

Unica Collaborate V12.1.1 Upgrade Guide



Contents

Chapter 1. Upgrade overview.....	1
Upgrade roadmap.....	1
How the installers work.....	2
Modes of installation.....	3
Sample response files.....	4
Correcting the registry file when upgrade installation fails.....	6
Collaborate documentation and help.....	7
Chapter 2. Planning the Collaborate upgrade.....	10
Prerequisites.....	11
Upgrade prerequisites for all HCL Unica products.....	14
Collaborate Installation database information worksheet.....	15
Information for creating JDBC connections.....	17
Chapter 3. Upgrading Collaborate.....	24
Backing up Collaborate.....	24
Undeploying Collaborate.....	24
Shutting down and restarting the web application server.....	24
Installing in upgrade mode.....	24
Verifying the Collaborate upgrade.....	25
Chapter 4. Deploying Collaborate web application.....	27
Deploying Collaborate on WAS from a WAR file.....	27
Deploying Collaborate on WAS from an EAR file.....	29
Specifying generic JVM arguments.....	31
Deploying Collaborate on WebLogic.....	32

Running the Java upgrade.....	33
Disabling notification threads.....	33
Restoring icons and workflow files after upgrading your clustered environment.....	34
.....	35
.....	35
.....	35
Chapter 5. Uninstalling Collaborate.....	37
Chapter 6. configTool.....	40
Index.....	a

Chapter 1. Upgrade overview

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

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

Upgrade roadmap

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

You can use Collaborate Upgrade roadmap to scan the tasks that must be completed for upgrading Collaborate:

Table 1. Collaborate upgrade roadmap

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

Topic	Information
Upgrade overview (on page 1)	<p>This topic provides the following information:</p> <ul style="list-style-type: none">• How the installers work (on page 2)• Modes of installation (on page 3)• Collaborate documentation and help (on page 7)
Planning Collaborate upgrade (on page 10)	<p>This topic provides the following information:</p> <ul style="list-style-type: none">• Prerequisites (on page 11)• Upgrade prerequisites for all HCL Unica products (on page 14)

Table 1. Collaborate upgrade roadmap

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

(continued)

Topic	Information
	<ul style="list-style-type: none"> • Backing up Collaborate (on page 24) • Unica Collaborate installation database information worksheet (on page 15)
Upgrading Collaborate (on page 24)	<p>This topic provides the following information:</p> <ul style="list-style-type: none"> • Planning Collaborate upgrade (on page 10) • Undeploying Collaborate (on page 24) • Running the Collaborate installer (on page 24)
Deploying the Collaborate web application (on page 27)	<p>This topic provides the following information:</p> <ul style="list-style-type: none"> • Deploying Collaborate on WAS from a WAR file (on page 27) • Deploying Collaborate on WAS from an EAR file (on page 29) • Deploying Collaborate on WebLogic (on page 32)
Uninstalling Collaborate (on page 37)	<p>This topic provides information about how to uninstall Collaborate.</p>
configTool (on page 40)	<p>Read more about the Config tool utilities in Collaborate.</p>

How the installers work

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

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

- The suite installer and the product installer must be in the same directory on the computer where you want to install the product.
- If you are planning to install a patch immediately after you install an HCL 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 HCL 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 HCL 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 Collaborate.

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

GUI Window System mode

Use the GUI mode for Windows™ to install Collaborate by using the graphical user interface.

UNIX™ X Window System mode

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

Console mode

Use the console mode to install Collaborate 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 Collaborate 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 Collaborate. You can use sample response files to create your response files. The sample response files are included with the installers in the [ResponseFiles](#) compressed archive.

The following table provides information about sample response files:

Table 2. Description of sample response files

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

Table 2. Description of sample response files

(continued)

Sample response file	Description
	For example, <code>installer_uln.n.n.n.properties</code> is the response file of the Leads installer, where n.n.n.n is the version number.
<code>installer_report_pack initials, product ini- tials, and version num- ber.properties</code>	<p>The sample response file for the reports pack installer.</p> <p>For example, <code>installer_urpcn.n.n.n.properties</code> is the response file of the Campaign reports pack installer, where n.n.n.n is the version number</p> <p>For example, <code>installer_urpl.properties</code> is the response file of the Leads reports pack installer.</p>

