Unica Plan V12.1.4 Upgrade Guide

Contents

Chapter 1. Upgrade overview

An upgrade of Unica Plan is complete when you upgrade, configure, and deploy Unica Plan. The Unica Plan Upgrade Guide provides detailed information about upgrading, configuring, and deploying Unica Plan.

Use the Upgrade roadmap section to obtain a broad understanding about using the Unica Plan Upgrade Guide.

Upgrade roadmap

Use the upgrade roadmap to quickly find the information that you need for upgrading Unica Plan.

You can use the following Table 1: Unica Plan upgrade roadmap *(on page 1)* table to scan the tasks that must be completed for upgrading Unica Plan:

Table 1. Unica Plan upgrade roadmap

This two-columned table describes the topics that are included in the Unica Plan Upgrade Guide in one column, and the explanation of the topics in the second column.

Topic	Information
Upgrade overview *(on page 1)*	This topic provides the following information: • How the installers work *(on page 3)* • Modes of installation *(on page 4)* • Unica Plan documentation and help *(on page 7)*
Planning the Unica Plan upgrade *(on page 9)*	This topic provides the following information: • Prerequisites *(on page 10)* • Upgrade prerequisites for all Unica products *(on page 13)*

Table 1. Unica Plan upgrade roadmap

This two-columned table describes the topics that are included in the Unica Plan Upgrade Guide in one column, and the explanation of the topics in the second column.

(continued)

Topic	Information
	• Error logs and warning messages *(on page 15)* • Unica Plan data source information worksheet *(on page 16)*
To upgrade Unica Plan *(on page 17)*	This topic provides the following information: • Backing up the system before you upgrade *(on page 17)* • Running the installer and updating the configuration properties *(on page 17)* • Upgrade your database manually *(on page 19)* • Deploying the upgraded web application and running the upgrade processes *(on page 22)* • Verifying the Unica Plan upgrade *(on page 23)* • Restoring trigger procedures *(on page 24)* • Upgrade with existing campaign projects or requests *(on page 16)* • Upgrading Unica Plan in a clustered environment *(on page 25)*
Overview *(on page 26)*	This topic provides the following information: • Deploying Unica Plan on WebSphere *(on page 36)* • Deploying Unica Plan on WebLogic *(on page 44)* • Deploying Unica Plan on JBoss *(on page 48)* • Deploying Unica Plan on Apache Tomcat *(on page 51)*
Uninstalling Unica Plan *(on page 58)*	This topic provides the information about how to uninstall Unica Plan.

Table 1. Unica Plan upgrade roadmap

This two-columned table describes the topics that are included in the Unica Plan Upgrade Guide in one column, and the explanation of the topics in the second column.

(continued)

Topic	Information
The config tool utility *(on page 60)*	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.

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 2. 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_ump`*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_ul`*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_urpc`*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.

Correcting the registry file when upgrade installation fails

If installation fails because the installer could not detect the base version of your installed product, you can correct the registry file as described here.

The InstallAnywhere Global registry file, named `.com.zerog.registry.xml`, is created when a Unica product is installed. The registry file tracks all installed Unica products, including their features and components, on that server.

1. Locate the `.com.zerog.registry.xml` file.

   Depending on the server on which you are installing, the `.com.zerog.registry.xml` file is in one of the following locations.

   - On Windows servers, the file is in the `Program Files/Zero G Registry` folder.

     `Zero G Registry` is a hidden directory. You must enable the setting to view hidden files and folders.
   - On UNIX systems, the file is in one of the following directories.
     - Root user - `/var/`
     - Non-root user - `$HOME/`
2. Make a backup copy of the file.
3. Edit the file to change all entries that refer to the version of your installed product.

   For example, this is a section of the file that corresponds to Unica Plan version 8.6.0.3.

   ```
   <product name="Plan" id="dd6f88e0-1ef1-11b2-accf-c518be47c366"
   version=" 8.6.0.3 " copyright="2013" info_url="" support_url=""
   location="<HCL_Unica_Home>\Plan" last_modified="2013-07-25 15:34:01">
   ```

   In this case, you would change all entries that refer to `version=" 8.6.0.3 "` to the base version, which is 8.6.0.0 in this case.

