///<Plan_home>\conf\plan_log4j.xml (in case of fresh install of 12.1)`
    - For version 12.1.0.4 or later, if you are installing, add the following parameters. If you are upgrading from version 12.1.0 or later, remove the parameters mentioned earlier on page 56, and add the following parameters:
        - `-Dlog4j.configuration=file:///<Plan_home>\conf\plan_log4j_1x.xml`
        - `-Dplan.log4j.config=<Plan_home>\conf\plan_log4j.xml`
        - `-Dplan.log4j.async=true (If not set, deafults to true)`

Note:

- For version 12.1.0, 12.1.0.1, 12.1.0.2, and 12.1.0.3, if you have customized the `plan_log4j.xml` and made some changes to suit your organization, then ensure that you make those settings in the `plan_log4j.xml` specified here as per the `-Dlog4j2.configurationFile` and the `-Dlog4j.configurationFile` that has been deployed.
- For version 12.1.0.4 or later, if you have customized the `plan_log4j.xml` and made some changes to suit your organization, then ensure that you make those settings in the `plan_log4j.xml` specified here as per the `-Dplan.log4j.config` file that has been deployed.
- In Windows, the file path contains a back slash. Whereas, in UNIX, it must contain a forward slash.
- The path separator in the log4j configuration file name should be set as Windows style() or UNIX (/) style based on the underlying operating system.
- Ensure that the parameter `relaxQueryChars` is present in the connector tag in `conf/server.xml` in tomcat

```
<Connector port="7002" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443"
relaxedQueryChars="|,[,],\,`,{,},^" />
```

- In the `<Plan_home>/conf/` location, the 12.1.7 installer creates a back up of the existing files:

File name before backup	File name after backup
`plan_log4j.xml`	`plan_log4j.xml_pre_12.1.7`
`plan_log4j_client.xml`	`plan_log4j_client.xml_pre_12.1.7`
`plan_log4j_socket_server.xml`	`plan_log4j_socket_server.xml_pre_12.1.7`

If the backed up files contain custom settings, you must ensure that the custom settings are copied to the newly added configuration files by the 12.1.7 installer.

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

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

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 setenv.sh file for the respective product instances script in the bin directory under your tomcat instances directory to add -`Dfile.encoding=UTF-8 -Dclient.encoding.override=UTF-8 to JAVA_VENDOR`.
   b. Restart Tomcat.
3. 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.

Chapter 7. Configuring Unica Plan after deployment

After you deploy and start the Unica Plan application, log in and verify the installation. There are some basic configuration steps such as configuring the system user and a test user, setting up email and markup.

Note: You can find more system setup tasks in the *Unica Plan Administrator's Guide*.

In addition, if you plan to use the Unica reporting feature, you must complete the tasks that are described in Installing reports on page 65.

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
- unable to edit flowcharts
- 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 installation

After you install Unica Platform, verify that the installation is successful. The installation is successful if you can see the name of the Unica product in the **Configuration** list on the **Settings** page after you log in to Unica.

About this task

Complete the following steps to verify your Unica Platform installation:

1. Access the Unica URL by using Internet Explorer.

   If you entered a domain when you installed, the URL is the following, where `host` is the machine where the 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. The username for administrator login is `asm_admin`.

   The first time you log in, the default value of user password is `password`. You are asked to change the password. You can enter the existing password, but you must choose a new one. You must change user's password when you verified your Unica Platform installation.

   The default home page is the Dashboard, which is a blank page until dashboards are set up. See the *Unica Platform Installation Guide* for instructions about deploying the Dashboard WAR file.
3. Select **Settings** > **Configuration** and verify that Unica Plan is displayed in the list on the left. Then, expand the Unica Plan section and verify that the **umoConfiguration** category is displayed in the list.
4. Optional: Make the **Configuration Settings** page as your "home" page until you configure the dashboard. That way, a blank page is not displayed each time you log in.

Granting the asm_admin user access to Unica Plan

The default admin user (`asm_admin`) can access the Unica Plan configuration properties automatically. However, there are no default users with access to the Unica Plan application until you configure them.

About this task

Complete the following steps to grant the `asm_admin` user access to Unica Plan:

1. Create a group. For example, select **Settings > User groups > New Group** and set up **Default-Plan-Group**.
2. Assign the **PlanAdminRole** and **PlanUserRole** roles to the group.
3. Assign the **asm_admin** user to the group.
4. Restart the application server.
5. Log back in as **asm_admin**.
6. Verify that you have access to Unica Plan features by accessing **Plan > Plans**.

Configuring the markup option

Unica Plan provides markup tools for making comments on attachments. When Unica Plan users send approvals for review, the approver can enter the comments directly in the electronic file which can be viewed by other users.

Unica Plan provides following markup tools:

- Native Unica Plan markup: The native markup option provides various markup functions that can be applied to files in PDF, HTML, JPG, PNG, GIF, and BMP format. Users can mark up whole websites if a URL is supplied. The comments can then be saved in Unica Plan. Native markup is the default option. It does not require Acrobat to be installed on client machines.
- Adobe™ Acrobat markup: This markup tool requires the installation of Adobe™ Acrobat on each client machine. Users can apply all Acrobat commenting features, and then save the edited PDF in Unica Plan.
- PDF Embed API markup: This markup tool is similar to Adobe™ Acrobat markup but works on Google Chrome, Microsoft Edge, and Safari, and also works with annotations (update, remove, and reply).

  Note: If you create Annotations using Native markup (MCM) or Adobe™ Acrobat markup (SOAP), they will not be visible in PDF Embed API markup. Similarly, if you create Annotations using PDF Embed API markup, they will not be visible in Native markup (MCM) or Adobe™ Acrobat markup (SOAP).

The markup option is a global setting. You cannot enable different markup options for different groups of users.

Note: For more information about the versions of Adobe Acrobat that are supported for your operating system, see the *Recommended Software Environments and Minimum System Requirements* guide.

Configuring native Unica Plan markup

You can enable native Unica Plan markup for review. Enabling native Unica Plan markup disables Adobe™ Acrobat markup.

1. Click **Settings > Configuration > Unica Plan > umoConfiguration > markup**.
2. Click **Edit Settings**.
3. Set the **markupServerType** property to `MCM`.
4. Click **Save**.
5. Restart Unica Plan for the change to take effect.

Configuring the Adobe™ markup option

When you deploy Unica Plan, the system is configured to use the native markup option by default. If you want to use the Adobe™ markup option instead, complete the following steps:

- Select **Settings > Configuration > Plan > umoConfiguration > markup**. Then, configure the markup properties with the following values
- Set **markupServerType** to `SOAP`.
- Set **markupServerURL** to the URL of the Unica Plan host server, including the fully qualified host name and the port where the web application server listens. Use the path format that is shown here, with your values substituted for *<server>* and *<port>*.

  ```
  http://<server>:<port>/plan/services/collabService?wsdl
  ```

Configuration settings enable Adobe™ markup for all users.

Installing and configuring Adobe™ on client machines

For users to effectively use Adobe™ markup, you must install the supported version of Adobe™ Acrobat Professional on each client machine that is used to access Unica Plan.

In addition, users who use the Internet Explorer browser to access Unica Plan must set Internet Explorer preferences to display PDFs in the browser.

Configuring PDF Embed API Markup

The prerequisites for PDF Embed API Markup are as follows:

- An account with Adobe to generate an API key.
- Internet connection to access the site: `https://documentcloud.adobe.com`.

Note: The following procedure contains steps related to a third-party website. HCL Software is not responsible for the accuracy of the steps. If there is a discrepancy, access Adobe's documentation for relevant and up-to-date details.

To generate an Adobe API key, complete the following steps:

1. Open a supported browser and access the site https://developer.adobe,com.
2. Click **Console**.
3. Log in using your Adobe credentials.
4. Select **Create new project**.
5. Select **Add API**.
6. Select **PDF Embed API**.
7. In the **Allowed domain** textbox, type or paste the appropriate domain.
8. Click **Save configured API**.
9. Within Credential details, copy the `API KEY`.

To configure PDF Embed API, complete the following steps:

1. Click **Settings > Configuration > Unica Plan > umoConfiguration > markup**.
2. Click **Edit Settings**.
3. Set the **markupServerType** property to `PDF Embed API`.
4. For the **adobeConsoleKey** parameter, paste the value of the `API KEY`, received in Step 9 of the earlier procedure.
5. Click **Save**.
6. Restart Unica Plan for the change to take effect.

Configuring the email settings

Identify your SMTP server when you install Unica Plan so you can configure the email settings during the installation process.

About this task

Complete the following steps to configure the email settings:

1. Select **Settings > Configuration > Plan > umoConfiguration > email**.
2. Click **Edit Settings**.
3. Set the value of the `notifyEMailMonitorJavaMailHost` property to the machine name or IP address of your organization's SMTP server.
4. Specify a valid email address for the `notifyDefaultSenderEmailAddress` property. The system uses this address to send emails when there is no valid email address available to send a notification email to.
5. Save your changes.

Configuring the integration with Unica Campaign

Unica Plan optionally integrates with Unica Campaign. When Unica Plan and Unica Campaign are integrated, you can use the marketing resource management features in Unica Plan to create, plan, and approve campaigns.

