

Agile Product Lifecycle Management

CAD for Cloud Integration Guide

Release 9.3.6.18

E99983-10

August 2023

Copyright © 2010, 2023, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	vii
Audience.....	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions.....	vii
 1 Introducing CAD for Cloud Integration	
CAD for Cloud User Roles and Integration Process	1-1
CAD for Cloud Development Sequence.....	1-1
 2 Installing and Configuring CAD for Cloud Integration	
Installation Procedures	2-1
Installing Agile PLM	2-1
Agile PLM 9.3.6.13 with Fusion Middle Ware 12.2.1.4	2-1
Use Database 12c R2 for Clustered Environments.....	2-2
Installing AutoVue for Agile PLM	2-2
Installing Oracle Web Services Manager.....	2-2
Configuration Procedures	2-2
Integration Security Considerations.....	2-3
Product Development Cloud Service Policy.....	2-4
Configuring CAD for Cloud Client Policy	2-4
Using the Web Services Security Configurator Tool	2-4
Prerequisites.....	2-4
Configuring oracle/wss_saml_token_bearer_over_ssl_client_policy	2-4
A. Attach Client Policy and Generate Key-pair using an Alias	2-5
B. Generate and Export Certificate Signing Request	2-5
C. Request Signature for the Certificate.....	2-6
D. Import Certifying Authority-signed Certificate into Agile PLM Key-store.....	2-6
E. Configure SAML Issuer Name to Enhance Security	2-6
F. Obtain Distinguished Name from the CA-Signed Certificate.....	2-7
G. Import Certificate Chain and Configure SAML Issuer and DN List in Oracle Applications Cloud	2-7
H. Validate Secure Web Services Connection	2-8
Disabling CAD for Cloud Client Policy	2-8

Cluster Support in CAD for Cloud Using JMS and MDB.....	2-9
Configuring Target C4CJMSServer and C4CStore in config.xml	2-9
Agile PLM Configurations.....	2-10
Define Reference Objects in CAD for Cloud	2-10
To configure the Product Development external application:	2-10
To enable reference objects classes in Java Client:	2-11
To create reference object subclass:	2-12
Define Reference Object Mappings for Items and Changes.....	2-13
To define attribute mappings:	2-13
Define Agile PLM Roles and Privileges.....	2-14
Map PLM Design/Change TB / P2 Attributes to PD Item EFF / Change DFF Attributes.....	2-16
Template of C4C_Mapping.xml Configuration File	2-16
Loading, Parsing, and Validating C4C_Mapping.xml File	2-20
Publish the Mapped Design TB / P2 Attributes to PD Item EFF Attributes	2-21
Configuring Rule-Generated Item Numbers.....	2-21
Adding the Publish Workspace Tab in Java Client	2-23
WebLogic Server Configurations	2-23
Configuring WebLogic Server.....	2-24
Setting Up the CloudIntegration.properties File.....	2-24
Setting Up Attachment Categories in Root Item Classes	2-28
Setting Up Attachment Categories in CAD for Cloud	2-30
Creating Attachment Category Association	2-32
CAD File Types Supported in CAD for Cloud.....	2-34
Administrator can configure a Design Page Two non-version-specific List Attribute to map to a Cloud PD Item Attachment Category 2-35	
Registering with Agile PLM	2-35
Generating PD Cloud WS Java Client Stub Jars.....	2-36
Exporting the Certificate from the Browser:	2-36
Certificates to the JDK Trust Store:.....	2-36
Generating the Change Service Jar:.....	2-37
Generating the Item Service Jar:.....	2-37
Generating the Relationship Service Jar:	2-37
Patching and Redeploying the .ear File:	2-38
Logging Request and Response Payloads	2-38
Manage Target System in Fusion.....	2-38

3 Using the CAD for Cloud Integration

Introducing Publish Workspace	3-1
Using the Publish Workspace	3-1
Relating and Publishing Designs in Agile PLM to Items in PD Cloud	3-3
Business Rules for Relating a PLM Design with a PD Item	3-3
Manual relate by search	3-5
Auto-relating a Design Drawing during Publish	3-7
Auto-create by Publish	3-7
Rule-Generated Item Numbers	3-9
Setting and Propagating Primary UOM in PD Items.....	3-9
Setting Design Revision to Push to Item Revision	3-10

Unrelating a Design in Agile PLM from an Item in PD Cloud	3-11
Relating a DFCO in Agile PLM to an ECO in PD Cloud.....	3-11
Unrelating a DFCO in Agile PLM to an ECO in PD Cloud	3-12
Validating the Publish Readiness of a Design	3-12
Publishing a Design or Design Structure in Agile PLM to PD Cloud.....	3-13
Publishing while some Designs are in Process	3-14
Publishing CAD Design Structures	3-14
Selective Publish with DESIGN_ATTRIBUTE_FOR_BOM_PUBLISH_OPTION.....	3-14
Selective CAD File Types with PUBLISH_FILETYPES	3-15
Administrator with Admin Privilege can cancel a C4C Publish Job	3-15
Using Save As to Create a New DFCO	3-16
C4C Mapping Sample File	A-1

Preface

Agile PLM is a comprehensive enterprise PLM solution for managing your product value chain.