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 3. Get up and running

This two-columned table describes the Unica Plan installation or upgrade tasks in one column, and their document in second column.

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 4. Configure and use Unica Plan

This two-columned table describes the Unica Plan administrative tasks in one column and task related document in second column.

Task	Documentation
• Set up and configure the system for users • Adjust security settings	*Unica Plan Administrator's Guide*

Table 4. Configure and use Unica Plan

This two-columned table describes the Unica Plan administrative tasks in one column and task related document in second column.

(continued)

Task	Documentation
• Map tables, define offer templates, and custom attributes • Run utilities and perform maintenance	
• 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 5. Getting help

This two-columned table contains information about how to start Online Help and access PDF documentation for the Unica Plan. Second column contains instructions for the tasks.

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 upgrade

For upgrading to Unica Plan 12.1.4 version, you must check from which version you are upgrading. Upgrade scenarios are based on the current version of the Unica Plan.

Unica Plan supports the following upgrade paths:

- 12.1.x → 12.1.4
- 12.1.0.x → 12.1.4

Customers on versions earlier than 8.6.x must:

- perform a Fast Upgrade from existing version to version 8.6.0 (for more information, see *HCL Unica 8.6.0 Fast Upgrade Guide*).
- perform a Fast Upgrade from version 8.6.0 to version 12.1.0 (for more information, see *HCL Unica 12.1.0 Fast Upgrade Guide*).
- perform an in-place upgrade from version 12.1.0 to version 12.1.4.

Customers on versions earlier than 11.1.x.x can:

- perform a Fast Upgrade from existing version to version 12.1.0 (for more information, see *HCL Unica 12.1.0 Fast Upgrade Guide*).
- perform an in-place upgrade from version 12.1.0 to version 12.1.4.

Customers on versions 11.1.x.x/12.0.x.x can use one of the following options for upgrade:

- **Option 1**
  - perform an in-place upgrade from existing version to version 12.1.0.
  - perform an in-place upgrade from version 12.1.0 to version 12.1.4.
- **Option 2**
  - perform a Fast Upgrade from existing version to version 12.1.0 (for more information, see *HCL Unica 12.1.0 Fast Upgrade Guide*).
  - perform an in-place upgrade from version 12.1.0 to verison 12.1.4.

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.

Upgrade prerequisites for all Unica products

Meet all requirements for permissions, operating system, and knowledge correctly before you upgrade Plan to ensure a seamless upgrade experience.

Removing response files generated by previous installations

If you are upgrading from a version before 8.6.0, you must delete the response files that are generated by previous Unica Plan installations. Old response files are not compatible with the 8.6.0 and later installers.

Failure to remove old response files can result in having incorrect data pre-filled in installer fields when the installer is run, or in the installer failing to install some files or skipping configuration steps.

The response file is named `installer.properties`.

The response files for each product are named `installer_productversion.properties`.

The installer creates response files in the directory that you specify during installation. The default location is the home directory of the user.

User account requirement for UNIX™

On UNIX™, the user account that installed the product must complete the upgrade, otherwise the installer fails to detect a previous installation.

32-bit to 64-bit version upgrades

If you are moving from a 32-bit to a 64-bit version of Unica Plan, ensure that you complete the following tasks:

- Ensure that the database client libraries for your product data sources are 64-bit.
- Ensure that all relevant library paths, for example startup or environment scripts, correctly reference the 64-bit versions of your database drivers.

Unloading unused files from memory on AIX®

For installations on AIX®, run the slibclean command that is included with your AIX® installation to unload unused libraries from the memory before you run the installer in the upgrade mode.

> **Note:** You must run the slibclean command as a root user.

Backing up custom files

Before starting the upgrade to Unica 12.1.4, backup all the custom files that exist in the `<UNICA_HOME>/jre/` location. If you do not backup the custom files that exist in the `<UNICA_HOME>/jre/` location, you will lose the files because the Unica 12.1.4 upgrade deletes the existing `jre` folder and installs a new `jre` folder containing Oracle JRE files.