About this task

When Unica Campaign integration is enabled, you can also enable offer integration to complete offer lifecycle management tasks in Unica Plan.

To enable your integration with Unica Campaign, log in to Unica Plan and set the following properties on the **Settings > Configuration** page:

- **Unica > Unica Platform**:
  - Unica Plan - Unica Campaign integration (MO_UC_integration must be enabled)
  - Unica Plan - Offer integration (optional, if Unica Campaign integration is enabled)
- **Unica > Unica Campaign > partitions > partition[n] > server > internal**:
  - MO_UC_integration (Set this option to Yes if you plan to enable any of the following three optional settings)
  - MO_UC_BottomUpTargetCells
  - Legacy_campaigns
  - Unica Plan - Offer integration
- **Unica > Unica Plan > umoConfiguration > campaignIntegration**:
  - defaultCampaignPartition
  - webServiceTimeoutInMilliseconds

For more information, see the *Unica Plan and Unica Campaign Integration Guide*.

Configuring DB2® database for integrated systems

Configure the time parameters in the DB2® database to avoid deadlock situations if you plan to use integrated systems and integrated offers in Unica Plan.

About this task

Complete the following steps to configure DB2® database:

1. Use the DB2® admin utility (`get db cfg`) to review the settings for the **LOCKTIMEOUT** and **DLCHKTIME** parameters.
2. Set the lock timeout period to 10 seconds as follows:

   ```
   update db cfg using LOCKTIMEOUT 10
   ```
3. Set the deadlock check time to 15,000 milliseconds as follows:

   ```
   update db cfg using DLCHKTIME 15000
   ```

Results

Setting deadlock check time prevents deadlock conditions from occurring when multiple users access database tables simultaneously.

Additional configuration for enhanced security

The procedures in this section describe some additional configurations in the web application server. They are optional, but you might want to perform them for enhanced security.

Disabling the X-Powered-By flag

If your organization is concerned that the X-Powered-By flag in the header variable is a security risk, you can disable it using this procedure.

1. If you are using WebLogic, in the administration console, under ***domainName*** **> Configuration > Web Application**, set **X-Powered-By Header** to **X-Powered-By Header will not be sent**.
2. If you are using WebSphere®, perform the following steps.
   a. In the WebSphere® administration console, navigate to **Servers > Server Types > WebSphere application servers >** ***server_name*** **> Web Container Settings > Web container**.
   b. Under **Additional Properties** select **Custom Properties**.
   c. On the Custom Properties page, click **New**.
   d. On the Settings page, create a custom property named `com.ibm.ws.webcontainer.disablexPoweredBy` and set the value to `false`.
   e. Click **Apply** or **OK**.
   f. Click **Save** on the console task bar to save your configuration changes.
   g. Restart the server.

Configuring a restrictive cookie path

In your web application server, you can restrict cookie access to a specific application for enhanced security. If not restricted, cookies are valid across all the deployed applications.

1. If you are using WebLogic, perform the following steps.
    a. Extract the `weblogic.xml` file from the WAR or EAR package where you want to add the restrictive cookie path.
    b. Add the following to the `weblogic.xml` file, where *context-path* is the context path for the deployed application. For Unica applications the context path is typically /unica.

    ```
    <session-descriptor>
        <session-param>
          <param-name>CookiePath</param-name>
          <param-value>/context-path> </param-value>
        </session-param>
    </session-descriptor>
    ```

    c. Rebuild the WAR or EAR file.
2. If you are using WebSphere®, perform the following steps.
    a. In the WebSphere® administration console, navigate to the **Session Manager > Cookie** tab.
    b. Set the **Cookie Path** to the context path of your application.

    For Unica applications the context path is typically /unica.

Activating the Checklist menu

You must perform some cofigurations to activate the Checklist menu in Plan.

About this task

To activate the Checklist menu, complete the following steps:

1. Select **Settings > Plan settings**.
2. Click **Unica Plan Upgrade**.
3. Ensure that **Upgrade Configuration** is checked. If not, select it.
4. Click **Upgrade**.
5. Post upgrade, select **Settings > Plan settings**.
6. Restart the server.
7. Select **Synchronize Menus**.
8. Log out and log back in.
9. To verify, select **Plan** menu. You will see the **Checklist** menu item in the list.

Chapter 8. Installing reports

Unica Plan supports integration with both IBM Cognos and Unica Insights.

IBM Cognos reports

Reporting relies on the following components:

- An IBM® Cognos® installation that meets the requirements that are specified in *Recommended Software Environments and Minimum System Requirements*.
- A set of Unica components that integrate the system with the IBM® Cognos® installation.
- The example reports for the Unica Plan application, which are built with IBM® Cognos® Report Studio.

Unica Platform provides the side of the reporting integration. To finish installing reporting, you run all of the following report package installers on the IBM® Cognos® system:

- HCL
- Unica Platform
- Unica Plan

For information about how to install and set reporting for Unica Plan, and information about the individual components and how they interact with each other, see the IBM Cognos *Reports Installation and Configuration Guide*.

Unica Insights reports

Unica Plan integrates with Unica Insights, which is shipped with Unica Platform. Reporting relies on the following components:

- A set of reports for the Unica Plan application, which are installed with the Unica Plan installer.
- The Unica Insights web application which is shipped with Unica Platform.

For more details on how to install and configure Unica Insights reporting for Unica Plan, see the Unica Platform Installation Guide and Unica Insights Installation and Configuration guide.

Next steps for reporting

After you successfully install the Report feature, use the following guidelines for working further with reports:

- If you configured the system to use the "authenticated per user" mode, ensure that the appropriate users can run the reports from the applications. The easiest way to do this is to assign the default ReportsUser role to the appropriate user groups or users.
- For general information about the Framework Manager data models and Report Studio reports, see the chapter named "Configuring Reporting" in the *Unica Platform Administrator's Guide*. For information about configuring and customizing the Unica Plan reports, see the chapter on "Using Reports" in the *Unica Plan Administration Guide*.
- To use the Cognos® dashboard reports in your dashboards, see the chapter on "Creating and Managing Dashboards" in the *Unica Platform Administrator's Guide*.

Chapter 9. Installing Unica Plan in a cluster

You can install Plan in a cluster by completing additional tasks while installing Plan.

To install Unica Plan in a cluster, follow the instructions from chapters 2 through 7 and supplement those instructions with information from this chapter.

When you install Unica Plan in a cluster, there are many ways to configure the installation. However, the basic process follows.

1. Run the installers on one system: typically, the administration server (or whatever the equivalent is for your application server type).
2. Create and share a file directory to hold the upload files for all the Unica Plan installations.
3. Create the EAR file, and then deploy it on each machine in the cluster.
4. Configure each system to share the same Unica Platform system tables and the same Unica Plan system tables.
5. Configure each system to use the shared file directory.
6. Determine which machine in the cluster should send notifications. Then suppress the notification process on all the other machines.
7. Enable the UMOSESSIONID cookie for all servers in the cluster.
8. Configure `plan_ehcache.xml` for distributed caching of templates and offer folders.

WebSphere® guidelines

If you are installing Unica Plan on a cluster in WebSphere®, complete the extra steps in addition to the steps for installing Unica Plan on WebSphere®.

Preparing the data sources

The data source chapter instructs you to create a database for Unica Plan and then configure a JDBC data source for it in your application server. When you complete these tasks for a cluster on WebSphere®, note the following extra instructions:

- The Unica Plan database must be on a machine that is accessible to all machines in the cluster, but it does not need to be on a machine in the cluster.
- When you configure the JDBC provider, specify the cluster as the scope.

Installing the products

When you follow the instructions for running the installer, be sure that you install Unica Platform and Unica Plan only once, on a machine that is accessible to all machines in the Unica Plan cluster.

You do not need to install the software on each cluster member. Instead, you install it only once, create the EAR, and then deploy the EAR file on each cluster member.

Extra pre-deployment steps

Before you deploy Unica Plan, complete the following tasks in addition to the ones specified in the pre-deployment configuration chapter:

- Share the top-level directory in which Unica Plan is installed. For example, say that Unica Plan is installed in `C:\MktOpsCluster\HCL_Unica\Plan`. In this case, you would share the entire `MktOpsCluster` directory.
- Create and share a folder on the administration server to contain the upload files for Unica Plan. This folder is known as the `Shared_UMO_Artifacts` folder. All cluster members must have full control (read, write, change, and delete) of this folder. Placing this folder below the home directory in the local file system hierarchy is optional.

Extra deployment steps

In addition to the instructions in the deployment chapter, note the following additions:

1. Map modules to servers

When you set the options in the **Select Installation Options** wizard for WebSphere®, select the cluster and the web server when mapping modules to servers.

2. Extra instructions for the Generic JVM property

Remember to configure the Generic JVM property on each machine in the cluster.

The path that you specify in the `plan.home` and other properties must point to the shared installation directory.