Audience

This document is intended for administrators and users of the Agile PLM products.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

Oracle's Agile PLM documentation set includes Adobe® Acrobat PDF files. The Oracle Technology Network (OTN) Web site <http://www.oracle.com/technetwork/documentation/agile-085940.html> contains the latest versions of the Agile PLM PDF files. You can view or download these manuals from the Web site, or you can ask your Agile administrator if there is an Agile PLM Documentation folder available on your network from which you can access the Agile PLM documentation (PDF) files.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Introducing CAD for Cloud Integration

CAD for Cloud lets CAD users develop designs; create Design business objects and Design Structures in Agile Product Lifecycle Management (Agile PLM); and publish the designs to Items and Item Structures in Oracle Product Development Cloud (PD Cloud). The integration of Agile PLM with PD Cloud:

- Gives PLM designs a Change process that is distinct from the Change process for PD Cloud items;

- Permits stricter control over users who can check out and modify designs;

- Incorporates designs so that the design is indicated as complete.

CAD for Cloud User Roles and Integration Process

Users in the CAD for Cloud process are granted generally one of these three roles:

Administrator - uses Agile Administrator to configure policy and environment in Agile PLM for the CAD Engineer and Engineering Coordinator. The administrator configures policy and environment in PD Cloud for Engineering Coordinators and downstream personnel.

Engineering Coordinator - in Agile PLM, creates and manages changes to design files using Design File Change Orders (DFCOs). In PD Cloud, the Engineering Coordinator creates and manages changes to engineering items and item structures using Engineering Change Orders (ECOs) and other change orders.

CAD Engineer - receives item numbers from the Engineering Coordinator; creates CAD data and saves it to Agile PLM as designs; and relates designs in Agile PLM to engineering items in PD Cloud.

CAD for Cloud Development Sequence

Once Agile PLM and PD Cloud have been configured for the CAD for Cloud integration, and tested, the day-to-day work by engineering coordinators and CAD engineers can commence. A possible 9-stage development sequence follows. It suggests the activities and transitions that typical users will be involved in. The configured user roles, activities, and development sequence may vary according to your organization's needs.

Note: In the CAD for Cloud integration, Agile PLM design files are converted to Product Development Cloud engineering items. In the development sequence, the Administrator and Engineering Coordinators may need to switch from Agile PLM to PD Cloud environments often.

In this guide, these are the business objects and their common names on the PLM and PD sides of the integration:

Agile PLM side: **Design files** (designs)

Product Development Cloud side: **Engineering items** (items)

Both sides have their own change orders:

Agile PLM side: **Design file change order** (DFCO)

Product Development Cloud side: **Engineering change order** (ECO)

This is a possible sequence for a development process with CAD for Cloud:

1. In a CAD application, the **CAD Engineer** creates CAD data - which may be designs or other specifications - and saves to Agile Designs via MCAD integration.
2. In PD Cloud, the **Engineering Coordinator** creates Items and Engineering Change Orders (ECO). In Agile PLM, they create Design File Change Orders (DFCOs).
3. Engineering Coordinator sends PD Cloud Item numbers, PD ECO number, and Agile DFCO number to CAD Engineers.
4. CAD Engineer assigns Agile Designs as Affected Files in the DFCO. When the engineer assigns the Change to the modified files via the MCAD integration, it is automatically added to the Affected Files tab in the DFCO.
5. CAD Engineer then relates the DFCO to the PD ECO, and relates the designs to their respective PD Item numbers in the Publish Workspace.
6. CAD Engineer updates CAD files and incorporates designs when complete.
7. CAD Engineer publishes the BOM structure for the Items to PD from the DFCO.
8. Engineering Coordinator reviews DFCO and PD ECO/Item BOM and releases DFCO.
9. Engineering Coordinator completes PD ECO work and releases ECO.

Chapter 2 is addressed to the Agile administrator. It details how to install and configure the CAD for Cloud integration. The installation tasks and configuration procedures are addressed to the CAD for Cloud integration administrator. However, in developing the integration, engineering coordinators may refer to this chapter and provide advisory specifications to their company's integration administrator.

Chapter 3 describes how to use the CAD for Cloud functionality in Agile PLM. This chapter is addressed to the PLM administrator, as well as end-users, such as engineering coordinators, CAD specialists, PLM users, and PD Cloud users.

Installing and Configuring CAD for Cloud Integration

This chapter details installation procedures and configuration tasks to integrate CAD-generated design files through Agile PLM design (business objects) and design structures (BOMs) to Product Development Cloud items and structures.

This chapter is addressed primarily to an Agile PLM administrator configuring the CAD for Cloud integration with PLM's Java Client Administrator; you may need to refer to *Agile Product Lifecycle Management Administrator Guide*, **Rel. 9.3.6**.

