



# HP Universal CMDB

Software Version: 10.20

## Upgrader Reference

Document Release Date: January 2015  
Software Release Date: January 2015

## Legal Notices

### Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

### Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

### Copyright Notice

© 2002 - 2015 Hewlett-Packard Development Company, L.P.

### Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

## Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: <https://softwaresupport.hp.com>

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: <http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

## Support

Visit the HP Software Support Online website at: <https://softwaresupport.hp.com>

This website provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support website to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

To find more information about access levels, go to:

[http://h20230.www2.hp.com/new\\_access\\_levels.jsp](http://h20230.www2.hp.com/new_access_levels.jsp)

**HP Software Solutions Now** accesses the HPSW Solution and Integration Portal website. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this website is <http://h20230.www2.hp.com/sc/solutions/index.jsp>

# Contents

Introduction .....	4
Upgrader Reference .....	5
Troubleshooting and Warning Logs for Resource Upgrade .....	21
Send Documentation Feedback .....	23

## Introduction

This document describes each step in the UCMDB upgrade process.

For each step, the following is described:

- A description of the step.
- Whether the step is critical. A step is considered critical in the following cases:
  - Skipping it would prevent the UCMDB server from starting after upgrade.
  - Skipping it would induce critical configuration or data loss that cannot be restored after upgrade.
  - Skipping it would prevent a critical component from operating properly after the upgrade.
- Whether the step can be re-run. In case of failure during the upgrade, whether or not this step can be re-run over the same schemas.
- Implications of failure. If this upgrade step fails, what is the effect on the UCMDB? If the step can be re-run, what can be done to resolve the issues?
- Log files: Important messages from the log file that are typical to this upgrade step, and the meaning of each message. Unless otherwise specified, all messages appear in the following log file:  
**C:\hp\UCMDB\UCMDBServer\runtime-upgrade\log\upgrade.short.log**

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
SchemaAdditionsUpgrader	Adds the new required tables and columns to the UCMDB.	✓	✓	UCMDB server cannot function properly if this upgrader fails.	upgrade.detailed.log
Global Settings Resources Upgrader	Upgrades all global settings from the old settings table (CCM_SETTINGS) to the new table: (URM_RESOURCES).	✓	✓	<p><b>Upgrade:</b> UCMDB server will not function properly</p> <p><b>Cleanup:</b> CCM settings table will not be dropped.</p> <p><b>Note:</b> This step cannot be rerun if the cleanup has already taken place because the old table is dropped the first time round.</p>	<ul style="list-style-type: none"> <li>• upgrade.detailed.log</li> <li>• urm.log</li> <li>• urm_detailed.log</li> <li>• urm_audit.log</li> <li>• urm_detailed_audit.log</li> </ul>
Customer Settings Resources Upgrader	Upgrades all customer specific settings from the old settings table (CCM_SETTINGS) to the new table: (URM_RESOURCES).	✓	✓	<p><b>Upgrade:</b> UCMDB server will not function properly</p> <p><b>Cleanup:</b> CCM settings table will not be dropped.</p> <p><b>Note:</b> This step cannot be rerun if the cleanup has already taken place because the old table is dropped the first time round.</p>	<ul style="list-style-type: none"> <li>• upgrade.detailed.log</li> <li>• urm.log</li> <li>• urm_detailed.log</li> <li>• urm_audit.log</li> <li>• urm_detailed_audit.log</li> </ul>
Dynamic Settings Upgrader	Converts dynamic settings to URM	✓	✓	Dynamic settings are not migrated	<ul style="list-style-type: none"> <li>• upgrade.detailed.log</li> <li>• urm.log</li> <li>• urm_detailed.log</li> <li>• urm_audit.log</li> <li>• urm_detailed_audit.log</li> </ul>
SyncDataStore Sequence Upgrader	Fixes the SyncDataStore sequence name. Renames the SyncDataStore sequence to new name convention, without "SEQUENCE_" prefix, and merges old and new sequence values.	✗	✓	<p>Push adapter may fail.</p> <p>This may happen only in the case where the user runs the push adapter in the 9.05.</p>	<ul style="list-style-type: none"> <li>• upgrade.log</li> <li>• upgrade.detailed.log</li> <li>• error.log</li> </ul>