You set the following extra parameters for clusters:

- `-DPLAN_CONFIG_GUID=Plan`
- `-Dplan.log.config=\\`*`umoMachine`*`\`*`SharedUnicaHome`*`\Plan\conf\plan_log4j_client.xml`
- `-Dplan.local.log.dir=`*`local_log_dir`* (where `local_log_dir` is the writeable folder on the physical machine where Unica Plan creates logs)
- On the machines that must not send notifications, set the "suppress notifications" parameter as follows:

  ```
  -Dplan.suppressNotifications=true
  ```

  Set this property on every node except one: the one you want to send notifications.
- To use a different `plan_ehcache.xml` file in place of the default file that is defined in the `CONF` directory for a node, set the `-Dplan_ehcache` parameter for that node to specify the location of the file.

Configure session management cookie

You must define the name of the session management cookie that is used by servers in the cluster. To configure the session management cookie:

1. In the WebSphere® console, access the properties for a server in the cluster. Go to the web container settings and open the session management configuration.
2. Enable cookies and specify `UMOSESSIONID` as the cookie name.
3. Save the settings and repeat this procedure for all servers in the cluster.

Extra post-deployment steps

If you are using a plug-in for load balancing, you must complete the following configuration steps:

- For Unica Plan to work effectively in a clustered environment, users must remain on a single node for their entire session. This option for session management and load balancing is known as session affinity. For more information about configuring your installation to use session affinity, refer to the documentation for your application server.

  Note: When a node fails on a system that uses this configuration option, every user session on that node also fails. The load balancer does not, and must not, switch users to another available node, as user authentication applies only to a single node in Unica Plan. Users are prompted to log in again, and in some cases an unexpected error and corresponding data loss can occur.

- Log in to Unica Plan. Select **Settings > Configuration**, and configure the following URL parameters so that all references to the Unica Plan server use the proxy host and port.
  - Plan | navigation | serverURL
  - Plan | umoConfiguration | markup | markupServerURL
  - Plan | umoConfiguration | notifications | notifyPlanBaseURL

WebLogic guidelines

You must complete extra steps if you plan to install Unica Plan on a cluster in WebLogic.

Preparing to install

Before you begin, you must create a WebLogic domain for the cluster. For help with this step, see the WebLogic documentation.

Preparing the data sources

The data source chapter instructs you to create a database for Unica Plan and then configure a JDBC data source for it in your application server. When you complete these tasks for a cluster, note the following extra instructions:

- You must configure the web application servers to use the correct JDBC driver on all machines in the cluster.
- Create the data source for the Unica Platform system tables (UnicaPlatformDS) on both the administration server and the cluster members.
- When you create the data source for the Unica Plan system tables (plands), deploy it to the cluster and not to the administration server. Select **All servers in the cluster**.

Installing the products

When you run the installer, be sure that you install Unica Platform and Unica Plan once on the machine that is designated as the administration server for the cluster. You do not need to install the software on each cluster member. Instead, you install it once on the administration server, create the EAR, and then deploy the EAR file on each cluster member.

Pre-deployment steps

Before you deploy Unica Plan, complete the following tasks in addition to the ones specified in the pre-deployment configuration chapter:

- Share the top-level directory in which Unica Plan is installed. For example, say that Unica Plan is installed in the following directory: `C:\PlanCluster\HCL_Unica\Plan`. In this case, you would share the entire `MktOpsCluster` directory.
- Create and share a folder on the administration server to contain the upload files for Unica Plan. This folder is known as the `Shared_UMO_Artifacts` folder. All cluster members must have full control (read, write, change, and delete) of this folder. Placing this folder below the home directory in the local file system hierarchy is optional.

Deploying the application on WebLogic

In addition to the instructions in the deployment chapter, note the following additions:

1. Set Source accessibility option

When you deploy the EAR on the administration server, set the **Source accessibility** option to **Use the defaults defined by the deployment's targets**.

2. Extra instructions for setting the **JAVA_OPTIONS**

Remember to configure the **JAVA_OPTIONS** property in the `setenv` file on each machine in the cluster.

The path that you specify in the `plan.home` property must point to the shared installation directory.

There are two extra parameters to set for clusters:

- `-DPLAN_CONFIG_GUID=Plan`
- On the machines that must not send notifications, set the "suppress notifications" parameter as follows:

  ```
  -Dplan.suppressNotifications=true
  ```

  Verify that the `suppressNotifications` property is set to `false` on the machine that is designated to send notifications. Set this property to `true` on every other machine.

3. Define an alternate `ehcache` file

The `plan_ehcache.xml` file that is defined in the `CONF` directory is used for every node in the cluster. To override this default file on a node, edit `startWeblogic.cmd` (Windows™) or `startWeblogic.sh` (UNIX™) on that node to configure the JAVA_OPTIONS property: add the `-Dplan_ehcache` parameter to specify the location of another `plan_ehcache.xml` file.

Configuring the session management cookie

To define the name of the session management cookie that is used by servers in the cluster, you edit the `plan.war` file that is created by the installer and deployed on your application server.

Complete the following steps to configure the session management cookie:

1. Open a Command Prompt and verify that the Java™ version is the same as the JRE used by Unica Plan: type `java -version`.
2. Copy `plan.war` into a temporary folder and rename the original `plan.war` file.
3. Unpack the contents of the new, temporary `plan.war` archive: type `jar -xvf plan.war`.
4. Delete the unpacked `plan.war`: type `rm plan.war`.
5. Change to the `WEB-INF` directory: type `cd WEB-INF`.

6. Edit the `web.xml` file to add this tag to override the cookie name:

   ```
   <init-param>
      <param-name>CookieName</param_name>
      <param-value>UMOSESSIONID</param-value>
   </init-param>
   ```

7. Repack the `plan.war`: type `cd ..` then `jar -cvf * plan.war`.
8. Copy the updated `plan.war` back to its original location on the server.
9. Deploy the updated `plan.war`.

Post-deployment steps

If you are using a plug-in for load balancing, complete the following configuration steps:

- For Unica Plan to work effectively in a clustered environment, users must remain on a single node for their entire session. This option for session management and load balancing is known as sticky sessions or sticky load balancing. For more information about configuring your installation to use this option, see the documentation for your application server.

  Note: When a node fails on a system that uses this configuration option, every user session on that node also fails. The load balancer must switch users to another available node, as user authentication applies only to a single node in Unica Plan. Users are prompted to log in again, and in some cases an unexpected error and corresponding data loss can occur.

- Log in to Unica Plan and select **Settings > Configuration**. Configure the following URL parameters so that all references to the Unica Plan server use the proxy host and port:
  - Plan | navigation | serverURL
  - Plan | umoConfiguration | markup | markupServerURL
  - Plan | umoConfiguration | notifications | notifyPlanBaseURL

Configuring the shared folder properties

The `Shared_UMO_Artifacts` folder is a shared folder, and is created before you deploy Unica Plan. After successfully deploying Unica Plan, make sure that the all the upload files point to the subfolders in the `Shared_UMO_Artifacts` folder.

About this task

Complete the following steps to set the shared folder properties:

1. Log in and select **Settings > Configuration**.
2. Select **Plan > umoConfiguration > templates**.
3. Click **Edit Settings** and then update the value for the `templatesDir` property to point to a subfolder in the `Shared_UMO_Artifacts` folder.
4. Save your changes
5. Select **Plan > umoConfiguration > attachmentFolders**.

6. Click **Edit Settings** and then update the values for all the properties in this category to point to subfolders in the `Shared_UMO_Artifacts` folder.
7. Save your changes.

Configure `ehcache`

The `ehcache` is an open source Java™ distributed cache for caching, Java™ EE, and light-weight containers. Every node in the cluster can use the same `plan_ehcache.xml` file or you can set a different `plan_ehcache.xml` file for each node. For installations in a cluster, edit the `plan_ehcache.xml` file so you do not have restart the computer if you make changes to templates or offer folders.

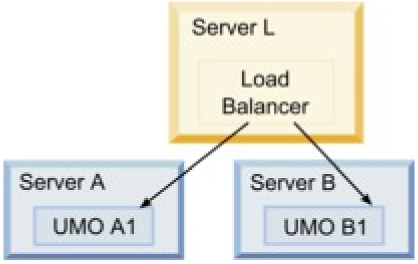
Important: If your installation is upgraded from a previous version, some or all of the sections in the `plan_ehcache.xml` file might not be present. In that case, add and edit the file as shown in the sections that follow.

Note: You must provide a unique name if not already present like `<ehcache updateCheck="false" name="PlanEhCacheManager">`.

Use one of the following procedures to configure the `ehcache` file.

Configuring `ehcache` with Remote Method Invocation (RMI)

Unica Plan systems that have the following topography typically use RMI:



Clustered topography 1:
Configure ehcache with RMI

Go to the `<HCL_Unica_Home>\<Plan_Home>\conf` directory and open the `plan_ehcache.xml` file in a text editor. Then, make the following edits:

- Uncomment the following section of the file.

  You must customize the items that are shown in bold (machineA, machineB, and port) to reflect your environment. Provide a pipe-separated list of all machines in the cluster, by using fully qualified host names.

```
<!--
<cacheManagerPeerProviderFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"
properties="peerDiscovery=manual,
rmiUrls=//<ServerA>:40000/planApplicationCache|//<ServerB>:
40000/planApplicationCache"/>
```

```
<cacheManagerPeerListenerFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"
properties="port=40000, socketTimeoutMillis=20000"/>
-->
```

- Uncomment the following section of the file.