Table 3. Description of sample response files

Sample response file	Description
<code>installer.properties</code>	The sample response file for the HCL Unica master installer.
<code>installer_product in- tials and product version number.prop- erties</code>	<p>The sample response file for the Collaborate installer.</p> <p>For example, <code>installer_ucn.n.n.n.properties</code> is the response file of the Campaign installer, where n.n.n.n is the version number.</p> <p>For example, <code>installer_umpn.n.n.n.properties</code> is the response file of the Unica Platform installer, where n.n.n.n is the version number.</p> <p>For example, <code>installer_uln.n.n.n.properties</code> is the response file of the Leads installer, where n.n.n.n is the version number.</p>



Note: After you install Collaborate successfully, copy the reports zip and model file from Collaborate installation directory, for example `<Collaborate_Home>/reports`, to the folder where Cognos® server and framework manager are installed, for example, `<cognos_installation_directory>/deployment`.

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 HCL Unica product is installed. The registry file tracks all installed HCL 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.

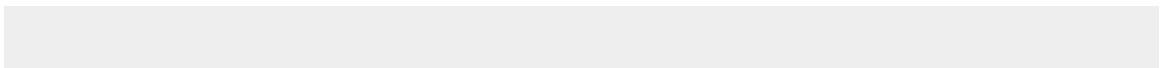
- Root user - `/var/`
- Non-root user - `$HOME/`

- On Mac OSX servers, the file is in the `/library/preferences/` folder.