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
Database Privileges Upgrader	Checks if the mandatory 'create type' grant exists.	✓	✓	<p>Fails the upgrade process with the following error message:</p> <p><b>"The following privilege is missing: 'GRANT CREATE TYPE TO &lt;USER_NAME&gt;"</b></p>	<ul style="list-style-type: none"> <li>• upgrade.log</li> <li>• upgrade.detailed.log</li> <li>• error.log</li> </ul>
Class Model Resources Upgrader	Migrates all the class model resources from the old format (held in CCM_* tables in version 9.x) into the URM (Unified Resource Manager).	✓	✗	<p><b>Upgrade:</b> Old class model is not imported into the URM, runs once per customer.</p> <p><b>Cleanup:</b> Deletes the CCM tables that are used in version 9.x to store the class model, runs once per schema</p>	<ul style="list-style-type: none"> <li>• upgrade.log</li> <li>• error.log</li> <li>• urm.log</li> <li>• urm_detailed.log</li> <li>• urm_audit.log</li> <li>• urm_detailed_audit.log</li> </ul>
Class Model Extensions Upgrader	Copies the meta-data of class model extensions from the table where it was stored in version 9.x (table name CLASS_EXTENSION) into the (Unified Resource Manager).	✓	✗	<p><b>Upgrade:</b> Class model meta data will not be imported into the URM, runs once per customer.</p> <p><b>Cleanup:</b> Drops the CLASS_EXTENSION table, runs once per schema.</p>	<ul style="list-style-type: none"> <li>• upgrade.log</li> <li>• error.log</li> </ul>
Consolidate Mode Upgrader	Consolidated mode is not supported as of ver. 10 of UCMDB. This upgrader 'isolates' consolidated customers.	✓	✓	<p><b>Upgrade:</b> The customer that was in consolidated mode in 9.x will not be isolated.</p> <p><b>Cleanup:</b> The data (tables, views) about the consolidated customer will remain in the database</p>	<ul style="list-style-type: none"> <li>• upgrade.detailed.log</li> </ul>
Data Model Capacity Upgrader	<p>Aligns the tables in the database for the data model in version 10.xx as follows:</p> <ul style="list-style-type: none"> <li>• Partitions the CMDB data model (CDM) table in the database for the root CI type into two tables: one for CIs and another for relations.</li> <li>• For every CI type derived from the <b>Node</b></li> </ul>	✓	✓	UCMDB server cannot function properly if this upgrader fails.	<ul style="list-style-type: none"> <li>• upgrade.log</li> <li>• upgrade.detailed.log</li> </ul>

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
	<b>Element</b> CI type: Copy relations of type <b>Composition</b> that are connected to the CIs of this CI type from the CDM table for composition relations to the CDM table of this CI type.				
Package And Resource Upgrader	Moves the packages meta-data from being represented as CIs (in version 9.x) to the URM	✓	✓	Package meta-data is not migrated	<ul style="list-style-type: none"> <li>• upgrade.detailed.log</li> <li>• urm.log</li> <li>• urm_detailed.log</li> <li>• urm_audit.log</li> <li>• urm_detailed_audit.log</li> </ul>
Check cp and sp version are correct for upgrade	Ensures that the Content Pack and the service pack of the version from which the upgrade is being run are correct for the upgrade. It is meant to prevent upgrades on systems of unsupported versions of content and service packs. The upgrader does not actually make any modifications, it only performs a check.  If the criteria are not satisfied, the upgrade fails.	✓	✓	The system configuration of the upgraded UCMDB is not correct. In this case, the upgrade cannot continue.	upgrade.log
Offline Resources Retriever	Extracts queries, views, reports, enrichments, correlations, folders and bundles from the database and stores them to disk. The resources are stored in <b>C:\hp\UCMDB\UCMDBServer\runtime\&lt;Customer ID&gt;\&lt;resource type&gt;\</b>	✓	✗	<p><b>Upgrade:</b> The failed resources will not be included in the environment after the upgrade.</p> <p><b>Cleanup:</b> The resources will remain as garbage in the Model.</p>	<ul style="list-style-type: none"> <li>• upgrade.short.log</li> <li>• upgrade.detailed.log</li> <li>• urm.log</li> <li>• urm_detailed.log</li> <li>• urm_audit.log</li> <li>• urm_detailed_audit.log</li> </ul> <p>Will contain the number of resources per type that were stored on disk, and warning and error messages in case there is a problem with a resource.</p>
Update internal class model from resources	Updates the internal class model from the	✓	✓	Varies, depending on what exact	<ul style="list-style-type: none"> <li>• cmdb.classmodel.log</li> </ul>