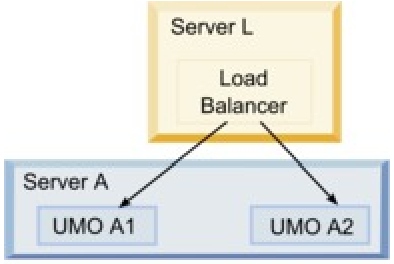
```
<!--
<cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
properties="replicateAsynchronously=true, replicatePuts=true,
replicateUpdates=true, replicateUpdatesViaCopy=true,
replicateRemovals=true"/>
<cacheEventListenerFactory
class="com.unicacorp.uap.common.cache.PlanCacheEventListenerFactory
"net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory" />
-->
```

- If the following lines are present in the file, remove them.

```
<bootstrapCacheLoaderFactory class=net.sf.ehcache.distribution.
RMIBootstrapCacheLoaderFactory"/>
```

Configuring `ehcache` with multicast

Unica Plan systems that have the following topography typically use multicast:



Clustered topography 2:
Configure ehcache with Multicast

Go to the *<HCL_Unica_Home>**<Plan_Home>*\conf directory and open the `plan_ehcache.xml` file in a text editor. Then, make the following edits:

- Uncomment the following section of the file.

  You must customize the items that are shown in bold (multicastGroupAddress and multicastGroupPort) to reflect the multicast group and port in your environment.

```
<!--<cacheManagerPeerProviderFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"
properties="peerDiscovery=automatic, multicastGroupAddress=230.0.0.1,
multicastGroupPort=4446, timeToLive=32"/>

<cacheManagerPeerListenerFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"/>
-->def
```

- Uncomment the following section of the file.

```
<!--
<cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"
properties="replicateAsynchronously=true, replicatePuts=true,
replicateUpdates=true, replicateUpdatesViaCopy=true,
replicateRemovals=true"/>
<cacheEventListenerFactory
class="com.unicacorp.uap.common.cache.PlanCacheEventListenerFactory" />
-->
```

- If the following lines are present in the file, remove them.

```
<bootstrapCacheLoaderFactory
class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory"/>
```

Chapter 10. Uninstalling Unica Plan

Run the Unica Plan uninstaller to uninstall Unica Plan. 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 an 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 Plan must run the uninstaller.

1. If you have deployed the Unica Plan web application, undeploy the web application from your web application server.
2. Shut down web application server.
3. Stop the processes that are related to Unica Plan.
4. Complete one of the following steps to uninstall Unica Plan:

   Choose from:

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

     Uninstall_`Product` -i console
   - In a command-line window, navigate to the directory where the uninstaller exists, and run the following command to uninstall Unica Plan by using the silent mode:

     Uninstall_`Product` -i silent

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

Chapter 11. 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 15. 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`

Table 15. Product names for configTool registration and unregistration (continued)

Product name	Name used in configTool
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"
  ```

Chapter 12. Unica Plan configuration properties

The Unica Plan configuration properties on available on the **Settings > Configuration** page. For more details on the configuration properties, see the Plan Administrator Guide.

Unica Plan

Properties in this category specify the default and supported locales for your installation of Unica Plan.

supportedLocales

Description

Specifies the locales available in your installation of Unica Plan. List only the locales that you are using. Each locale you list uses memory on the server. The amount of memory that is used depends on the size and number of templates.

If you add locales after the initial installation or upgrade, you must run the upgrade servlets again. See upgrade documentation for details.

If you change this value, you must stop and restart your Unica Plan deployment before the change takes effect.

Default value

`en_US`

defaultLocale

Description

Specifies the supported locale in which you want Unica Plan to display for all users, unless explicitly overridden for specific users by Unica Plan administrators.

If you change this value, you must stop and restart your Unica Plan deployment before the change takes effect.

Default value

`en_US`

Unica Plan | navigation

The properties in this category specify options for navigation, such as Uniform Resource Identifiers, URLs, and ports.

welcomePageURI

Description

The Uniform Resource Identifier of the Unica Plan index page. This value is used internally by Unica applications. Changes to this value are not recommended.

Default value

`affiniumPlan.jsp?cat=projectlist`

projectDetailpageURI

Description

The Uniform Resource Identifier of the Unica Plan detail page. This value is used internally by Unica applications. Changes to this value are not recommended.

Default value

blank

seedName

Description

Used internally by Unica applications. Changes to this value are not recommended.

Default value

`Plan`

type

Description

Used internally by Unica applications. Changes to this value are not recommended.

Default value

`Plan`

httpPort

Description

The port number that is used by the application server for connections to the Unica Plan application.

Default value

`7001`

httpsPort

Description

The port number that is used by the application server for secure connections to the Unica Plan application.

Default value

`7001`

serverURL

Description

The URL of the Unica Plan installation. Accepts locators with either the HTTP or HTTPS protocol.

If users access Unica Plan with the Chrome browser, use the fully qualified domain name (FQDN) in the URL. If the FQDN is not used, the Chrome browser cannot access the product URLs.

Default value

`http://<server>:<port>/plan`

Note: <server> must be lowercase.

logoutURL

Description

Used internally. Changes to this value are not recommended.

Unica Platform uses this value to call the logout handler of each registered application if the user clicks the logout link in suite.

Default value

`/uapsysservlet?cat=sysmodules&func=logout`

displayName

Description

Used internally.

Default value

`Unica Plan`

serverURLInternal

Description

This is a new parameter and it specifies the internal URL of the Unica Plan server. If you are using reverse proxy or a web access management software (ISAM, Siteminder,etc), set this parameter to the internal URL of Unica Plan or else set its value to the value of serverURL (Affinium|Plan|navigation|serverURL)

This value is used by other Unica applications when they consume Plan's new rest APIs v2, for example Offer Approval in Centralized Offer Management.

Default value

`http://<server>:<port>/plan`

Unica Plan | about

The configuration properties in this section list information about your Unica Plan installation. You cannot edit these properties.

displayName

Description

The display name of the product.

Value

Unica Plan

releaseNumber

Description

The currently installed release.

Value

<version>.<release>.<modification>

copyright

Description

The copyright year.

Value

<year>

os

Description

The operating system on which Unica Plan is installed.

Value

<operating system and version>

java

Description

The current version of Java™.

Value

<version>

support

Description

Read documentation and place service requests.

Value

`https://hclpnpsupport.hcltech.com/csm`

appServer

Description

The address of the application server on which Unica Plan is installed.

Value

<IP address>

otherString

Description

Value

blank

Unica Plan | umoConfiguration

These properties specify information about the basic configuration of Unica Plan.

serverType

Description

Application Server Type. Used for Calendar export.

Valid values

`WEBLOGIC`, `WEBSPHERE`, `JBOSS`, `TOMCAT`

Default value

Websphere

DBType

Description

Database Type. Indicates on which type of database the system tables are stored.

Valid values

DB2, ORACLE, SQLSERVER, MARIADB

Note: PostgreSQL support is not available if you are upgrading to 12.1.7. It is only available for clean installation of 12.1.7.

Default value

DB2

usermanagerSyncTime

Description

Time in milliseconds to between scheduled synchronizations with Unica Platform.

Default value

`10800000` (milliseconds: 3 hours)

encodeCSV

Description

When the value of this setting is true, when the TCS grid in a Campaign integrated project is exported, the values of strings in cells are enclosed in quotes. When such data is imported and if there is a formula in a string, it is prevented from running when user visits the TCS grid page

Valid values

`True | False`

Default value

`False`

firstMonthInFiscalYear

Description

Set to the month that you would like your account fiscal year to begin. The Summary tab for the account contains a view-only table, which lists budget information by month for the fiscal years of the account. The first month in this table is determined by this parameter.

January is represented by 0. To have your fiscal year to begin in April, set **firstMonthInFiscalYear** to 3.

Valid values

Integers `0` to `11`

Default value

`0`

maximumItemsToBeRetainedInRecentVisits

Description

The maximum number of links to recently viewed pages to display on the **Recent** menu.

Default value

`10` (links)

maxLimitForTitleString

Description

The maximum number of characters that can display in a page title. If titles are longer than the specified number, Unica Plan clips them.

Default value

`40` (characters)

maximumLimitForBulkUploadItems

Description

The maximum number of attachments you can upload at the same time.

Default value

`5` (attachments)

workingDaysCalculation

Description

Controls how Unica Plan calculates durations.

Valid values

- `bus`: Business days only, includes working days only. Does not include weekends and days off.
- `wkd`: Business days + Weekends, includes working days and weekends. Does not include days off.
- `off`: Business days + Days off, includes all working days and days off. Does not include weekends.
- `all`: includes all days in the calendar.

Default value

`all`

validateAllWizardSteps

Description

When users create a program, project, or request with the wizard, Unica Plan automatically validates that the required fields on the current page have values. This parameter controls whether Unica Plan validates the required fields on all pages (tabs) when a user clicks **Finish**.

Valid values

- `True`: Unica Plan checks the required fields on pages that the user did not view (except workflow, tracking, and attachments). If a required field is blank, the wizard opens that page and displays an error message.
- `False`: Unica Plan does not validate required fields on pages the user did not view.

Default value

`True`

enableRevisionHistoryPrompt

Description

Ensures that users are prompted to add change comments when they save a project, request, or approval.

Valid values

`True | False`

Default value

`False`

useForecastDatesInTaskCalendar

Description

Specifies the type of dates that are used when tasks display in calendar view.

Valid values

- `True`: uses forecast and actual dates to display tasks.
- `False`: uses target dates to display tasks.

Default value

`False`

copyRequestProjectCode

Description

Controls whether you want to carry the Project Code (PID) over from a request to a project. If you set this parameter to `False`, the project and the request use different codes.

Valid values

`True | False`

Default value

`True`

projectTemplateMonthlyView

Description

Controls whether the monthly view is allowed in the workflow for a project template.

Valid values

`True | False`

Default value

`False`

disableAssignmentForUnassignedReviewers

Description

Specifies how work is assigned by role for approvals. The **disableAssignmentForUnassignedReviewers** parameter controls the behavior of **Assign work by Role** on the People tab for assignment of approvers in workflow approvals.

Valid values

- `True`: unassigned reviewers in the People tab are not added to the approval as new steps.
    - Append option: The existing, owner-assigned approvers without an assigned role do not change. New approver steps are not added even if the People tab has reviewers with the role "unassigned."
    - Replace option: The existing owner assigned approvers without a role are replaced with a blank. New approver steps would not be added even if the people tab has reviewers with the role "unassigned."
- `False`: unassigned reviewers are added to the approval.
    - Append option: All reviewers without a role are appended to the approval as reviewers if the approval has owner assigned steps without defined roles.
    - Replace Option: The existing approvers of approvals are replaced with the unassigned approvers in the People tab.

Default value

`False`

enableApplicationLevelCaching

Description

Indicates whether application-level caching is enabled or not. For best results in a clustered environment on which multicasting of caching messages is not enabled, consider turning off application level caching for Unica Plan.

Valid values

`True | False`

Default value

`True`

customAccessLevelEnabled

Description

Determines whether you use custom access levels (project roles) in Unica Plan.

Valid values

- `True`: user access to projects and requests is evaluated according to Object Access Levels and Custom Access Levels (project roles). Tab security is enabled for custom tabs.
- `False`: user access to projects and requests is evaluated according only to Object Access Levels (object implicit roles), and tab security is turned off for custom tabs.

Default value

`True`

enableUniqueIdsAcrossTemplatizableObjects

Description

Determines whether you use unique internal IDs for all of the objects that are created from templates, including programs, projects, plans, and invoices.

Valid values

- `True` enables unique internal IDs across all objects that are created from templates. This configuration simplifies cross-object reporting by allowing the system to use the same table for different object types.
- `False` disables unique internal IDs across all objects that are created from templates.

Default value

`True`

FMEnabled

Description

Enables and disables the Financial Management Module, which determines whether the Accounts, Invoices, and Budget tabs appear in the product.

Valid values

`True | False`

Default value

`False`

FMProjVendorEnabled

Description

Parameter that is used to show/hide vendor column for project line items.

Valid values

`True | False`

Default value

`False`

FMPrgmVendorEnabled

Description

Parameter that is used to show/hide vendor column for program line items.

Valid values