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 Campaign version 8.6.0.3.



```
<product name="Campaign" id="dd6f88e0-1ef1-11b2-accf-c518be47c366"
version=" 8.6.0.3 " copyright="2013" info_url="" support_url=""
location="<HCL_Unica_Home>\Campaign" 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.

Collaborate documentation and help

Collaborate provides documentation and help for users, administrators, and developers.

The following table describes the various tasks of Collaborate. The **Documentation** column contains names of documents where you can find more information about the tasks.

Table 4. Get up and running

This two-columned table describes the Collaborate installation or upgrade tasks in one column, and the corresponding documents in the second column.

Task	Documentation
Find out about new features, known issues, and limitations	Collaborate Release Notes
Learn about the structure of the Collaborate system tables	Collaborate System Tables and Data Dictionary
Install or upgrade Collaborate	One of the following guides: <ul style="list-style-type: none"> • Collaborate Installation Guide • Collaborate Upgrade Guide
Implement the IBM® Cognos® reports provided by Unica Collaborate	Cognos Reports Installation and Configuration Guide
Implement Unica Insights reports provided by Unica Collaborate	Unica Insights Installation and Configuration Guide

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

Table 5. Configure and use Collaborate

This two-columned table describes the Collaborate administrative and user tasks in one column and the corresponding documents in the second column.

Task	Documentation
<ul style="list-style-type: none"> • Set up and configure the system for users • Adjust security settings • Map tables, define offer templates and custom attributes • Run utilities and perform maintenance 	<i>Unica Collaborate Administrator's Guide</i>
<ul style="list-style-type: none"> • Create and deploy marketing campaigns • Analyze campaign results 	Unica Collaborate Corporate Marketer's Guide Unica Collaborate Field Marketer's Guide
Use REST APIs	Unica Collaborate REST APIs

The following tables contain information about online help and obtaining PDFs for Collaborate. The **Instructions** column describes how to open online help and access documentation for Collaborate.

Table 6. Getting help

This two-columned table contains information about how to start online help and access PDF documentation for the Collaborate. Second column contains instructions for the tasks.

Task	Instructions
Open on-line help	<ol style="list-style-type: none"> 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. <p>You must have web access to view context-sensitive online help. Contact HCL Support for more information about accessing and installing Unica Knowledge Center locally for offline documentation.</p>
Obtain PDFs	<p>Use either of the following methods:</p> <ul style="list-style-type: none"> • Choose Help > Product documentation to access Collaborate PDFs. • Choose Help > All HCL Unica suite documentation to access all available documentation.
Unica Knowledge Center	To access the Unica Knowledge Center, choose Help > Support for this product .

Chapter 2. Planning Collaborate upgrade

Examine upgrade scenarios when you upgrade Collaborate. Upgrade scenarios provide guidelines for upgrading Collaborate to the specific version.

See the following table when you upgrade Collaborate.

Table 7. Version and upgrade paths

Source version	Upgrade path
Earlier than Collaborate 8.6.x	To upgrade to version 12.1.1, complete the following steps. 1. Upgrade to Collaborate 8.6.0. 2. Fast upgrade from version 8.6.0 to version 12.1.0 by following the instruction in this guide. 3. Regular upgrade from version 12.1.0 to version 12.1.1 by following instructions mentioned in this guide.
Collaborate versions 8.6.x and higher but earlier than version 12.1.0	To upgrade to version 12.1.1, complete the following steps. 1. Fast upgrade from "8.6.x and higher" to version 12.1.0 by following the instructions mentioned for Collaborate in Unica Fast Upgrade Guide. 2. Regular upgrade from version 12.1.0 to version 12.1.1 by following instructions mentioned in this guide.
Collaborate 12.1.0	To upgrade to version 12.1.1, complete the following steps. 1. Regular upgrade from 12.1.0 to 12.1.1 by following instructions mentioned in this guide.



Note:

To upgrade Collaborate to version 12.1.1 with MariaDB as a system database, complete the following steps.



1. Install Platform and Campaign, version 12.1.0.x.
2. Install Collaborate, version 12.1.0.4.



Note: Collaborate directly can be directly installed on Platform / Campaign 12.1.0.x.

3. Upgrade Platform/Campaign version 12.1.0.x to version 12.1.1.
4. Upgrade Collaborate, version 12.1.0.4 to version 12.1.1.

Prerequisites

Before you install or upgrade any HCL 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 HCL Unica products that are installed as a suite must be installed on the same network domain to comply with the browser restrictions that are designed to limit the security risks that can occur with cross-site scripting.

JVM requirements

HCL Unica applications within a suite must be deployed on a dedicated Java™ virtual machine (JVM). HCL Unica products customize the JVM that is used by the web application server. If you encounter errors that are related to the JVM, you must create an Oracle Weblogic domain or IBM Websphere profile that is dedicated to the HCL Unica products.

Knowledge requirements

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

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

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

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

The following additional permissions are necessary for UNIX™:

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

JAVA_HOME environment variable