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
	xml resources			resource has failed	<ul style="list-style-type: none"> <li>error.log</li> </ul>
Upgrade reconciliation resources	Upgrades CIT identification rules and reconciliation priority definitions of external data sources.	✓	✓	CI reconciliation will not work properly when data is populating UCMDB.	<ul style="list-style-type: none"> <li>upgrade.detailed.log</li> <li>urm.log</li> <li>urm_detailed.log</li> <li>urm_audit.log</li> <li>urm_detailed_audit.log</li> </ul>
Migrate discovery resources to URM	Migrates Discovery resource from UCMDB to URM	✓	✓	Discovery resources can be lost – as a results all discovery flows can become unstable	<ul style="list-style-type: none"> <li>Upgrade.detailed.log</li> <li>Mam.autodiscovery.log</li> <li>Error.log</li> </ul>
Remove result mechanism from integration jobs	Replaces the <b>result mechanism</b> tag with <b>remove-cis</b> tag in integration jobs.	✓	✓		<ul style="list-style-type: none"> <li>Upgrade.detailed.log</li> <li>Mam.autodiscovery.log</li> <li>Error.log</li> </ul>
Decrypt Domain Scope Document (DSD)	Decrypts the Domain Scope Document (DSD)	✗	✓	Credentials and Probe ranges can be lost. <b>Note:</b> For backup, export the Domain Scope Document before performing the upgrade.	<ul style="list-style-type: none"> <li>mam.autodiscovery.log</li> <li>error.log</li> </ul>
Extract Domain Scope Document (DSD) ranges to Domain Ranges Document (DRD)	Extracts Probe ranges from the Domain Scope Document configuration file and copies them to the Domain Ranges Document configuration file.	✗	✓	Probe ranges can be lost. <b>Note:</b> For backup, export the Domain Scope Document before performing the upgrade.	<ul style="list-style-type: none"> <li>mam.autodiscovery.log</li> <li>error.log</li> </ul>
History Upgrader	<ul style="list-style-type: none"> <li>Runs the following steps for every CI type that is audited in the history. <ul style="list-style-type: none"> <li>Creates all the necessary monthly history tables (HDM/HDML) depending on the dates of the events for this CI type in the old history and the <i>history.purging.months.to.save.back</i> setting.</li> </ul> </li> </ul>	✓	✓	<p>If a CI type's history is not upgraded:</p> <ul style="list-style-type: none"> <li>Model update and history operations will fail since history tables will not be created for this CI type.</li> <li>No history events from the old history will be upgraded to the new</li> </ul>	<p>upgrade.detailed.log</p> <p>Logs the following:</p> <ol style="list-style-type: none"> <li>Upgrade progress</li> <li>History table creations</li> <li>For every CI type, the number of CIs that will be upgraded and the</li> </ol>

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
	<ul style="list-style-type: none"> <li>○ Upgrades each event that should not have been purged from the old history to the new history tables.</li> <li>○ Inserts into the history root table a record for every CI that had at least one history event upgraded to the new history.</li> <li>● Purges the data while executing based on the <i>history.purging.months.to.save.back</i> setting, meaning that history events that should be purged would not be copied to the new history tables.</li> <li>● The <i>history.purging.days.to.save.back</i> setting from 9.0x is converted to months (rounding up) and is saved in the <i>history.purging.months.to.save.back</i> setting.</li> </ul>			<p>history for this CI type.</p> <p>After the upgrade is finished and the server is running, you can run the <b>alignHistoryForType</b> operation from the history JMX services to create the history tables for a specific CI type that the history upgrader failed to upgrade or for all the CI types.</p> <p>You can also run <b>initializeHistoryDBFromModel</b> using the JMX console to create baseline records for all classes, based on current data model state. This operation ensures that queries return correct previous values and data of removed CIs. It may take several minutes and creates baseline events even for classes that upgraded properly (does not affect the history data correctness).</p> <p>In this case, the Model update and history operations will not fail, but the new history will not contain any history events for this CI type.</p>	<p>progress of their upgrade. (debug mode)</p>
View Archive Upgrader	Copies view archive tables (VIEW_ARCHIVE and ARCHIVE_CHUNK) from the old history schema to the UCMDB schema. Archives are copied in bulks as they contain their TQL result.	✓	✓	<p><b>Upgrade:</b> The archive will not be copied to the new schema and will not be available in the upgraded system.</p> <p><b>Cleanup:</b> The archives will remain as garbage in the old schema.</p>	<ul style="list-style-type: none"> <li>● upgrade.short.log</li> <li>● upgrade.detailed.log</li> </ul> <p>Will contain information about the archive being copied and the copy progress per-table.</p>
Removed CIs History Upgrader	For every CI type that is audited in the new history, all the CIs that were removed in the time frame for saving removed CIs will be	✓	✓	<p><b>Upgrade:</b> If a CI type's removed data history was not upgraded, the removed data history table will not be created or</p>	<p>upgrade.detailed.log</p> <p>Logs the following:</p>