```
True | False
```

Default value

```
False
```

Marketing Object Default Report Type

Description

Define the default report type for custom marketing objects including offers. If you change the value of the parameter, you must restart the application.

Valid values

```
Revision History | Cross-Refrence
```

Default value

```
Revision History
```

enablePlanAssetPickerIntegration

Description

Set `True` to enable Unica Content Integration to integrate with Unica Plan

Valid values

```
True | False
```

Default value

```
False
```

Unica Plan | umoConfiguration | Approvals

These properties specify options for approvals.

specifyDenyReasons

Description

Enables a customizable list of reasons for denying an approval. When enabled, administrators populate the Approval Deny Reasons list with options, then associate deny reasons with each workflow template and each project template that defines a workflow. Users who deny an approval, or an item within an approval, are required to select one of these predefined reasons.

Valid values

```
True | False
```

Default value

```
False
```

approveWithChanges

Description

Enables the **Approve with changes** option for an approval. When enabled, The **Allow approvers to approve with changes** option is selected by default when a user sets up approvals in a project template, project, or a standalone approval. The **Allow approvers to approve with changes** option can be edited if the **overrideApproveWithChanges** property is set to `True`.

An approver can approve a task by selecting the **Approve with changes** option if the **Allow approvers to approve with changes** option is selected when the approval is set up.

Valid values

```
True | False
```

Default value

```
True
```

overrideApproveWithChanges

Description

Set to `True` to allow a user to edit the default setting for the **Allow approvers to approve with changes** option when a user sets up approvals in a project template, project, or a standalone approval. The default setting is determined by the **approveWithChanges** property.

Valid values

```
True | False
```

Default value

```
True
```

Unica Plan | umoConfiguration | templates

These properties specify information about templates in Unica Plan. For best results, do not change the default values of these parameters.

templatesDir

Description

Identifies the directory that contains all of your project template definitions, which are stored in XML files.

Use a fully qualified path.

Default value

```
<HCL_Unica_Home>/<Plan_Home>/templates
```

assetTemplatesFile

Description

The XML file that defines the templates for assets. This file must be in the directory that is specified by **templatesDir**.

Default value

```
asset_templates.xml
```

planTemplatesFile

Description

The XML file that defines the templates for plans. This file must be in the directory that is specified by **templatesDir**.

Default value

```
plan_templates.xml
```

programTemplatesFile

Description

The XML file that defines the templates for programs. This file must be in the directory that is specified by **templatesDir**.

Default value

```
program_templates.xml
```

projectTemplatesFile

Description

The XML file that defines the templates for projects. This file must be in the directory that is specified by **templatesDir**.

Default value

```
project_templates.xml
```

invoiceTemplatesFile

Description

The XML file that defines the templates for invoices. This file must be in the directory that is specified by **templatesDir**.

Default value

```
invoice_templates.xml
```

componentTemplatesFile

Description

The XML file that defines the templates for custom marketing object types. This file must be in the directory that is specified by **templatesDir**.

Default value

```
component_templates.xml
```

metricsTemplateFile

Description

The XML file that defines the templates for metrics. This file must be in the directory that is specified by **templatesDir**.

Default value

```
metric_definition.xml
```

teamTemplatesFile

Description

The XML file that defines the templates for teams. This file must be in the directory that is specified by **templatesDir**.

Default value

```
team_templates.xml
```

offerTemplatesFile

Description

The XML file that defines the templates for offers. This file must be in the directory that is specified by **templatesDir**.

Default value

```
uap_sys_default_offer_comp_type_templates.xml
```

Unica Plan | umoConfiguration | attachmentFolders

These properties specify directories to upload and store attachments.

uploadDir

Description

The upload directory where attachments for projects are stored.

Default value