> **Note:** If the operating system is IBM AIX, Unica 12.1.4 installs IBM JRE.

Running database query for cleanup

Before you upgrade Unica Plan, run the database query to remove duplicate project request ids, if any.

Run the query in your database to find and remove all duplicate results returned by this query to ensure the successful upgrade of Unica Plan.

Complete the following steps to run the database query:

1. Open your database console that has Unica Plan system tables.
2. Enter the following query:

```
SELECT proj_request_id, count(proj_request_id) num

FROM uap_projects

WHERE proj_request_id in (SELECT project_id FROM uap_projects WHERE
state_code = 'ACCEPTED')

group by proj_request_id

having count(proj_request_id) > 1
```

3. The query returns project request ids that are duplicate. Analyze the results to determine which of the duplicate rows you use and which can be deleted. You can look in the `uap_projects_last_mod_date` table or view the data in the reference tables to determine which records to delete. To delete the rows, run the delete query on the database. If the duplicate rows are not deleted, the upgrade may fail.

Error logs and warning messages

When you upgrade, the system logs the messages that are produced during the process. View the log files to see any informational or error messages that occurred during upgrade.

For reference, you can find the log files with these messages in the following files and database tables.

- *<HCL_Unica_Home>*/HCL_Unica_Installer_Install<date_time>.log
- *<Plan_Home>*/Plan_Install_<date_time>.log
- *<Platform_Home>*/Platform_Install<date_time>.log
- *<USER_HOME>*/HCL_Unica_Installer_stdout.log
- *<USER_HOME>*/HCL_Unica_Installer_stderr.log
- *<USER_HOME>*/Platform_stdout.log
- *<USER_HOME>*/Platform_stderr.log
- *<USER_HOME>*/Plan_stdout.log
- *<USER_HOME>*/Plan_stderr.log

Upgrade with existing campaign projects or requests

If you upgrade Unica Plan system that is integrated with Unica Campaign and you have existing campaign projects without corresponding linked campaigns, create the linked campaigns before you upgrade to Unica Plan. Likewise, if you have existing project requests for campaign projects, accept or reject the requests before you upgrade to Unica Plan.

If you do not link them before you upgrade, then when you attempt to create the campaigns for these projects or accept the requests after the system is upgraded, the campaigns are not linked properly to the Unica Plan projects.

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 6. Data source information worksheet

Item	Value
Data source type	SQL Server / IBM DB2 / Oracle / MariaDB / OneDB.
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.

Chapter 3. To upgrade Unica Plan

To upgrade Unica Plan, you back up your existing installation, ensure that Unica Platform is upgraded and running, run the installer, restore any trigger procedures, deploy the upgraded applications, and then run some post-deployment processes.

Note that older versions of Unica Plan were named Affinium Plan or Marketing Operations. All versions are now referred to as Unica Plan in this guide.

Backing up the system before you upgrade

Take a back up of your system before you start the upgrade process. If the upgrade fails, you can restore the last version of Unica Plan.

Complete the following steps to back up the system:

1. Undeploy your existing version of Unica Plan.
2. Back up all files and directories in your existing installation folder.

   > **Note:** If you have modified the sample trigger procedures or the `procedure_plugins.xml` file, you must restore files from the backup after upgrading to avoid losing trigger procedures. The files you must restore are in the `/devkits/integration/examples/src/procedure` folder.

3. Back up the database that holds your Unica Plan system tables.

Running the installer and updating the configuration properties

Before you run the installer, ensure that you have the appropriate database connection information for both the Unica Platform database and the Unica Plan database.

Complete the following steps to run the installer and update configuration properties:

1. Run the installer and specify your existing installation directory as the installation directory to use. For more information, see How the installers work *(on page 3)*.

   The installer detects that you have an earlier version installed and runs in upgrade mode.

2. Follow the instructions in the installation wizard.

   📝 **Note:** Note that the installer can upgrade the database automatically. If your company policy does not permit you to use this feature, select the **Manual database setup** option and then run the scripts manually after installing the software and before deploying the web application.