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
	<p>upgraded to the removed CIs tables (HDMR\HDMRL) of this CI type.</p> <p>The time frame for saving removed CIs is based on the <i>history.purging.months.to.save.back</i> and <i>history.purging.extra.months.to.save.back.re.moved.data</i> settings (default value: 1 month).</p> <p>The time frame is from the date of the upgrade minus the number of months the purging saves date to the same date minus the number of months to save removed CIs.</p> <p>For example, if <i>&lt;history.purging.months.to.save.back&gt;</i> is 3 months and <i>history.purging.extra.months.to.save.back.re.moved.data</i> is 2 months and the upgrade runs at 15.7.2012, the time frame will be from 15.2.2012 to 15.4.2012.</p>			<p>the removed data events from the old history will be missing in table. If the removed CIs history table was not created for a CI type, purging will not save removed CIs.</p> <p>When the upgrade is finished and the server is running, you can run the <b>alignHistoryForType</b> operation from the history JMX services to create the history tables for a specific CI type that the history upgrader failed to upgrade or for all the CI types.</p> <p>In this case, the removed CIs tables for all the CI types that the operation ran for will not contain any events from the old history.</p> <p><b>Cleanup:</b> Old history tables will not be deleted from the old history schema</p>	<ol style="list-style-type: none"> <li>1. Upgrade progress</li> <li>2. Removed CIs History table creation</li> <li>3. Each class number of removed CIs found in the time frame</li> </ol>
Authorization Upgrader	<p>Upgrades 9.0x roles and users, as follows:</p> <ol style="list-style-type: none"> <li>1. Upgrades roles: <ul style="list-style-type: none"> <li>○ Each 9.x role is converted to a 10.xx role (there is specific conversion for each internal permission)</li> <li>○ If 9.x role1 is associated with 9.x role2, the corresponding upgraded role1 in 10.xx will contain the permissions of both role1 and role2 (because in 10.xx there are no role associations).</li> <li>○ UCMDB Browser CI access permissions are added as follows: <ul style="list-style-type: none"> <li>▪ <b>View All CIs</b> permission is added by default to each role.</li> </ul> </li> </ul> </li> </ol>	✓	✓	<p><b>Upgrade:</b></p> <ul style="list-style-type: none"> <li>• If failed while upgrading the roles: some of the roles that existed in 9.x will not exist in 10.xx; no user will exist in 10.xx (except for the out-of-the-box users)</li> <li>• If failed while upgrading the users: some of the users that existed in 9.x will not exist in 10.xx; others may have only some of the permissions they had in 9.x</li> </ul> <p><b>Cleanup:</b></p> <ul style="list-style-type: none"> <li>• Old CI types may remain in the class model: user, integration_user,</li> </ul>	<p>security.authorization.management.log : Logs every modification to the authorization model, such as roles creation (with the exact permissions), user creation, user role assignments, and so on.</p> <p>Users and Roles moves resources to the URM and writes to these log files:</p> <ul style="list-style-type: none"> <li>• upgrade.detailed.log</li> <li>• urm.log</li> <li>• urm_detailed.log</li> <li>• urm_audit.log</li> <li>• urm_detailed_audit.log</li> </ul>

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
	<ul style="list-style-type: none"> <li>▪ <b>Edit All CIs</b> permission is only added to roles which have <b>Data Update</b> permission.</li> </ul> <p>2. Upgrades users:</p> <ul style="list-style-type: none"> <li>○ Upgrades each user's properties</li> <li>○ If the 9.x user is associated with the 'admin' role, or with a role that is associated with 'admin' – creates an association between the corresponding 10.xx user and the SuperAdmin role.</li> <li>○ In all other cases <ul style="list-style-type: none"> <li>▪ Creates an association between the 10.xx user and the 10.xx upgraded roles, according to the role associations it had in 9.x</li> <li>▪ If the 9.x user had permission assigned directly to the user, then a new role is created in 10.xx with these permissions, and an association is created between the 10.xx converted user and this new role. The new role is named ConvertedRole1/2/3.... This name may be used by several 10.xx users that had the same permissions in 9.x.</li> <li>▪ If a 9.x user had at least one discovery-related permission (whether directly assigned or via a role), then this user is granted the Discovery and Integrations Admin role in 10.xx.</li> </ul> </li> </ul> <p>3. Upgrades integration users:</p>			<p>acl_role, acl_attachment, together with the 9.x CIs representing the users and roles. This does not have any effect on the 10.xx security management.</p> <ul style="list-style-type: none"> <li>• The obsolete Basic_Security.zip package may remain in the Package Manager.</li> </ul>	