```
<Plan_Home>/projectattachments
```

planUploadDir

Description

The upload directory where attachments for plans are stored.

Default value

<Plan_Home>/planattachments

programUploadDir

Description

The upload directory where attachments for programs are stored.

Default value

<Plan_Home>/programattachments

componentUploadDir

Description

The upload directory where attachments for marketing objects are stored.

Default value

<Plan_Home>/componentattachments

taskUploadDir

Description

The upload directory where attachments for tasks are stored.

Default value

<Plan_Home>/taskattachments

approvalUploadDir

Description

The upload directory where approval items are stored.

Default value

<Plan_Home>/approvalitems

assetUploadDir

Description

The upload directory where assets are stored.

Default value

<Plan_Home>/assets

accountUploadDir

Description

The upload directory where attachments for accounts are stored.

Default value

`<Plan_Home>/accountattachments`

invoiceUploadDir

Description

The upload directory where attachments for invoices are stored.

Default value

`<Plan_Home>/invoiceattachments`

graphicalRefUploadDir

Description

The upload directory where attribute images are stored.

Default value

`<Plan_Home>/graphicalrefimages`

templateImageDir

Description

The upload directory where template images are stored.

Default value

`<Plan_Home>/images`

recentDataDir

Description

The temporary directory that stores the recent data (serialized) for each user.

Default value

`<Plan_Home>/recentdata`

workingAreaDir

Description

The temporary directory that stores CSV files that are uploaded during grid imports.

Default value

`<Plan_Home>/umotemp`

managedListDir

Description

The upload directory where managed list definitions are stored.

Default value

`<Plan_Home>/managedList`

Unica Plan | umoConfiguration | fileUpload

The properties in this category specify options for file uploads.

validateFileUpload

Description

You can choose `True` to validate a file that is uploaded or `False` if you do not want to validate a file that is uploaded.

allowedFileTypes

Description

The type of file allowed to be uploaded. File types can include `.doc`, `.ppt`, `.xls`, `.pdf`, `.gif`, `.jpeg`, `.png`, and `.mpp`.

fileMaxSize

Description

The maximum size allowed for the file that you upload.

Unica Plan | umoConfiguration | Email

These properties specify information for sending email notifications in Unica Plan.

notifyEMailMonitorJavaMailHost

Description

Optional string that specifies either the DNS host name of the email notifications mail server or its dot-formatted IP address. Set to the machine name or IP address of your SMTP server.

This parameter is necessary if you did not provide Unica Plan with an existing JavaMail™ session that uses the session parameter and the delegate is marked "Complete."

Default value

`[CHANGE-ME]`

notifyDefaultSenderEmailAddress

Description

Set to a valid email address. The system sends email messages to this address when there is no valid email address available to send the notification email messages.

Default value

`[CHANGE-ME]`

notifySenderAddressOverride

Description

Use this parameter to specify a standard value for the REPLY-TO and FROM email addresses for notifications. By default, these addresses are populated with the email address of the event owner.

Default value

blank

Unica Plan | umoConfiguration | markup

These properties specify markup options. Unica Plan provides markup tools for making comments on attachments. You can use either Adobe™ Acrobat markup or native Unica Plan markup. Use the properties in this category to configure which option to use.

markupServerType

Description

Determines which markup option to use.

Valid values

- `SOAP` enables users to edit and view markups in PDF documents. Adobe™ Acrobat Professional is required for markups. If specified, users cannot view markups that were made previously in a web browser with the native Unica Plan method.

  If you specify `SOAP`, you must also configure the **markupServerURL** parameter.

  If you specify `SOAP`, you must delete the customized `UMO_Markup_Collaboration.js` that was copied in the JavaScripts subdirectory of the directory where Adobe Acrobat is installed. For example: `C:\Program files (x86)\Adobe\Acrobat 10.0\Acrobat\Javascripts\UMO_Markup_Collaboration.js`. This file is no longer required.
- `MCM` enables the native Unica Plan markup method that allows users to edit and view markups in a web browser. If specified, users cannot edit or view markups that were made previously in a PDF with Adobe™ Acrobat.
- `PDF Embed API` enables the PDF Embed API. You can use the feature to provide feedback and work with annotations.
- If blank, the markup function is disabled and the **View/Add Markup** link does not appear.

Default value

`MCM`

markupServerURL

Description

Dependent on **markupServerType** = `SOAP`.

Set to the URL for the computer that hosts the markup server, including the number of the port the web application server uses for listening. The URL must contain the fully qualified host name.

Accepts locators with either the HTTP or HTTPS protocol.

Default value

`http://<server>:<port>/plan/services/collabService?wsdl`

instantMarkupFileConversion

Description

If `True`, Unica Plan converts PDF attachments to images as soon as they are uploaded, rather than doing this conversion the first time a user opens the item for markup.

Valid values

`True | False`

Default value

`False`

adobeConsoleKey

Description

The console key generated from the `http://developer.adobe.com` website. For more details, see the *Unica Plan Administrator's Guide*. This field is applicable only if you select `PDF Embed API` for the markupServerType on page 97 parameter.

Default value

-

Unica Plan | umoConfiguration | grid

These properties specify options for grids.

gridmaxrow

Description

An optional integer to define the maximum number of rows to be retrieved in grids. The default, `-1`, retrieves all rows.

Default value

`-1`

reloadRuleFile

Description

An optional boolean parameter that indicates whether the grid validation plug-in needs to be reloaded or not.

Valid values

`True | False`

Default value

`True`

gridDataValidationClass

Description

An optional parameter to specify custom grid data validation class. If not specified, the default, the built in plug-in is used for grid data validation.

Default value

blank

tvcDataImportFieldDelimiterCSV

Description

Delimiter to use to parse data imported into a grid. Default is comma (,).

Default value

`,` (comma)

maximumFileSizeToImportCSVFile

Description

Represents the maximum file size in MB that can be uploaded while importing comma-separated data for TVC.

Default value

`0` (unlimited)

maximumRowsToBeDisplayedPerPageInGridView

Description

Specifies the number of rows to display per page in grid view.

Valid values

positive integers

Default value

```
100
```

griddataxsd

Description

Name of grid data XSD file.

Default value

```
griddataschema.xsd
```

gridpluginxsd

Description

Name of grid plug-ins XSD file.

Default value

```
gridplugin.xsd
```

gridrulesxsd

Description

Name of grid rules XSD file.

Default value

```
gridrules.xsd
```

Unica Plan | umoConfiguration | workflow

These properties specify options for the workflow in Unica Plan.

hideDetailedDateTime

Description

Optional show/hide parameter for detailed date time in the tasks page.

Valid values

```
True | False
```

Default value

```
False
```

daysInPastRecentTask

Description

This parameter determines how long tasks are considered "recent." If the task is "active" and started less than this number of days ago, or the Target End Date of the task is between today and this number of days in the past, the task displays as a recent task.

Valid values

positive integers

Default value

`14` (days)

daysInFutureUpcomingTasks

Description

This parameter determines how many days in the future to look for upcoming tasks. If the task starts in the next **daysInFutureUpcomingTasks** or does not end before the current date, it is an upcoming task.

Valid values

positive integers

Default value

`14` (days)

beginningOfDay

Description

Begin hour of the working day. This parameter is used to calculate the datetimes in workflow using fractional durations.

Valid values

integers from `0` to `12`

Default value

`9` (9 AM)

numberOfHoursPerDay

Description

Number of hours per day. This parameter is used to calculate the datetimes in workflow using fractional durations.

Valid values

integers from `1` to `24`

Default value

`8` (hours)

mileStoneRowBGColor

Description

Defines the background color for workflow tasks. To specify the value, insert the # character before the six-character Hex code for the color. For example, #0099CC.

Default value

`#DDDDDD`

UI

Description

Defines the version of the workflow UI. The possible values are `Enhanced` to show the modern UI, or `Classic` to show UI as in version before 12.1. This is a system-level setting and not a user-level setting or project-level setting. If you modify this setting, you must restart the Plan application.

Default value

Enhanced

Unica Plan | umoConfiguration | integrationServices

These properties specify information about Unica Plan Integration Services module. The Integration Services module extends the function of Unica Plan with web services and triggers.

enableIntegrationServices

Description

Enables and disables the Integration Services module.

Valid values

`True | False`

Default value

`False`

integrationProcedureDefinitionPath

Description

Optional full file path to the custom procedure definition XML file.

Default value

`[plan-home]``/devkits/integration/examples/src/procedure/procedure-plugins.xml`

integrationProcedureClasspathURL

Description

URL to the class path for custom procedures.

Default value

`file:///[plan-home]/devkits/integration/examples/classes/`

Unica Plan | umoConfiguration | campaignIntegration

The properties in this category specify options for the Unica Campaign integration.

defaultCampaignPartition

Description

When Unica Plan is integrated with Unica Campaign, this parameter specifies the default Unica Campaign partition if the campaign-partition-id is not defined in the project template.

Default value

`partition1`

webServiceTimeoutInMilliseconds

Description

Added for Web Service integration API calls. This parameter is used as a timeout for web services API calls.

Default value

`1800000` milliseconds (30 minutes)

Unica Plan | umoConfiguration | reports

These properties specify information about reports that are used by Unica Plan.

reportsAnalysisSectionHome

Description

Indicates the home directory for the Analysis Section reports. The value for Cognos reports must be `/content/folder[@name='Affinium Plan']`. The value for Unica Insights reports must be Plan/Affinium Plan.

Default value

`/content/folder[@name='Affinium Plan']`

reportsAnalysisTabHome

Description

Indicates the home directory for the Analysis Tab reports. The value for Cognos reports must be `/content/folder[@name='Affinium Plan - Object Specific Reports']`. The value for Unica Insights reports must be Plan/Affinium Plan - Object Specific Reports

Default value

```
/content/folder[@name='Affinium Plan - Object Specific Reports']
```

cacheListOfReports

Description

This parameter enables caching of a list of reports on object instance's analysis page.

Valid values

```
True | False
```

Default value

```
False
```

Unica Plan | umoConfiguration | invoiceRollup

The properties in this category specify options for invoice rollups.

invoiceRollupMode

Description

Specifies how rollups occur. Acceptable values follow.

Valid values

- `immediate`: rollups occur every time that an invoice is marked PAID.
- `schedule`: rollups occur on a scheduled basis.

  If this parameter is set to `schedule`, the system uses the following parameters to determine when rollups occur.

  - `invoiceRollupScheduledStartTime`
  - `invoiceRollupScheduledPollPeriod`

Default value