If a `JAVA_HOME` environment variable is defined on the computer where you install an HCL Unica product, verify that the variable points to a supported version of JRE. For information about system requirements, see the HCL 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 HCL 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 HCL Unica installer installs a JRE in the top-level directory for the HCL Unica installation. Individual HCL Unica application installers do not install a JRE. Instead, they point to the location of the JRE that is installed by the HCL Unica installer. You can reset the environment variable after all installations are complete.

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

Platform requirement

You must install or upgrade Platform before you install or upgrade any HCL Unica products. For each group of products that work together, you must install or upgrade Platform only

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

Campaign requirement

You must install or upgrade and configure Campaign before you install or upgrade Collaborate.

Upgrade prerequisites for all HCL Unica products

Meet all requirements for permissions, operating system, and knowledge correctly before you upgrade Collaborate 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 Collaborate 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 Collaborate, 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.

Unica Collaborate installation database information worksheet

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

Use the Unica Collaborate installation database information worksheet to gather information about the database that contains your Collaborate system tables.

Table 8. Collaborate database information worksheet

Following table contains Field and Notes columns.

Field	Notes
Database type	
Database name	

Table 8. Collaborate database information worksheet

Following table contains Field and Notes columns.

(continued)

Field	Notes
Database account user name	
Database account password	
JNDI name	
ODBC name	

Checklist for Platform

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

- Database type
- JNDI name
- JDBC driver class
- JDBC connection URL
- JDBC driver class path on your computer
- Database host name
- Database port
- Database name or schema ID
- User name and password for the database account

Information about web component

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

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

Information for creating JDBC connections

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



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

WebLogic

Use the following values if your application server is WebLogic:

SQLServer

- Database Driver: Microsoft™ MS SQL Server Driver (Type 4) Versions: 2012, 2012 SP1 and SP3, 2014, 2014 SP1, 2016 SP1
- Default port: 1433
- Driver class: `com.microsoft.sqlserver.jdbc.SQLServerDriver`
- Driver URL: `jdbc:sqlserver://<your_db_host>[\<named_instance>]:<your_db_port>;databaseName=<your_db_name>`
- Properties: Add `user=<your_db_user_name>`

Oracle

- Driver: Other
- Default port: 1521

- Driver class: `oracle.jdbc.OracleDriver`

- Driver URL:

`jdbc:oracle:thin:@<your_db_host>:<your_db_port>:<your_db_service_name>`

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

- Properties: Add `user=<your_db_user_name>`

DB2®

- Driver: Other
- Default port: 50000
- Driver class: `com.ibm.db2.jcc.DB2Driver`
- Driver URL: `jdbc:db2://<your_db_host>:<your_db_port>/<your_db_name>`
- Properties: Add `user=<your_db_user_name>`

MariaDB

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

WebSphere®

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

SQLServer

- Driver: N/A
- Default port: 1433

- Driver class: `com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource`
- Driver URL: N/A

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

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

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

Add the following custom property:

Name: `webSphereDefaultIsolationLevel`

Value: 1

Datatype: Integer

Oracle

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

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

DB2®

- Driver: JCC Driver Provider
- Default port: 50000
- Driver class: `com.ibm.db2.jcc.DB2Driver`
- Driver URL: `jdbc:db2://<your_db_host>:<your_db_port>/<your_db_name>`

Add the following custom property:

Name: `webSphereDefaultIsolationLevel`

Value: 2

Datatype: Integer

MariaDB

- Database Driver: MariaDB JDBC Driver
- Default Port: 3306
- Driver Class: `org.mariadb.jdbc.MySQLDataSource`
- Driver URL: `org.mariadb://<your_db_host> [\`
`\<named_instance>]:<your_db_port>;databaseName=<your_db_name>, valid-`
`connection-checker class`
`name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker"`

For example: `/subsystem=datasources/data-source=UnicaPlatformDS:add(jndi-`
`name="java:/UnicaPlatformDS",connection-url="jdbc:mariadb://`
`localhost:3306/plat11",driver-name=MariaDBDriver,user-`
`name=root,password=test123,valid-connection-checker-class-`
`name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker")`

JBoss

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

Use the following values if your application server is JBoss:

SQLServer

- Database Driver: Microsoft MS SQL Server Driver (Type 4) Versions: 2012, 2012 SP1 and SP3, 2014, 2014 SP1, 2016 SP1
- Default port: 1433
- Driver class: `com.microsoft.sqlserver.jdbc.SQLServerDriver`

- Driver URL: `jdbc:sqlserver://<your_db_host>[\
\<named_instance>]:<your_db_port>;databaseName=<your_db_name>,valid-connection-checker-class-name=org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker`

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

Oracle

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

For example

DB2

- Driver: JCC Driver Provider
- Default port: 50000
- Driver class: `com.ibm.db2.jcc.DB2Driver`
- Driver URL: `jdbc:db2://<your_db_host>:<your_db_port>/<your_db_name>`

MariaDB

- Database Driver: MariaDB JDBC Driver
- Default Port: 3306
- Driver Class: `org.mariadb.jdbc.MySQLDataSource`
- Driver URL: `org.mariadb://<your_db_host> [\
\<named_instance>]:<your_db_port>;databaseName=<your_db_name>, valid-`