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
	<ul style="list-style-type: none"> <li>○ Upgrades each integration user's properties</li> <li>○ Assigns SuperAdmin role to each integration user</li> </ul>				
Authorization System Users Upgrader	<p>Upgrades system users (sysadmin, UISysadmin, and possibly others that were created by the user). The following steps are performed:</p> <ol style="list-style-type: none"> <li>1. Upgrades the properties of the system user.</li> <li>2. Marks the 10.xx converted user as "server administrator"</li> <li>3. Assigns the SuperAdmin/NonUISuperAdmin role to the system user. The NonUISuperAdmin role is a new role created in 10.xx, that is used for system users that did not have permission to access the UI in 9.x (like sysadmin).</li> </ol> <p><b>Note:</b> 9.x system users are converted to 10.xx "regular" users. There is no "system user" concept in 10.xx.</p>	✓	✓	<p><b>Upgrade:</b> System users will not be upgraded.</p> <p>sysadmin and UISysadmin will exist in 10.xx, because they are taken from the Basic_Authorization.zip package. But if the user created other system users in 9.x, they may not exist in 10.xx</p> <p><b>Cleanup:</b> The SYS_USERS table will remain in the database; old 9.x system users will remain. This will have no effect on 10.xx security management.</p>	<ul style="list-style-type: none"> <li>• Upgrade.log: Logs the upgrade steps</li> <li>• security.authorization.management.log: Logs every modification to the authorization model, such as role creation (with the exact permissions), user creation, user role assignments, and so on.</li> <li>• upgrade.detailed.log</li> <li>• urm.log</li> <li>• urm_detailed.log</li> <li>• urm_audit.log</li> <li>• urm_detailed_audit.log</li> </ul>
LDAP Group Mapping Upgrader	<p>Upgrades LDAP mappings. In 9.x each LDAP user was mapped to a role. In 10.xx, each user is mapped to a user group, and the user group is mapped to a role. The upgrader performs the following:</p> <ol style="list-style-type: none"> <li>1. Collects all 9.x roles assigned to LDAP users.</li> <li>2. For each 9.x role, creates a user group in 10.xx, and assigns it to the corresponding 10.xx role.</li> <li>3. Assigns each 10.xx user to a 10.xx user</li> </ol>	✓	✓	LDAP users will not belong to user groups, and thus will have no permissions	<ul style="list-style-type: none"> <li>• Upgrade.log: Logs the upgrade steps</li> <li>• security.authorization.management.log: Logs every modification to the authorization model, such as role creation (with the exact permissions), user creation, user role assignments, and so on.</li> </ul>

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
	group.				
LDAP Default Role Upgrader	Creates a new user group in UCMDB and assigns the previously upgraded role that was configured as the LDAP default role to this group.	x	✓	LDAP default role will be assigned when authenticating with an LDAP user that is part of an LDAP group that has no group mapping in the UCMDB.	upgrade.detailed.log
Offline Resource Loader	Loads the upgraded resources created in the “Offline Resource Retriever” step from the disk to the database.	✓	✓	<p><b>Upgrade:</b> The failed resources will not be included in the environment after the upgrade. However, factory resources will be included as they are deployed in a different step. Only the user resources are missing.</p> <p><b>Cleanup:</b> The CI types that represent the old resources in the Model will remain in the Class Model.</p>	<ul style="list-style-type: none"> <li>• upgrade.short.log</li> <li>• upgrade.detailed.log</li> </ul> <p>Contains the number of resources per type that are stored on disk, and warning and error messages if there is a problem with a resource.</p> <p><b>Note:</b> If you need to rerun this step you will get a warning for each resource that was already deployed the first time the step ran.</p>
UI Resources Upgrader	Converts the old UI recipients, LDAP mappings, user preferences, saved reports, and scheduled jobs to the new URM format.	✓	✓	The entities mentioned in the description are not migrated.	<ul style="list-style-type: none"> <li>• upgrade.detailed.log</li> <li>• error.log</li> </ul>
CIs Tenants Upgrader	<p>Handles CI tenant assignments in a multi-tenant environment as follows:</p> <ol style="list-style-type: none"> <li>1. Collects all tenant names from the <b>TenantOwner</b> and <b>TenantsUses</b> properties of all 9.x CIs.</li> <li>2. Creates tenants with these names</li> <li>3. For CIs that do not have these properties, sets the system default tenant as their tenant “<b>TenantOwner</b>” and “<b>TenantsUses</b>”</li> </ol>	✓	✓	<ul style="list-style-type: none"> <li>• Tenants that were defined on CIs in 9.x may not be created as “real tenants” in 10.xx</li> <li>• Some CIs may remain with no tenant owner – illegal state</li> </ul>	<ul style="list-style-type: none"> <li>• Upgrade log: Logs the upgrade steps</li> <li>• security.authorization.management.log: Logs tenant assignments to resources</li> <li>• upgrade.detailed.log</li> <li>• urm.log</li> <li>• urm_detailed.log</li> <li>• urm_audit.log</li> <li>• urm_detailed_audit.log</li> </ul> <p><b>Note:</b> CI tenant assignments are not logged</p>
Resources Tenants Upgrader	In a multi-tenant environment, sets default	✓	✓	Some resources may not have tenant	<ul style="list-style-type: none"> <li>• Upgrade log: Logs the upgrade steps</li> </ul>