3. When the installer finishes, verify that the conf/plan_ehcache.xml file contains the attribute name="`PlanEhCacheManager`". If not, edit the file and add it. Example:

   `<ehcache updateCheck="false" name="PlanEhCacheManager">`

4. Log in to the upgraded Unica Platform application and select **Settings > Configuration**. Review the properties in the Unica Plan categories and set or change any.

Configurations to be performed if upgrading from a version earlier than 12.0.0

If you are upgrading from Unica Plan version earlier than 12.0.0, after the upgrade, there may be two Plan menus in Unica Platform: one menu with all the features of Plan and one with just the Checklist feature.

To avoid the issue of two Plan menus, complete the following steps:

1. If you have not completed the upgrade, complete the following steps:

   a. Run the upgrade installer.

   b. After upgrade and before starting the web application server, complete the following steps for each locale:

      i. Check the file `sysmenu.xml` in the `conf/locale` location.

     ii. If you find `<menugroup id="projectmanagement">`, replace it with `<menugroup id="Operations">`.

2. If you have completed the upgrade and the two Plan menus already exist, complete the following steps to fix the issue:

   a. Stop the Plan web application.

   b. Access the file `sysmenu.xml` from the `conf/locale` location.

   c. In the file, if you find `conf/locale/<menugroup id="projectmanagement">`, remove the `<menugroup id="Operations">` section within it, and rename `<menugroup id="projectmanagement">` as `<menugroup id="Operations">`.

   d. Perform *Step c* for all locales.

   e. Restart the Plan web application.

   f. From the Unica Platform homepage, navigate to **Settings > Plan Settings** The **Administrative settings** page appears.

   g. Inside **System administration** settings and within **Restrcited options**, select **Synchronize menus**.

   h. Restart both Unica Platform and Unica Plan.

Upgrade your database manually

The installer can upgrade the Unica Plan database during the upgrade process. Use the database setup utility, `umodbsetup`, to upgrade the tables manually if your company policy doesn't allow you to upgrade the database.

The `umodbsetup` utility does one of the following actions:

- Option 1: Upgrades the system tables in the Unica Plan database and populates the tables with required default data.
- Option 2: Outputs the database upgrade and population scripts to a file that you or your database administrator can then run in your own database client.

Configuring environment variables

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

1. From the `<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 the environment variable, see the *Plan Installation Guide*.
3. Save and close the file.
4. From the `<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.

Database setup utility

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

For example, the following command runs an upgrade, sets the locale to en_US, and sets the logging level to medium:

```
./umodbsetup.sh -t upgrade -L en_US -l medium
```

The following is the description of all the possible variables for the utility:

Table 7. Variables for the `umodbsetup.sh` utility

This two-columned table describes variables in one column, and description in the second column.

Variable	Description
-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 may have been updated even though the upgrade failed. When you have corrected 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 12.1.0.0`
-f	For upgrades only. Instructs the utility to use the base version specified by the `-b` variable, overriding the base version it may detect in the database. See the description of the `-b` variable.
-h	Provides help for the utility.
-l	Records the output from the actions that the `umodbsetup` utility performs in the `umo-tools.log` file. This file is located 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.

Table 7. Variables for the `umodbsetup.sh` utility

This two-columned table describes variables in one column, and description in the second column.

(continued)

Variable	Description
-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 need to 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`
-v	Verbose.

Running database scripts manually, if necessary

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 upgrade and populate the system tables.

Deploying the upgraded web application and running the upgrade processes

You must deploy the upgraded web applications to the web application server. After you deploy the web applications, you can start the upgrade process.

> **Note:** If your Unica Plan is integrated with Campaign, verify that Campaign is upgraded and running before you proceed.

1. Deploy Unica Plan in your web application server as described in Overview *(on page 26).*
2. Restart the application server.
3. When the application is running, log in and verify that the upgrade worked properly. Select **Settings > Configuration** and verify that **Unica Plan** is in the list on the left. Then expand the **Unica Plan** section and verify that the **umoConfiguration** category is in the list.
4. Select **Settings > Plan Settings**.
5. Scroll down and click **Unica Plan Upgrade**. A list of upgrade processes displays. These processes change the configuration of the application by upgrading database tables and the files that store site-specific customizations.