```
connection-checker class
```

```
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker"
```

For example: /subsystem=datasources/data-source=UnicaPlatformDS:add(jndi-

```
name="java:/UnicaPlatformDS",connection-url="jdbc:mariadb://
```

```
localhost:3306/plat11",driver-name=MariaDBDriver,user-
```

```
name=root,password=test123,valid-connection-checker-class-
```

```
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker")
```

Tomcat

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



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

Use the following values if your application server is Tomcat:

SQLServer

- Database Driver: Microsoft MS SQL Server Driver (Type 4) Versions: 2012, 2012 SP1 and SP3, 2014, 2014 SP1, 2016 SP1
- Default port: 1433
- Driver class: `com.microsoft.sqlserver.jdbc.SQLServerDriver`
- Driver type: `javax.sql.DataSource`
- Driver URL: `jdbc:sqlserver://<your_db_host>[\ \<named_instance>]:<your_db_port>;databaseName=<your_db_name>`

Oracle

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

DB2

- Driver: JCC Driver Provider
- Default port: 50000
- Driver class: `com.ibm.db2.jcc.DB2Driver`
- Driver URL: `jdbc:db2://<your_db_host>:<your_db_port>/<your_db_name>`

MariaDB

- Database Driver: MariaDB JDBC Driver
- Default Port: 3306
- Driver Class: `org.mariadb.jdbc.MySQLDataSource`
- Driver URL: `org.mariadb://<your_db_host> [\`
`\<named_instance>]:<your_db_port>;databaseName=<your_db_name>, valid-`
`connection-checker class`
`name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker"`

For example: `/subsystem=datasources/data-source=UnicaPlatformDS:add(jndi-`
`name="java:/UnicaPlatformDS",connection-url="jdbc:mariadb://`
`localhost:3306/plat11",driver-name=MariaDBDriver,user-`
`name=root,password=test123,valid-connection-checker-class-`
`name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker")`

Chapter 3. Upgrading Collaborate

Before you upgrade from any version of Collaborate, be sure that you understand all the steps about how to upgrade Collaborate to the new version.

Backing up Collaborate

Back up all files and the Collaborate database before you begin the upgrade process. If the upgrade process fails, a backup provides a way to restore to a known working state.

Undeploying Collaborate

When you undeploy Collaborate, the web application server releases the lock on the WAR file from your Collaborate source system. The source system is updated by the Collaborate upgrade installation.

After you undeploy the Collaborate, upgrade process registers the new version of Collaborate in Platform.

Shutting down and restarting the web application server

After you undeploy Unica Collaborate, shut down and restart the web application server to ensure that the lock on the WAR file is released.

Running the Collaborate installer

When you run the Collaborate installer, the installer detects your existing version of Collaborate and prompts you to confirm the upgrade process. After your confirmation, the installer starts the upgrade process.

Follow the guidelines while you upgrade Collaborate:

- When the installer prompts you for the installation location, specify the location of the parent directory of the Collaborate installation directory.
- The upgrade process updates the previously migrated registration information for the new version of Collaborate.

The Collaborate upgrade installer overwrites `setenv.sh/bat` and `udm_jdbc.properties` in the `<Collaborate Home>/tools/bin` directory. If your installation is using the manual database set up, the files contain the database connection information. When you upgrade, add the database information in these files before starting the manual upgrade process.



Note: For detailed installation steps to install Collaborate in upgrade mode, see "How installers work" in the *Collaborate Installation Guide*.

Verifying the Collaborate upgrade

After the upgrade process completes, you must verify that your Collaborate application is upgraded.

Complete the following steps to verify the Collaborate upgrade:

1. After you finish the upgrade, check the `installation log`, `stdout`, and `stderr` files for details.
2. Log in to the Collaborate application, and check whether Collaborate settings are accessible.
3. Check **Help > About version** to verify your version of Collaborate.
4. Verify that List and On-demand Campaign (ONDC) instances are accessible.

Verifying the Collaborate database upgrade

For a database upgrade, you can select manual upgrade or automatic upgrade.

If you are upgrading manually, update the unencrypted value for

`udm_data_source.password` in the `udm_jdbc.properties` file that is located in the `<Collaborate_Home>/tools/bin` directory before you run `udmdbsetup`.

To verify that your database is upgraded successfully, complete the following actions for automatic and manual database upgrade:

- Verify that you get the successful upgrade message on the upgrade wizard complete window. At the end of upgrade process, you must not have any errors.
- You must check `installation log`, `stderr`, and `stdout` files for details.
- Run `udmdbsetup` manually by using the upgrade parameter. You must use the correct parameter values for your locale, installation type, and version you are upgrading from. The following example creates and populates the Collaborate system tables for an upgrade from version 10.1 on UNIX in the `fr_FR` locale:

```
udmdbsetup.sh -b 10.1.x.x -L fr_FR -t upgrade -v
```



Note: When you upgrade Collaborate, you are normally not required to run `udmdbsetup` manually. The installer runs `udmdbsetup` in both full and upgrade modes.

- If you are running a full installation, the installer runs `udmdbsetup` in full mode (`-t full`), and populates all system tables on Collaborate system tables database.
- If you are running an upgrade, the installer runs `udmdbsetup` in upgrade mode (`-t upgrade`), and upgrades existing and populates new tables as required.
- The `udmdbsetup` utility updates the database directly only in the case of automated installation.
- If you are running in manual mode (`-m` flag), the `udmdbsetup` utility generates a full installation or an upgrade script in the `DM_HOME\tools` folder. The name of this script depicts whether it was generated for full or upgrade installation; that is, for full installation, the name is `install_<version>.sql`, or for an upgrade installation, the name is `upgrade_<baseVersion>_<currentVersion>.sql`.

Chapter 4. Deploying the Collaborate web application

You can deploy the web application by either including Collaborate in an EAR file or by deploying the Collaborate WAR file.

Use the following guidelines to deploy Collaborate:

- When you run the Unica installer, you might include Collaborate in an EAR file, or you can choose to deploy the Collaborate WAR file. If you included the Platform or other products in an EAR file, you must follow all the deployment guidelines that are detailed in the individual installation guides for the products included in the EAR file.
- Set the JSP Compilation level in the WebSphere application server to 18.
- You must know how to work with web application server. Consult your web application server documentation for details such as navigation in the Administration console.



Note: In case of Collaborate 12.1.0.4 and higher, remove the following JVM parameters, if they are set for application server.

- `-Dlog4j.configurationFile`
- `-Dlog4j2.configurationFile`
- `-Dlog4j2.contextSelector`

Deploying Collaborate on WAS from a WAR file

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

Complete the following tasks before you deploy Collaborate:

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

1. Go to the WebSphere® Integrated Solutions Console.
2. Complete the following steps if your system tables are in DB2®:
 - a. Click the data source that you created. Go to the Custom Properties for the data source.
 - b. Select the Custom properties link.
 - c. Set the value for the **resultSetHoldability** property to 1.

If you do not see the **resultSetHoldability** property, create the **resultSetHoldability** property and set its value to 1.
3. Go to **Applications > Application Types > WebSphere enterprise applications** and click **Install**.
4. In the **Preparing for the application installation** window, select the **Detailed - Show all options and parameters** check box and click **Next**.
5. Click **Continue** to see the **Install New Application** wizard.
6. Accept the default settings on the windows of the **Install New Application** wizard except the following windows:
 - In step 1 of the **Install New Application** wizard, select the **Precompile JavaServer Pages files** check box.
 - In step 3 of the installation wizard, set the **JDK Source Level** to 18.
 - In step 9 of the installation wizard, set the **Context Root** to /collaborate.
7. In the left navigation panel of WebSphere® Integrated Solutions Console, navigate to **Applications > Application Types > WebSphere enterprise applications**.
8. In the Enterprise Applications window, click the **collaborate.war** file.
9. In the **Web Module Properties** section, click **Session Management** and select the following check boxes:
 - **Override session management**
 - **Enable Cookies**

10. Click **Enable Cookies**, and in the **Cookie name** field, enter a unique cookie name.
11. If you are using version 8 of WebSphere® Application Server, select **Servers > WebSphere application server > server 1 > Session management > Enable Cookies** and clear the check box for **Set session cookies to HTTPOnly to help prevent cross-site scripting attacks**.
12. In the **Applications > Enterprise Applications** section of the server, select the WAR file that you deployed.
13. In the **Detail Properties** section, select **Class loading and update detection**.
14. In the **Class loader order** section, select the **Classes loaded with local class loader first (parent last)** option.
15. In the **Modules** section, select **Manage modules > detect**, and under **Class loader order** select the **Classes loaded with local class loader first (parent last)** option.
16. Enter the context root for the application as `/collaborate`.
17. Go to **Application Servers > your server > Process definition > Java Virtual Machine**.
18. In the **Generic JVM arguments** section, enter the following JVM arguments:

```
-Dcom.ibm.websphere.webservices.DisableIBMJAXWSEngine=true
-Dibm.cl.verbose=PersistenceProvider
-Dibm.cl.verbose=PersistenceProviderImpl
-agentlib:getClasses -verbose:ynload
-Dcom.ibm.xml.xlsp.jaxb.opti.level=3
```

Deploying Collaborate on WAS from an EAR file

You can deploy Collaborate by using an EAR file if you included Collaborate in an EAR file when you ran the HCL Unica installer.

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

1. Go to the WebSphere® Integrated Solutions Console.
2. Complete the following steps, if your system tables are in DB2®:

- a. Click the data source that you created. Go to the **Custom Properties** for the data source.
 - b. Select the Custom properties link.
 - c. Set the value for the **resultSetHoldability** property to 1.
- If you do not see the **resultSetHoldability** property, create the **resultSetHoldability** property and set its value to 1.
3. Go to **Applications > Application Types > WebSphere enterprise applications** and click **Install**.
4. In the **Preparing for the application installation** window, select the **Detailed - Show all options and parameters** check box and click **Next**.
5. Click **Continue** to see the **Install New Application** wizard.
6. Accept the default settings on the windows of the Install New Application wizard except the following windows:
 - In step 1 of the **Install New Application** wizard, select the **Precompile JavaServer Pages files** check box.
 - In step 3 of the installation wizard, set the **JDK Source Level** to 18.
 - In step 8 of the installation wizard, set the **Context Root** to /collaborate.
7. In the left navigation panel of WebSphere® Integrated Solutions Console, navigate to **Applications > Application Types > WebSphere enterprise applications**.
8. In the **Enterprise Applications** window, select the EAR file that you want to deploy.
9. In the **Web Module Properties** section, click **Session Management** and select the following check boxes:
 - **Override session management**
 - **Enable Cookies**
10. Click **Enable Cookies**, and in the **Cookie name** field, enter a unique cookie name.
11. If you are using version 8 of WebSphere® Application Server, select **Servers > WebSphere application server > server 1 > Session management > Enable Cookies** and clear the check box for **Set session cookies to HTTPOnly to help prevent cross-site scripting attacks**.
12. In the **Detail Properties** section, select **Class loading and update detection**.

13. In the **Class loader order** section, select the **Classes loaded with local class loader first (parent last)** option.
14. In `Collaborate.ear`, open the **Manage Modules** section and select the **Classes loaded with local class loader first (parent last)** option.
15. For **WAR class loader policy**, select the **Class loader for each WAR file in application** option.
16. In WebSphere Enterprise Applications, select ***Your Application* > Manage Modules > *Your Application* > Class Loader Order > Classes loaded with local class loader first (parent last)**.
17. Go to **Application Servers > your server > Process definition > Java Virtual Machine**.
18. In the **Generic JVM arguments** section, enter the following JVM arguments:

```
-Dcom.ibm.websphere.webservices.DisableIBMJAXWSEngine=true
-Dibm.cl.verbose=PersistenceProvider
-Dibm.cl.verbose=PersistenceProviderImpl
-agentlib:getClasses -verbose:dynload
-Dcom.ibm.xml.xlsp.jaxb.opti.level=3
```

For more information about WebSphere® Application Server version 8.5, see [Welcome to the WebSphere® Application Server information center](#).

Specifying generic JVM arguments

The generic Java™ Virtual Machine (JVM) arguments are used to configure and adjust how the JVM runs in the application. Specify the generic JVM arguments while you deploy Collaborate on WebSphere®.

Specify following generic JVM arguments in the server's JVM properties field:

- `-Dcollaborate.home= <Collaborate installation directory> -noverify`
- `-Dclient.encoding.override=UTF-8`
- Set the memory heap size parameters for the JVM by entering 1024 in both the **Initial Heap Size** and **Maximum Heap Size** fields.

Deploying Collaborate on WebLogic

You can deploy HCL Unica products on WebLogic.

Use the following guidelines when you deploy Collaborate on WebLogic:

- HCL Unica products customize the JVM used by WebLogic. You might need to create a WebLogic instance that is dedicated to HCL Unica products if you encounter JVM-related errors.
- Verify that the SDK selected for the WebLogic domain you are using is the Sun SDK by looking in the startup script (`startWebLogic.cmd`) for the `JAVA_VENDOR` variable. It should be set to: `JAVA_VENDOR=Sun`. If it is set to `JAVA_VENDOR=BEA`, JRockit has been selected. JRockit is not supported. To change the selected SDK, refer to the WebLogic documentation.
- Deploy the HCL Unica products as web application modules.
- On UNIX™ systems, you must start WebLogic from the console to allow correct rendering of graphical charts. The console is usually the machine on which the server is running. However, in some cases the web application server is set up differently.

If a console is not accessible or does not exist, you can emulate a console using Exceed. You must configure Exceed so that your local Xserver process connects to the UNIX™ machine in root window or single window mode. If you start the web application server using Exceed, you must keep Exceed running in the background to allow the web application server to continue running. Contact the Technical Support for detailed instructions if you encounter problems with chart rendering.

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

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

```
-Dcollaborate.home=Collaborate installation directory  
-Dfile.encoding=UTF-8
```

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

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

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

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

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

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

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

Running the Java upgrade

After you deploy Collaborate on either a stand-alone environment or a clustered environment, you must run the Collaborate Java upgrade.

To run the Collaborate Java upgrade, perform the following steps.

1. Log in to Unica Collaborate and go to **Collaborate Settings**.
2. Click **Collaborate Upgrade**.
3. Run the upgrade process after you select the appropriate options for the upgrade.
4. Restart Unica Collaborate.

Disabling notification threads

You must disable notification threads if you are running a notification on a single cluster.

If you are running a notification on a single node in a cluster, you must provide the following JVM parameter to disable notification threads on a particular cluster node.