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
	<p>tenant owner and consumers, as well as user owner, for all resources in 10.xx.</p> <ol style="list-style-type: none"> <li>For every 10.xx resource, sets its tenant owner and tenant consumer to be the system default tenant.</li> <li>For every 10.xx resource, sets its user owner to be admin user.</li> </ol>			owner or user owner – illegal state.	<ul style="list-style-type: none"> <li>security.authorization.management.log: Logs tenant assignments to resources</li> </ul>
Tenants Information History Upgrader	<p>For multi tenancy environments, for all the events that were upgraded, tenant owner/user will be entered:</p> <p>Tenant information (tenant owner and tenant user) was not tracked in version 9.x but is tracked in 10.xx.</p> <p>Since there is no information from 9.x to enter in the tenant owner/tenant user columns in the 10.xx model, the upgrader does the following:</p> <ul style="list-style-type: none"> <li>If there was any tenant information for the CIs that existed in the model, the tenant information is copied from the model.</li> <li>For CIs that were removed and do not exist in the model, the tenant owner and tenant user columns will be populated with the default tenant in the system.</li> </ul> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>Even if these attributes were set to being tracked by the history they will not be taken into account during the upgrade.</li> <li>For single tenancy system the upgrade does not do anything.</li> </ul>	✓	✓	Tenant information will be missing in the history events.	<p>upgrade.detailed.log:</p> <ul style="list-style-type: none"> <li>Logs upgrade progress</li> <li>Logs all the CIs that were not found in the model and enters the default tenant for them (debug mode).</li> </ul>

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
Federation URM Resources Upgrader	Upgrades the adapter, integration point and data push job resources to the URM and removes the Cmdb8xAdapter	✓	✗	<ul style="list-style-type: none"> <li>• <b>Upgrade:</b> Saved integration points, adapters, and jobs will not be available.</li> <li>• <b>Cleanup:</b> Unnecessary CI Types and CIs will still be kept in the CMDB.</li> </ul>	<ul style="list-style-type: none"> <li>• upgrade.detailed.log</li> <li>• urm.log</li> <li>• urm_detailed.log</li> <li>• urm_audit.log</li> <li>• urm_detailed_audit.log</li> </ul>
UndeployOldAdaptersUpgrader	Undeploys the old RMI and Changes adapters and updates the data push jobs that used these adapters as a source to use a dummy source. In addition, data push jobs that used the old data push engine and were configured to have the RMI adapter as the source (jobs that were saved as RMI/topology) are marked to run data push jobs with a full layout TQL.	✓	✓	The old saved data push integrations that were configured will not be available.	upgrade.detailed.log
Remove Unused Resources	Cleans old reconciliation resources from version 8.x	✓	✓	Errors might occur in log files when performing the export to the Content Pack.	upgrade.detailed.log
Basic Packages Deployer	Deploys packages related to the UCMDB infrastructure.	✓	✓	Varies, depending on what exact resource has failed	<ul style="list-style-type: none"> <li>• log\package_reports\customer_1\deployReports</li> <li>• Mam.packaging.log</li> <li>• error.log</li> <li>• cmdb.classmodel.log</li> </ul>
CP Deployer	Deploys the Content Pack provided with the new version	✓	✓	Varies very much on what exact resource has failed	<ul style="list-style-type: none"> <li>• log\package_reports\customer_1\deployReports</li> <li>• Mam.packaging.log</li> <li>• error.log</li> <li>• cmdb.classmodel.log</li> </ul>
CmdbAdapter Integration upgrader	Updates the CmdbAdapter integration and population jobs to suitable for the Cmdb9xAdapter.	✓	✗	The saved integrations and jobs that were defined for the CmdbAdapter will be lost.	upgrade.detailed.log

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
Integration Deactivation Upgrader	<p>From version 10.xx all push/federation integrations run on a Data Flow Probe. This upgrader disables all push-/federation-type integrations from 9.0x that do not have a Data Flow Probe defined.</p> <p><b>Note:</b> Population-type integrations from 9.0x are not disabled because they are defined to run on Data Flow Probes.</p>	✗	✓	Integrations will not be editable from the Integration Studio. You will be forced to deactivate them one by one.	upgrader.detailed.log
Consistency Upgrader	Fixes consistency issues in the database.	✓	✗	Database may contain inconsistent data (links with missing ends, IDs which exist only in the root table, or IDs which exist only in the CIT's data table)	cmdb.db.tool
OIDToHostClass upgrader	<p>Upgrades oidToHostClass user-defined rules to the new rule format and saves the new rules in the <b>UserDefinedRules</b> package with the name, <b>userDefinedOidToHostClassRules</b>.</p> <p>Removes the old <b>oidToHostClass</b> resource from the Network package.</p> <p>Notes:</p> <ul style="list-style-type: none"> <li>Part of the oidToHostClass upgrade is to recognize the user-defined rules and transform only them to the new discovery rules format (the out-of-the-box rules are not taken into consideration in this specific process). This recognition is done by comparing the oidToHostClass.xml content under CP.zip with the oidToHostClass resource deployed. The upgrader removes all of the out-of-the-box content from the deployed data, and works only on what is left. The 9.x CP.zip file is not available when</li> </ul>	✓	✓	<ul style="list-style-type: none"> <li><b>Upgrade:</b> User-defined oidToHostClass rules will not be changed to the new normalization rule format and will not be part of the discovery rule engine.</li> <li><b>Cleanup:</b> The old oidToHostClass resource will not be removed.</li> </ul>	<p>upgrade.detailed.log:</p> <p>Messages logged:</p> <ul style="list-style-type: none"> <li>“Start upgrading oidToHostClass.xml”</li> <li>“DONE upgrading oidToHostClass.xml”</li> <li>Errors that will be logged during the upgrade: <ul style="list-style-type: none"> <li>“Previous CP.zip was not found under &lt;defaultCpFile.getAbsolutePath&gt; using default file from CP10.”</li> <li>“oidToHostClass resource was not found in CP.zip. using default file from CP10.”</li> <li>“FAILED Upgrading oidToHostClass.xml”</li> </ul> </li> </ul>