   For information about an upgrade process, click **Help** next to that process.
6. To run the selected processes, click **Upgrade**.

Verifying the Unica Plan upgrade

You must upgrade and deploy the Unica Platform before you upgrade Unica Plan.

Complete the following steps to verify the upgrade:

1. Review the log files in the application server logs directory for error messages. The message "UAPContext Init failed" indicates that the upgrade did not complete successfully.
2. Use Internet Explorer or another supported browser to access the Unica URL.
3. Navigate to different Unica Plan features, including the asset file. In case you see two "Operations" or "Plan settings" menus, go to **Settings > Plan settings**, click **Synchronize Menus**, logout from the application and login again.
4. Create instances of different Unica Plan objects, including plans, programs, projects, and your own custom marketing object types.
5. **Note:** This step is applicable only if you plan to migrate offer management from Unica Plan to Unica Centralized Offer Management using the **Migration to Centralized Offer Management** utility.

If you are using reverse proxy or web access management software (ISAM, Siteminder, etc), populate `serverURLInternal` with the direct internal URL of Unica Centralized Offer Management.

`serverURLInternal=http://<hostname>:<port>/Offer`

6. Select **Settings > Plan Settings > Template Configuration** and then click **Validate Templates**.

7. Click **Synchronize Menus**. Logout from the application, login again, and verify the **Plan Settings menu** item.

8. If your installation customizes Unica Plan with the application programming interface, verify that your customizations are not affected by compatibility issues.

9. If you use trigger procedures, restore them.

Restoring trigger procedures

You can restore trigger procedures after you upgrade the Unica Plan application.

Complete the following steps to restore the trigger procedures:

1. Restore the procedures and the `procedure_plugins.xml` file from the backup you made earlier. Put them in the default location for the files:

   `<Plan_Home>\devkits\integration\examples\src\procedure`

2. If necessary, rebuild the integration services procedures using the `build` file located in the `<Plan_Home>\devkits\integration\examples\build` directory under your Unica Plan installation.

3. Update the following parameters on the **Settings > Configuration > Plan > umoConfiguration > attachmentFolders** page. Set the values to point to the directories you created in the previous step.
   - Set **graphicalRefUploadDir** to `<Plan_Home>\graphicalrefimages`
   - Set **templateImageDir** to `<Plan_Home>\images`
   - Set **recentDataDir** to `<Plan_Home>\recentdata`
   - Set **workingAreaDir** to `<Plan_Home>\umotemp`

Upgrading Unica Plan in a clustered environment

Use the following guidelines when you upgrade multiple instances of Unica Plan in a clustered environment:

- Undeploy all of the instances of Unica Plan.
- Follow the directions in this chapter to upgrade.
- Use your web application server's automatic deployment features to deploy the EAR file in your cluster.

See the Unica Plan Installation Guide for considerations if you install Unica Plan in a clustered environment.

> **Note:** You must check or update all plan_ehcache.xml files.
>
> You must provide a unique name if not already present, like `<ehcache updateCheck="false" name="PlanEhCacheManager">`"

Chapter 4. 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 8. Unica Products and the JVM Parameters to be Configured

Unica Product Name	JVM Parameters
Unica Campaign	`-DUNICA_PLATFORM_CACHE_EN-ABLED=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.ss-l=false` `-Dcom.sun.management.jmxremote.au-thenticate=false`

Table 8. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters

	• `-Dcom.sun.management.jmxremote-`
	`.password.file=../conf/jmxre-`
	`mote.password`
	• `-Dcom.sun.management.jmxremote.ac-`
	`cess.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.keyStore-Type="JKS"
	
	• `-Djavax.net.ssl.trustStore="C:\se-`
	`curity\myTrustStore.jks"`
	• `-Djavax.net.ssl.trustStorePass-`
	`word=myPassword`
	
	Configuring the `ACOOptAdmin` tool (`ACOOp-`
	`tAdmin.sh` (UNIX) or `ACOOptAdmin.bat`
	(Windows) file)