```
-Ddm.suppressNotifications=true
```

- If the WebSphere Application Server cluster has four nodes, then you need to add the JVM parameter in three JVM cluster nodes except the one you want to run the notifications.
- If the WebSphere Application Server cluster has four nodes and you added this JVM parameter in two nodes and notification threads are enabled on more than one cluster node, then there is a chance multiple notifications are sent.

Restoring icons and workflow files after upgrading your clustered environment

If you have a default or relative path for the **templatesDir** configuration parameter, you must restore all your lost icons and workflow XML files after you upgrade your WebSphere® Application Server clustered environment.

1. Back up your templates directory under the exploded war directory before redeployment of `collaborate.war`.
2. Change the value of the **templatesDir** configuration parameter to a valid file system path.
For example, `Collaborate_Home/templates`.
3. Stop your application server.
4. Copy the backup template directory to the path pointing to the value of **templatesDir**.
5. Redeploy the new `collaborate.war` file, and verify that all the icons and template workflows and template attachments are shown properly.

The exploded war directory can be located at

```
<path_to_WebSphere_installation>/AppServer/profiles/  
<profileName>/installedApps/node01cell/collaborate.ear/  
collaborate.war.
```

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 HCL 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 HCL Unica applications the context path is typically `/unica`.

Chapter 5. Uninstalling Collaborate

Run the Collaborate uninstallers to uninstall Collaborate. When you run the Collaborate 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 HCL Unica products, an uninstaller is included in the `Uninstall_Product` directory, where Product is the name of your Unica 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 Unica product in the same location. After uninstalling a product, its database is not removed. The uninstaller only removes default files that are created during installation. Any file that is created or generated after installation is not removed.

In addition to the general instructions for uninstalling HCL Unica products, follow the guidelines when you uninstall Collaborate:

- If you have multiple Collaborate runtime installations using the same Platform installation, you must remove network connectivity for the Collaborate runtime workstation before you run the uninstaller. Failure to do so uninstalls the configuration data for all the other Collaborate runtime installations from Platform.
- You can safely ignore any warnings about failure to unregister with Platform.
- You can export a copy of your configuration as a precaution before uninstalling Collaborate.
- If you uninstall the Collaborate design time environment, after you run the uninstaller, you may need to manually unregister Collaborate. Use the configtool utility to unregister `full_path_to_Collaborate_installation_directory\Collaborate\conf\collaborate_navigation.xml`.



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

1. If you have deployed the Collaborate web application, undeploy the web application from WebSphere® or WebLogic.
2. Shut down WebSphere® or WebLogic.
3. Stop the processes that are related to Collaborate.
4. If the `ddl` directory exists in the product installation directory, run the scripts that are provided in the `ddl` directory to drop tables from the system table database.
5. Make sure that the IBM® SPSS® Modeler Collaboration and Deployment Service (C&DS) is uninstalled.

Open the Installation Manager and verify that C&DS is uninstalled. If C&DS is not uninstalled, use the IBM® Installation Manager to uninstall C&DS.

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


```
Uninstall_Product -i console
```
 - If you are uninstalling on a UNIX™ environment, navigate to the directory where the uninstaller exists, and run the following command to uninstall Collaborate:


```
./Uninstall\ Interaction\ History.
```



Note: UNIX™ requires spaces after \.

- In a command-line window, navigate to the directory where the uninstaller exists, and run the following command to uninstall Collaborate by using the silent mode:

Uninstall `_Product` -i silent

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

Chapter 6. configTool

The properties and values on the **Configuration** page are stored in the 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 Campaign, which you can then modify and duplicate by using the **Configuration** page.
- To register (import configuration properties for) HCL 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 HCL 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 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 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 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 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 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 Platform installer as described in the *Platform Installation Guide*.
- After the initial installation, to re-register products other than the 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 HCL 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
Platform	Manager
Campaign	Campaign
Collaborate	Collaborate
Deliver	emessage
Interact	interact
Optimize	Optimize
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 Platform installation.

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

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


Index

C

configTool

40

configTool utility

40

U

utilities

configTool

40