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
	<p>doing the upgrade - since the upgrade process is done by installing UCMDB 10.xx from scratch without configuring a new schema and upgrading only the schema, and the CP.zip used for UCMDB 10.xx does not contain the oidToHostClass.xml file. To solve this issue: As part of the upgrade, the oidToHostClass.xml is taken from CP10.zip. If the 9.x server that is being upgraded used a newer Content Pack, you should put the Content Pack used under: &lt;server content dir&gt;\content_packs_9x. The upgrader first looks for the Content Pack in the folder specified above. If it is not there, it uses the saved file from Content Pack 10.00.</p> <ul style="list-style-type: none"> <li>The upgrader will add to each output attribute in the user defined rules the prefix "discovered_" (for example, discovered_vendor, discovered_model) This does not happen in the out-of-the-box rules since the values there are already normalized. If the user wants to have the rules insert values to real attributes and not "discovered" attributes – he should open the userDefinedOidToHostClass file, check that output values (that he defined on the 9.x server) are normalized and change the rule to use real output attributes (for example, change "discovered_vendor" to be "vendor")</li> </ul>				
SysObjectIdUpgrader	Adds a "." (period) to the beginning of the	✘	✔	The value of the <b>sys_object_id</b> attribute	upgrade.detailed.log

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
	<p><b>sys_object_id</b> attribute of the Node CIs.</p> <p>This is necessary for consistency with newly discovered <b>sys_object_id</b> attributes which start with a "." (period).</p> <p><b>Note:</b> If the <b>sys_object_id</b> attribute is empty or already starts with a "." (period), it is not changed.</p>			will not be consistent with the new discovered Nodes.	<ul style="list-style-type: none"> <li>• Reports upgrader start</li> <li>• Reports the number of CIs to upgrade</li> <li>• Reports upgrader finish</li> <li>• In case of failure, reports the cause of failure</li> </ul>
HistoryAutoCompletionUpgrader	Completes history changes data in the history tables, for changes made during other upgrade steps.	✘	✓	May cause low performance the first time a user gets history data for specific CIs.	<ul style="list-style-type: none"> <li>• upgrade.detailed.log: Upgrader progress data log</li> <li>• history.log: History operation log</li> </ul>
Deletes obsolete records from LOA table (reconciliation priorities)	In UCMDB 9.x the reconciliation priorities of some of the deleted CIs were left in persistence (database table "LOA"). This upgrader removes obsolete reconciliation priorities.	✓	✓	Obsolete reconciliation priorities may cause failures when data is populating UCMDB.	upgrader.detailed.log
Search Configuration Upgrader	<p>New search-engine configurations are saved in URM, while configurations for versions of UCMDB Browser prior to version 1.7 were defined through qualifiers in the class model.</p> <p>This upgrader upgrades the search configuration to support the new search engine.</p> <p><b>Actions taken:</b></p> <p>Any CI Type with qualifier CMS_BROWSER_SEARCH / MODELING_ENABLED is added to search-ranking-configuration with Ranking-Priority 2.</p> <p>Any attribute with qualifier CMS_SEARCHABLE_ATTRIBUTE is added to search-indexer-configuration and to search-</p>	✘	✓	Out-of-the-box search-configuration is used.	upgrade.detailed.log

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
	<p>ranking-configuration with Ranking-Priority 2.</p> <p><b>Notes:</b></p> <p>Runs only if the <b>Enable search</b> option was selected during upgrade (<b>cmdb.search.enabled</b> is set to <b>TRUE</b>).</p> <p>Class model does not change following this upgrader, the qualifier used for versions of the UCMDB Browser prior to 1.7 remains.</p>				
UCMDB Search Upgrader	<p>Indexes all relevant CI attributes to the new search engine, based on the upgraded configuration.</p> <p>Runs only if the <b>Enable search</b> option was selected during upgrade (<b>cmdb.search.enabled</b> is set to <b>TRUE</b>).</p> <p>Is performed in two phases:  (1) writing DB tables content to CSV files in the &lt;UCMDB Installation directory&gt;\search\index_csvs\customer&lt;customer-ID&gt;\ folder.  (2) indexing the CSV files content to the new search engine.</p> <p>May take time to complete: approximately 1 hour per 1million CIs.</p>	✘	✔	CIs will not be indexed to Solr (search will not work)	<p>search.log</p> <p>Progress is reported in logs only during CSV files writing and not during indexing.</p>
Disable Aging Upgrader	Sets the <b>model.aging.is.aging.enabled</b> settings to <b>false</b> for all the customers	✘	✘	The <b>model.aging.is.aging.enabled</b> setting will be <b>true</b> after the upgrade. This could cause the deletion of CIs if the server is not used after the upgrade.	
Upgrade IpAddress, 'IP Lease Time' attribute Upgrader	Sets the default value Permanent-or-Long (0) in <b>IP Lease Time</b> attribute for all existing IP Address CIs.	✔	✔	IP Address and Node identification will be inconsistent.  May cause duplications in the	<ul style="list-style-type: none"> <li>• upgrade.log</li> <li>• upgrade.detailed.log</li> <li>• error.log</li> </ul>