Table 8. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters

	`-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	`-DUNICA_PLATFORM_CACHE_ENABLED=true` `-DUNICA_PLATFORM_LOCAL_CACHE_ENABLED=true`

Table 8. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters

	• `-Dclient.encoding.override=UTF-8`
	• `-DAUDIENCE_HOME=D:\sourcecode\lat-`
	  `est\AudienceCentral\Audience\om-`
	  `server`

Unica Contact Central	• `CONTACTCENTRAL_HOME` • `UNICA_PLATFORM_HOME`
Unica Content Integration	• `-DASSET_PICKER_HOME` • `-DUNICA_PLATFORM_HOME`
Unica Deliver	• `-Dcampaign.deliveretl.dis-` `abled=true`

Unica Interact	**Interact design time**
	
	The following are the JVM parameters for In-
	teract design time:
	
	• `Interact.UsernameToAlwaysDeployFor`
	• `com.unicacorp.interact.deliver-`
	  `.templateTimeout`
	• `DeploymentServletParameterEncoding`
	• `com.unicacorp.interact.flexoffers-`
	  `.defaultDateTimeFormat`
	• `com.unicacorp.interact.flexoffers-`
	  `.defaultDateFormat`
	• `com.unicacorp.Campaign.interact-`
	  `.offermapping.batchsize`
	• `com.unicacorp.Campaign.interact-`
	  `.offermapping.service.synctimeout`
	• `com.unicacorp.interact.cacheTTL`

Table 8. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	• `com.unicacorp.interact.cacheRe-freshIntervalInMin` • `com.unicacorp.interact.enableDT-PerfLogging` • `com.unicacorp.interact.compress-APIResponse` • `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.proxyProto-col` • `com.hcl.interact.http.proxyHost` • `com.hcl.interact.http.proxyPort` • `com.hcl.interact.http.proxyUser-name` • `com.hcl.interact.http.proxyPass-word`

Table 8. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	• `interact.jmx.monitoring.port` • `interact.offerserving.maxOfferAl-locationInMemoryPerInstance` • `interact.offerserving.maxDistribu-tionPerIntervalPerInstanceFactor` • `interact.ignitePort` • `com.unicacorp.interact.chDupe-CheckLimit` • `com.unicacorp.interact.rhDupe-CheckLimit` • `com.unicacorp.interact.chSuppress-Dupe` • `com.unicacorp.interact.rhSuppress-Dupe` • `com.unicacorp.interact.testclient-.nullValue` • `interact.ehcache.config` • `interact.api.dateFormat` • `com.hcl.interact.testrun.rowlimit` • `Interact.DisableExceptionStack-TracesInMacros` • `com.unicacorp.interact.enableDe-tailStats` • `com.unica.interact.deployment-.timeoutInSecs` • `com.ibm.interact.instance.name` • `com.unicacorp.interact.invalid-Paths` • `interact.XSessResponseConsumerMan-ager.generateOnlyOneResponse`

Table 8. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	• `tryToPreserveInexactFloatValues` • `com.unicacorp.interact.propertyRe-freshInterval` • `com.unicacorp.interact.scheduled-TasksProcessInterval` • `com.unicacorp.interact.eventpat-terns.parallelism` • `com.unicacorp.interact.eventpat-terns.restartRetries` • `com.unicacorp.interact.eventpat-terns.evaluateTimeoutMilli` • `com.unicacorp.interact.eventpat-terns.restartRetryDelayInSec` • `Interact.advisoryMessageEncoding-Overrides` • `com.unica.interact.api.insertSes-sionIDAsCookie` • `com.unica.interact.api.SessionID-CookieName` • `InteractMsgCode` • `com.ibm.interact.triggeredmessage-.enableJMSConsumer` • `com.unicacorp.interact.maxString-LengthInFormatMacro` • `ContinueEvaluatingBranchAndAdvOpt-TreatmentLogicDespiteExMessageList` • `DisableDecisionProcessBoxAndAdv-OptTreatmentLogging`