```
immediate
```

invoiceRollupScheduledStartTime

Description

If **invoiceRollupMode** is `schedule`, this parameter is used as follows.

- If this parameter contains a value (for example, 11:00 pm), that value is the start time for the schedule to start.
- If this parameter is undefined, the rollup schedule starts when the server starts.

If **invoiceRollupMode** is `immediate`, this parameter is not used.

Default value

`11:00 pm`

invoiceRollupScheduledPollPeriod

Description

If **invoiceRollupMode** is `schedule`, this parameter specifies the poll period in seconds for rollup to occur.

If **invoiceRollupMode** is `immediate`, this parameter is not used.

Default value

`3600` (1 hour)

Unica Plan | umoConfiguration | database

These properties specify information about the database that is used for Unica Plan.

fileName

Description

Path to file for loading data sources using JNDI lookup.

Default value

`plan_datasources.xml`

sqlServerSchemaName

Description

Specifies the database schema to use. This parameter applies only if you are using SQL Server for your Unica Plan database.

Default value

`dbo`

db2ServerSchemaName

Important: Changes to the default value supplied for this parameter are not recommended.

Description

Used internally by Unica applications.

Default value

blank

thresholdForUseOfSubSelects

Description

Specifies the number of records beyond which a subquery must be used in the IN clause of SQL (for listing pages) instead of the actual entity IDs in the IN clause. Setting this parameter improves performance for Unica Plan installations that have a large application data set. As a best practice, do not change this value unless you encounter performance issues. If this parameter is missing or commented out, the database behaves as if the threshold is set to a large value.

Default value

`3000` (records)

commonDataAccessLayerFetchSize

Description

This parameter specifies resultset fetch size for certain performance sensitive, critical queries.

Default value

`0`

commonDataAccessLayerMaxResultSetSize

Description

This parameter specifies maximum resultset size for certain performance sensitive, critical queries.

Default value

`-1`

useDBSortForAllList

Description

This parameter is used to configure ALL Unica Plan List Handlers. Use another **useDBSortFor<module>List** parameter to override the paging behavior of a particular list.

Valid values

- `True`: get one page of list data from the database at a time.
- `False`: cache all list data.

Default value

`True`

useDBSortForPlanList

Description

This parameter is used to configure the Plan List Handler.

Valid values

- `True`: get one page of list data from the database at a time.
- `False`: cache all list data.

Default value

`True`

useDBSortForProjectList

Description

This parameter is used to configure the Project List Handler.

Valid values

- `True`: get one page of list data from the database at a time.
- `False`: cache all list data.

Default value

`True`

useDBSortForTaskList

Description

This parameter is used to configure the Task List Handler.

Valid values

- `True`: get one page of list data from the database at a time.
- `False`: cache all list data.

Default value

`True`

useDBSortForProgramList

Description

This parameter is used to configure the Program List Handler.

Valid values

- `True`: get one page of list data from the database at a time.
- `False`: cache all list data.

Default value

`True`

useDBSortForApprovalList

Description

This parameter is used to configure the Approval List Handler.

Valid values

- `True`: get one page of list data from the database at a time.
- `False`: cache all list data.

Default value

`True`

useDBSortForInvoiceList

Description

This parameter is used to configure the Invoice List Handler.

Valid values

- `True`: get one page of list data from the database at a time.
- `False`: cache all list data.

Default value

`True`

useDBSortForAlerts

Description

This parameter is used to configure the Alerts List Handler.

Valid values

- `True`: get one page of list data from the database at a time.
- `False`: cache all list data.

Default value

`True`

Unica Plan | umoConfiguration | listingPages

These properties specify information about listing items, such as marketing objects or projects, on pages in Unica Plan.

listItemsPerPage

Description

Specifies how many items (rows) are displayed in one list page. This value must be greater than 0.

Default value

```
10
```

listPageGroupSize

Description

Specifies the size of visible page numbers in the list navigator in the list page. For example, pages 1-5 is a page group. This value must be greater than 0.

Default value

```
5
```

maximumItemsToBeDisplayedInCalendar

Description

The maximum number of objects (plans, programs, projects, or tasks) the system displays on calendars. Use this parameter to limit the number of objects that display when users select the calendar view. The number 0 indicates that there is no restriction.

Default value

```
0
```

listDisplayShowAll

Description

Display "Show All" link on listing pages.

Default value