Upgrader	Description	Critical	Can be rerun	Implications of Failure	Log Files
				environment.	
CleanupRemoveColumnsUpgrader	Removes the old and obsolete tables and columns from the CMDB.	✓	✗	Garbage will be left in the database and the system will not be cleaned properly.	upgrade.detailed.log
Run database statistics Upgrader	Runs database statistics to optimize performance	✗	✓	Database statistics will not be run; might have performance implications	upgrade.detailed.log

# Troubleshooting and Warning Logs for Resource Upgrade

The following issues can occur during resource upgrade. Be aware that none of these issues cause the upgrade to fail but might impact which resources are upgraded.

- If a TQL query was created in a previous version with an invalid layout, it is inserted during upgrade and does not fail the upgrade; rather, a warning is added to the log. These TQL queries could have been created by copy/pasting a valid TQL query and their xml, then manually edited. You may see the following error message in the error logs:

**Pattern: <pattern name> has invalid layout: <stack trace>.**

- You cannot define a grouping on a contact node of a perspective. However, there was no validation to prevent such a grouping prior to version 10.00. A view created in this way does not work correctly and is removed during the upgrade process. You may see the following error message in the `cmdb.upgrade.short` and `cmdb.upgrade.detailed` logs:

**You cannot define grouping on a contact query node.**

**Failed to add CMDB\_VIEW <view name> You cannot define grouping on a contact query node.**

**This would appear in `cmdb.upgrade.short` and `cmdb.upgrade.detailed` logs.**

- You cannot define a TQL query, view, impact analysis rule, or enrichment with an empty space in its suffix. However, there was no validation to prevent creating such TQL queries prior to version 10.00. A TQL query that contains spaces in its suffix is not upgraded, an error appears in the log, and you must define the TQL query again using a valid name. You can perform this step prior to upgrade. If these names are not fixed prior to upgrade, you may see the following error message in the `cmdb.upgrade.short` and `cmdb.upgrade.detailed` logs:

**Resource name is not valid.**

**Failed to add <resource type> <resource name> Resource name is not valid.**

- In multi-tenancy implementations, you cannot define TQL queries with an @ character because the @ character is used internally in MT environments. If you are upgrading a single tenant environment to a multi-tenant environment and you have TQL queries with the @ character, these TQL queries are not upgraded and are lost and a warning is added to the log. You will have to define these TQL queries again using a new name. You can perform this step prior to upgrade. If these names are not fixed prior to upgrade, you may see the following error message in the `cmdb.upgrade.short` and `cmdb.upgrade.detailed` logs:

**Resource <resource name> contains the @ character, which is forbidden for multi tenant**

**environment. The resource will be removed from the upgraded system. Please create it again with different name.**

- If you have any folders that contain both a resource and a subfolder with the same name, the resource does not upgrade and a warning is added to the log. It is recommended to rename the sub folder or save the resource with different name prior to performing the upgrade. You may see the following message in the error logs:

**WARN: Failed to add <resource type> <resource name> duplicate subfolder or resource [<resource name>] in [<folder name>]**

- If you use a custom **key.bin** to encrypt the credentials and you did not copy it to **C:\hp\UCMDB\UCMDBServer\conf\discovery\** BEFORE starting the upgrade process, ranges and credentials will be lost. You may see the following message in the error logs:

**FATAL - Failed to decrypt the Domains Scope Document finished.**

**Solution:** Ensure that the key.bin file that you backed up prior to starting the upgrade has been copied to **C:\hp\UCMDB\UCMDBServer\conf\discovery\**.

- If a legacy adapter contains an illegal character inside a job template name, the Data Flow Probes Setup window will fail to start after performing the upgrade. You may see the following messages in the error logs:

**Template contains characters not supported in XML in the name of job "<job Id>".{<job template name>}}**

**<Adapter name>-Adapter Template contains characters not supported in XML in the name of job <job name>**

**Solution:** Manually remove the adapter with the illegal character.

# Send Documentation Feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

**Feedback on Upgrader Reference (Universal CMDB 10.20)**

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to [cms-doc@hp.com](mailto:cms-doc@hp.com).

We appreciate your feedback!