Table 8. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	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.MuffleErrorsForMutu-allyExclusiveBranchesWithNullVal-ues` Valid values are `True` and `False`. The default value is `False`. If you set the value to `False` and if the value of `-DDisable-DecisionProcessBoxAndAdvOptTreat-mentLogging` 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.minTreat-mentsPerThread` • `com.unicacorp.interact.maxTreat-mentPoolSize` • `CircuitBreaker.processTimeoutMil-lis`

Table 8. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters

	`com.unicacorp.interact.event.a-`
	`syncTimeoutMSec` `com.unicacorp.interact.eventAc-`
	`tionTimeout` `Interact.HTMI.Enabled` `Interact.HTMI.MaxRequestDuration-`
	`InMs` `Interact.HTMI.RecordIndividualAPIs` `Interact.HTMI.MaxStartSessionDura-`
	`tionInMs` `Interact.HTMI.MaxGetOffersDura-`
	`tionInMs` `Interact.HTMI.MaxPostEventDura-`
	`tionInMs` `Interact.HTMI.MaxGetProfileDura-`
	`tionInMs` `Interact.HTMI.LogErrorsEveryNth-`
	`Time` `Interact.HTMI.UseMillisecondTimers` `Interact.HTMI.Debug` `com.unicacorp.interact.suppress-`
	`WarningOnAnonymousUser` `com.hcl.interact.eventpatterns-`
	`.printPatternAction` `com.hcl.interact.eventpatterns.ea-`
	`gerPersist` `com.ibm.interact.triggeredmessage-`
	`.addPerfData` `com.unicacorp.interact.learning-`
	`.disableAggregator`

Table 8. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
	`com.unicacorp.interact.learning-.disableDeletion` `com.unicacorp.interact.learning-.ignoreInterval` `interact.services.loader.saveLoad-erFiles` `ConvertEveryNULLAttributeValueToA-JEPNullConstant` `includeJoinInfo` `com.unicacorp.interact.deployment-.reloadTimeout` `com.ibm.interact.lockTimeWarning-Threshold` `com.unicacorp.interact.cache.max-WaitTime` `DEFAULT_PERSISTENCE_PROVIDER` `com.unicacorp.interact.playback-.purgeBatchSize` ✎ **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	

Table 8. Unica Products and the JVM Parameters to be Configured (continued)

Unica Product Name	JVM Parameters
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_LOG-GING=TRUE` `-DENABLE_PERSISTENT_NAMEID_FOR-MAT=TRUE` `-DUNICA_PLATFORM_HOME=<platform_-home_directory_path>`
Unica Audience Central	`-DAUDIENCE_CENTRAL_HOME=<UNICA_-HOME>\<Audience_Home>`

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).

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.

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.

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

> 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.

   - 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.AsyncLogge`
        - `-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 , 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 12.1.4 installer creates a back up of the existing files:

File name before backup	File name after backup
`plan_log4j.xml`	`plan_log4j.xml_pre_12.1.4`
`plan_log4j_client.xml`	`plan_log4j_client.xml_pre_-12.1.4`

File name before backup	File name after backup
`plan_log4j_socket_server.xml`	`plan_log4j_socket_server.xml_pre_12.1.4`

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 12.1.4 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.

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.

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**.

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.

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.

- 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.AsyncLogge`
    - `-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 , 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 12.1.4 installer creates a back up of the existing files:

File name before backup	File name after backup
`plan_log4j.xml`	`plan_log4j.xml_pre_12.1.4`
`plan_log4j_client.xml`	`plan_log4j_client.xml_pre_-12.1.4`
`plan_log4j_socket_server.xml`	`plan_log4j_socket_server.xml_pre_12.1.4`

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 12.1.4 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.

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 + 12.1.4.