```
False
```

Valid Values

```
True | False
```

Auto wrap

Description

Enables the display of the long names without truncation, instead of displaying the ellipsis at the end. It is applicable for Project name, Task name, Approval name, Offer name, and Offer list name. The long name is wrapped and shown in the next line of the same row.

By default **Auto wrap** is set to `False`. After setting the value of **Auto wrap** to `True`, you must restart Platform and Plan application, to activate this configuration.

Default value

`False`

Valid Values

`True | False`

Unica Plan | umoConfiguration | objectCodeLocking

These properties specify information about object locks for plans, programs, projects, assets, and marketing objects in Unica Plan.

enablePersistentObjectLock

Description

This parameter must be set to `True` if Unica Plan is deployed in a clustered environment. The object lock information is persistent in the database.

Valid values

`True | False`

Default value

`False`

lockProjectCode

Description

Determines whether users can edit the Project Code or PID on the Summary tab of a project.

Valid values

- `True`: enables locking.
- `False`: disables locking.

Default value

`True`

lockProgramCode

Description

Determines whether users can edit the Program Code or PID on the Summary tab of a program.

Valid values

- `True`: enables locking.
- `False`: disables locking.

Default value

`True`

lockPlanCode

Description

Determines whether users can edit the Plan Code or PID on the Plan Summary tab for a plan.

Valid values

- `True`: enables locking.
- `False`: disables locking.

Default value

`True`

lockMarketingObjectCode

Description

Determines whether users can edit the Marketing Object Code or PID on the Summary tab of a marketing object.

Valid values

- `True`: enables locking.
- `False`: disables locking.

Default value

`True`

lockAssetCode

Description

Determines whether users can edit the Asset Code or PID on the Summary tab of an asset.

Valid values

- `True`: enables locking.
- `False`: disables locking.

Default value

`True`

Unica Plan | umoConfiguration | thumbnailGeneration

These properties specify information about how and when Unica Plan generates thumbnails.

trueTypeFontDir

Description

Specifies the directory where the True Type fonts are located. This parameter is required for thumbnail generation on operating systems other than Windows™ that use Aspose. For Windows™ installations, this parameter is optional.

Default value

blank

coreThreadPoolSize

Description

Specifies the number of persistent threads that are kept in the thread pool for thumbnail generator threads.

Default value

`5`

maxThreadPoolSize

Description

Specifies the maximum number of threads that are allowed in the thread pool for thumbnail generator threads.

Default value

`10`

threadKeepAliveTime

Description

Parameter to configure the keep-alive time for thumbnail generator threads.

Default value

`60`

threadQueueSize

Description

Parameter to configure the thread queue size for thumbnail generator threads.

Default value

`20`

disableThumbnailGeneration

Description

Determines whether thumbnail images are generated for uploaded documents. A value of `True` enables thumbnail generation.

Default value

`False`

Valid values

`True | False`

markupImgQuality

Description

Magnification or zoom factor to apply to the rendered page.

Default value

`1`

Unica Plan | umoConfiguration | Scheduler | intraDay

This property specifies how frequently the scheduler runs during the day.

schedulerPollPeriod

Description

Defines how frequently, in seconds, a batch job to calculate project health status runs each day.

Note: Only the daily batch job updates the project health status history, which is used by reports.

Default value

`60` (seconds)

Unica Plan | umoConfiguration | Scheduler | daily

This property specifies what time the scheduler starts each day.

schedulerStartTime

Description

Defines the start time for a batch job that calculates project health status. This job also:

- Updates the project health status history that is used by reports.
- Initiates distribution of email notifications to users who subscribe to them.

Note: The system initiates this batch job only if the calculation is not already running. Define this parameter so that the job starts at a different time than the **intraDay** parameter, and at a time when users are not likely to request this calculation manually.

Default value

`11:00 pm`

Unica Plan | umoConfiguration | Notifications

These properties specify information about notifications in Unica Plan, including information about the event monitor.

notifyPlanBaseURL

Description

The URL for your Unica Plan deployment, including the host name and port number. Unica Plan includes this URL in notifications that contain links to other information in Unica Plan.

Note: Do not use "localhost" as a server name unless your mail client and Unica Plan server are running on same server.

Default value

`http://<server>:<port>/plan/affiniumplan.jsp`

notifyDelegateClassName

Description

The fully qualified Java™ class name of the delegate implementation to be instantiated by the service. This class must implement the `com.unicacorp.afc.service.IServiceImpl` interface. Defaults to a local implementation if not specified.

Default value

blank

notifyIsDelegateComplete

Description

Optional Boolean string that indicates whether the delegate implementation is complete. Defaults to `True` if not specified.

Default value

`True`

Valid Values

`True | False`

notifyEventMonitorStartTime

Description

Specifies when the event notification monitor process begins for the first time after Unica Plan starts. Format the value according to the short version of the `java.text.DateFormat` class for the current locale. For example, in US English locale, a valid string might be `11:45 pm`.

Default value

Blank (Immediately after Unica Plan is started.)

notifyEventMonitorPollPeriod

Description

Defines the approximate time, in seconds, for the event monitor to sleep between polls. Events accumulate in the event queue between polling periods; shorter polling periods process notifications sooner, but impose more system overhead. If you erase the default and leave the value blank, the poll period defaults to a short period, usually under a minute.

Default value

`5` (seconds)

notifyEventMonitorRemoveSize

Description

Specifies the number of events to remove from the queue in one shot. The event monitor removes events from the event queue in the increments that are specified by this value until none are left.

Note: You can set this value to a number other than 1 to increase event processing performance. However, if the service host goes down before removed events are processed, there is a risk of event loss.

Default value

`10`

alertCountRefreshPeriodInSeconds

Description

Specifies, in seconds, the system-wide alert count refresh period for the alerts count. This count displays near the top of the navigation bar after a user logs in.

Note: Changing the refresh period to poll faster can have performance implications in a multi-user environment.

Default value

`180` (3 minutes)

Unica Plan | umoConfiguration | Notifications | Email

These properties specify information about email notifications in Unica Plan.

notifyEMailMonitorStartTime

Description

Specifies when the email monitor process runs for the first time after Unica Plan starts. Format the value according to the short version of the `java.text.DateFormat` class for the current locale. For example, in US English locale, a valid string might be `11:59 pm`.

Default value

Blank (Immediately after Unica Plan starts.)

notifyEMailMonitorPollPeriod

Description

Defines the approximate time, in seconds, for the email monitor to sleep between polls.

Note: As with events, email messages accumulate in the queue between polling periods; shorter polling times send email messages sooner, but can increase system overhead.

Default value

`60` (seconds)

notifyEMailMonitorJavaMailSession

Description

JNDI name of an existing, initialized JavaMail™ Session to use for email notifications. If not specified and the delegate is marked `Complete`, then the JavaMail™ host parameter must be supplied so Unica Plan can create a session.

Default value

blank

notifyEMailMonitorJavaMailProtocol

Description

Specifies the mail server transport protocol to use for email notifications.

Default value

`smtp`

notifyEMailMonitorRemoveSize

Description

Specifies the number of email messages to remove from queue at one time. The email monitor continues to remove messages from the email queue incrementally until none remain.

Note: You can set this value to a number other than 1 to increase email processing performance. However, if the service host goes down before removed email messages are processed, there is a risk of message loss.

Default value

`10` (messages)

notifyEMailMonitorMaximumResends

Description

Specifies the maximum number of times the system attempts to send an email message that failed in the first attempt to send it. When a send fails, the email is put back into the queue until it reaches the maximum number of attempts that are allowed by this parameter.

For example, **notifyEMailMonitorPollPeriod** is set to poll every 60 seconds. Setting the **notifyEMailMonitorMaximumResends** property to 60 attempts causes the email monitor to resend a failed message one time in each poll (every minute), for up to 1 hour. A value of 1440 (24x60) causes the email monitor to try every minute for up to 24 hours.

Default value

`1` (attempt)

showUserNameInEmailNotificationTitle

Description

Specifies whether the Unica Plan notification and alert systems includes the user name in the **From** field of email notifications.

Note: This setting is applicable only to email messages sent by the notification and alert systems in Unica Plan.

Valid values

- `True`: Unica Plan appends the user name to the title of the message and displays both in the **From** field of the email
- `False`: Unica Plan displays only the message title in the **From** field

Default value

`False`

notifyEMailMonitorJavaMailDebug

Description

Specifies whether JavaMail™ debug mode is set.

Valid values

- `True`: enables JavaMail™ debug.
- `False`: disables debug tracing.

Default value

`False`

Unica Plan | umoConfiguration | Notifications | project

These properties specify information about project alarms in Unica Plan.

notifyProjectAlarmMonitorStartTime

Description

Specifies when the project alarm monitors run for the first time after Unica Plan starts. Format the value according to the short version of the `java.text.DateFormat` class for the current locale. For example, in US English locale, a valid string might be `11:59 pm`. If you erase the default and leave the value blank, this monitor starts immediately after you create it.

Default value

`10:00 pm`

notifyProjectAlarmMonitorPollPeriod

Description

Defines the approximate time, in seconds, for the project and program alarm monitors to sleep between polls.

Default value

Blank (60 seconds)

notifyProjectAlarmMonitorScheduledStartCondition

Description

Defines the number of days before the start date of a project for Unica Plan to send notifications to users.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`1` (day)

notifyProjectAlarmMonitorScheduledEndCondition

Description

Defines the number of days before the end date of a project for Unica Plan to send end notifications to users.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`3` (days)

notifyProjectAlarmMonitorTaskScheduledStartCondition

Description

Defines the number of days before the start date of a task for Unica Plan to send start notifications to users.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`1` (day)

notifyProjectAlarmMonitorTaskScheduledEndCondition

Description

Defines the number of days before the end date of a task for Unica Plan to send end notifications to users.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`3` (days)

notifyProjectAlarmMonitorTaskLateCondition

Description

Defines the number of days after the start date of a task for Unica Plan to send users notification that a task did not start.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`3` (days)

notifyProjectAlarmMonitorTaskOverdueCondition

Description

Defines the number of days after the end date of a task for Unica Plan to send users notification that a task did not finish.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`3` (days)

notifyProjectAlarmMonitorTaskScheduledMilestoneCondition

Description

Defines the number of days before the start date of a milestone task for Unica Plan to send notifications.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`1` (day)

Unica Plan | umoConfiguration | Notifications | projectRequest

These properties specify information about project request alarms in Unica Plan.

notifyRequestAlarmMonitorLateCondition

Description

Defines the number of days for Unica Plan to send a notification that the request is late.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`3` (days)

notifyRequestAlarmMonitorScheduledEndCondition

Description

Defines the number of days before the end date of a request for Unica Plan to send end notifications to users.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`1` (day)

Unica Plan | umoConfiguration | Notifications | program

The properties in this category specify options for program notification schedules.

notifyProgramAlarmMonitorScheduledStartCondition

Description

Defines the number of days before the start date of a program that Unica Plan sends start notifications to users.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`1` (day)

notifyProgramAlarmMonitorScheduledEndCondition

Description

Defines the number of days before the end date of a program that Unica Plan sends end notifications to users.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`3` (days)

Unica Plan | umoConfiguration | Notifications | marketingObject

These properties specify information about marketing object alarms in Unica Plan.

notifyComponentAlarmMonitorScheduledStartCondition

Description

Specifies the number of days before the start date of a marketing object for Unica Plan to send start notifications to users.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`1` (day)

notifyComponentAlarmMonitorScheduledEndCondition

Description

Specifies the number of days before the end date of a marketing object for Unica Plan to send end notifications to users.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`3` (days)

Unica Plan | umoConfiguration | Notifications | approval

These properties specify information about approval alarms in Unica Plan.

notifyApprovalAlarmMonitorStartTime

Description

Specifies when the approval alarm monitor begins processing for the first time after Unica Plan starts. Format the value according to the short version of the `java.text.DateFormat` class for the current locale. For example, in US English locale, a valid string might be `11:59 pm`. If you delete the default and leave this value blank, the monitor starts immediately after it is created.

Note: For best results, configure the alarm monitors to start during off-peak hours and stagger their start times to spread out the data processing load.

Default value

`9:00 pm`

notifyApprovalAlarmMonitorPollPeriod

Description

Specifies the approximate time, in seconds, for the approval alarm monitor to sleep between polls.

Default value

Blank (60 seconds)

notifyApprovalAlarmMonitorLateCondition

Description

Specifies the number of days after the start date of an approval for the system to begin notifying users that the approval is late.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`3` (days)

notifyApprovalAlarmMonitorScheduledEndCondition

Description

Specifies the number of days before the end date of an approval for the system to begin sending end notifications to users.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`1` (day)

Unica Plan | umoConfiguration | Notifications | asset

These properties specify information about asset alarms in Unica Plan.

notifyAssetAlarmMonitorStartTime

Description

Specifies when the asset alarm monitor process runs for the first time after Unica Plan starts. Format the value according to the short version of the `java.text.DateFormat` class for the current locale. For example, in US English locale, a valid string might be `11:59 pm`. If you delete the default and leave this value blank, the monitor starts immediately after it is created.

Note: For best results, configure the alarm monitors to start during off-peak hours and stagger their start times to spread out the data processing load.

Default value

`11:00 pm`

notifyAssetAlarmMonitorPollPeriod

Description

Specifies the time, in seconds, for the asset alarm monitor to sleep between polls.

Default value

Blank (60 seconds)

notifyAssetAlarmMonitorExpirationCondition

Description

Specifies the number of days before an asset is going to expire for Unica Plan to notify users that the asset is about to expire.

Note: If this value is `-1`, Unica Plan does not check for expiration.

Default value

`5` (days)

Unica Plan | umoConfiguration | Notifications | invoice

These properties specify information about invoice alarms in Unica Plan.

notifyInvoiceAlarmMonitorStartTime

Description

Specifies when the invoice alarm monitor process runs for the first time after Unica Plan starts. Format the value according to the short version of the `java.text.DateFormat` class for the current locale. For example, in US English locale, a valid string might be `11:59 pm`. If you delete the default and leave the value blank, the monitor starts immediately after you create it.

Note: For best results, configure the alarm monitors to start during off-peak hours and to stagger their start times to spread out the data processing load.

Default value

`9:00 pm`

notifyInvoiceAlarmMonitorDueCondition

Description

Specifies the number of days before the due date for Unica Plan to notify users that an invoice is due.

Note: If this value is `-1`, then Unica Plan does not send these notifications.

Default value

`5` (days)

Unica Plan | umoConfiguration | Notifications | Out of Office

notifyOutOfOfficeMonitorStartTime

Description

The time to start monitoring the Out of Office status of the Plan users.

Default value

-

notifyOutOfOfficeMonitorPollPeriod

Description

The time interval, in seconds, to recheck the out of office status of Plan users. For example, if you set the value as `600`, the system checks every `10` minutes to confirm Out of Office status of Plan users.

Default value

`1800`

Unica Plan | umoConfiguration | WorkflowService

This feature is not supported.