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.AsyncLoggerCont`
        - `-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 , 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 12.1.4 installer creates a back up of the
existing files:

File name before backup	File name after backup
`plan_log4j.xml`	`plan_log4j.xml_pre_12.1.4`
`plan_log4j_client.xml`	`plan_log4j_client.xml_pre_-12.1.4`
`plan_log4j_socket_server.xml`	`plan_log4j_socket_server.xml_pre_12.1.4`

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 12.1.4 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.AsyncLoggerCont`
    - `-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 , 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 12.1.4 installer creates a back up of the existing files:

File name before backup	File name after backup
plan_log4j.xml	plan_log4j.xml_pre_12.1.4
plan_log4j_client.xml	plan_log4j_client.xml_pre_-12.1.4

File name before backup	File name after backup
`plan_log4j_socket_server.xml`	`plan_log4j_socket_server.xml_pre_12.1.4`

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 12.1.4 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.

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.

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.

Synchronizing Labels in Plan Tasks

After upgrading to 12.1.4, if you access **Plan > Tasks > All tasks**, you will see a checkbox without a label. Perform the following steps to fix this issue:

1. Select **Settings > Plan Settings**.
2. In the **Administrative settings** page, scroll down to the **Other options** section and select **Unica Plan upgrade**.

3. Deselect all selections by selecting the tick mark preceding the column title **Module**.

4. Select **Upgrade configuration**.

5. Click **Upgrade**.

6. Log out and restart the application server.

7. Log back in to Unica Plan to see the changes.

Chapter 5. 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.

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:
    - 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 6. 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 9. Product names for configTool registration and unregistration

Product name	Name used in configTool
Unica Platform	`Manager`
Unica Campaign	`Campaign`
Unica Collaborate	`Collaborate`
Unica Deliver	`Deliver`
Unica Journey	`Journey`
Unica Insights	`UnicaInsights`
Unica Content Integration	`assetPicker`
Unica Offer	`Offer`
Unica Interact	`interact`
Unica Optimize	`Optimize`
Unica Plan	`Plan`
Opportunity Detect	`Detect`
IBM SPSS Modeler Advantage Enterprise Marketing Management Edition	`SPSS`
Digital Analytics	`Coremetrics`

-u *productName*

Unregister an application that is specified by *productName*. You do not have to include a path to the product category; the product name is sufficient, and it is required. The process removes all properties and configuration settings for the product.

Options

-o

When used with `-i` or `-r`, it overwrites an existing category or product registration (node).

When used with `-d`, you can delete a category (node) that does not have the **Delete Category** link on the **Configuration** page.

Examples

- Import configuration settings from a file named `Product_config.xml` in the `conf` directory under the Unica Platform installation.

  ```
  configTool -i -p "Affinium" -f Product_config.xml
  ```

- Import one of the supplied Unica Campaign data source templates into the default Unica Campaign partition, partition1. The example assumes that you placed the Oracle data source template, `OracleTemplate.xml`, in the `tools/bin` directory under the Unica Platform installation.

  ```
  configTool -i -p "Affinium|Campaign|partitions|partition1|dataSources" -f
  OracleTemplate.xml
  ```

- Export all configuration settings to a file named `myConfig.xml` in the `D:\backups` directory.

  ```
  configTool -x -f D:\backups\myConfig.xml
  ```

- Export an existing Unica Campaign partition (complete with data source entries), save it to a file named `partitionTemplate.xml`, and store it in the default `tools/bin` directory under the Unica Platform installation.

  ```
  configTool -x -p "Affinium|Campaign|partitions|partition1" -f
  partitionTemplate.xml
  ```

- Manually register an application named productName, by using a file named `app_config.xml` in the default `tools/bin` directory under the Unica Platform installation, and force it to overwrite an existing registration of this application.

  ```
  configTool -r product Name -f app_config.xml -o
  ```

- Unregister an application named productName.

  ```
  configTool -u productName
  ```

- Run the following command to enable encodeCSV feature:

```
configTool -vp -p "Affinium|Plan|umoConfiguration" -f Plan_Home\conf
\Plan_encodeProperty_11.1.xml
```

- Register Unica Interact Settings as configuration menu under `AffiniumWebApps` `\Campaign\interact\conf\interact_setup_navigation.xml` using

```
configTool.bat -v -i -p "Affinium|suite|uiNavigation|settingsMenu" -f
"interact_setup_navigation.xml"
```

Chapter 7. 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

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 | 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 new feature introduced in version 12.1.4, 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 90)* 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`