The same administrator or other person may be tasked with configuration of Product Development Cloud to receive PLM designs, that is, configure PD Cloud for the CAD to Cloud integration. You may need to refer to *Oracle SCM Cloud Implementing Product Management*, **Ver.19.01**, and *Oracle SCM Cloud Using Product Development*, **Ver.19.01**.

Installation Procedures

The integration of CAD design files from Agile PLM to PD Cloud requires that you complete the following installation processes:

- [Installing Agile PLM](#);
- [Installing AutoVue for Agile PLM](#);
- [Installing Oracle Web Services Manager](#).

Installing Agile PLM

Refer to the *Agile PLM Application Installation Guide Rel. 9.3.6* for instructions to complete the base installation of Agile PLM 9.3.6.

This document, *Agile PLM CAD for Cloud Integration Guide*, refers to **Agile PLM upgrade Release 9.3.6.9**.

During the installation, write down and keep track of these:

- Repository Creation Username (RCU) DB URL; and
- RCU username and password that you have chosen.

Agile PLM 9.3.6.13 with Fusion Middle Ware 12.2.1.4

Fusion Middle Ware Agile PLM Release 9.3.6.13 is certified with (FMW) **12.2.1.4**.

Customers can perform a fresh installation of Agile PLM 9.3.6 directly on FMW 12.2.1.4 using an installer patch (see Patch Number 27871474 for details).

For more details, search for Support Note 2383539.1 on My Oracle Support:
<https://support.oracle.com/>.

If you choose to continue using FMW 12.2.1.1 and you encounter an issue, you may be required to upgrade to FMW 12.2.1.4 as part of the SR action plan with Oracle Support.

Customers using FMW 12.2.1.1 for Agile PLM 9.3.6 must upgrade to FMW 12.2.1.4, or make plans to do so as soon as possible.

Notes: FMW 12.2.1.4 may be a prerequisite for future Agile PLM patches and/or Release Update Packs.

Patches for FMW 12.2.1.1 are no longer being provided.

Use Database 12c R2 for Clustered Environments

In Oracle Database 12c R1, the maximum length of identifiers is 30 bytes. Using this database in a clustered environment will likely cause a failure while creating the newly introduced tables for persistent Java Message Service (JMS) store that is required for CAD for Cloud Publish in clustered environments.

To avoid this identifier length limitation, install Oracle Database 12c R2, in which the maximum length of identifiers is increased to 128 bytes for table name identifiers.

Installing AutoVue for Agile PLM

To install AutoVue for Agile PLM, refer to *Oracle AutoVue for Agile PLM Installation Guide* for complete instructions.

Installing Oracle Web Services Manager

Install Oracle Web Services Manager (OWSM) using the `configureWSSecurity.cmd` file.

Complete the following steps to install OWSM to apply to CAD for Cloud integration. Ensure that you have the Repository Creation Utility (RCU) DB URL and RCU username/password.

1. Shut down Agile PLM, if it is running.
2. Open the command prompt and navigate to the `AGILE_HOME\Install\bin` directory.
3. Run the following command:
`configureWSSecurity.cmd <RCU_DB_URL> <RCU_MDS_USER> <RCU_MDS_USER_PASSWORD>.`

You must add `_MDS` to the user name. For example:

`configureWSSecurity.cmd jdbc:oracle:thin:@<domain>:1521/<SID> <RCU_User>_MDS <RCU_PWD>`

4. After the command runs successfully, restart Agile PLM.

Configuration Procedures

The version of Oracle Product Development Cloud that you should configure your CAD for Cloud integration to is **Version 19.01**.

The CAD-for-Cloud integration between Agile PLM and Product Development Cloud requires that you complete the following procedures in this sequence. Before going to the next procedure, verify that each previous task has been successfully completed.

[Integration Security Considerations](#)

[Product Development Cloud Service Policy](#)
[Configuring CAD for Cloud Client Policy](#)
[Disabling CAD for Cloud Client Policy](#)
[Cluster Support in CAD for Cloud Using JMS and MDB](#)
[Configuring Target C4CJMSServer and C4CStore in config.xml](#)
[Agile PLM Configurations](#)
[Adding the Publish Workspace Tab in Java Client](#)
[WebLogic Server Configurations](#)
[Setting Up the CloudIntegration.properties File](#)
[Configuring Rule-Generated Item Numbers](#)
[Registering with Agile PLM](#)
[Generating PD Cloud WS Java Client Stub Jars](#)
[Logging Request and Response Payloads](#)

Integration Security Considerations

Oracle Agile PLM's CAD for Cloud integration uses web services in Product Development and Product Hub applications to look up and update PD items, structures (BOMs), and changes. In this integration, Oracle Agile PLM acts as a web-services client to PD Cloud, and it must be installed in a deployment configuration such that:

1. Agile PLM is used to connect and access PD Cloud securely using Web Services Security (WSS) policies in OWSM.

`oracle/wss_username_token_over_ssl_client_policy`

Add "WSS.ClientPolicy.CADforCloud=oracle/wss_username_token_over_ssl_client_policy" to the end of `<agilehome>/agileDomain/config/agile.properties`, and then restart application server to make the change active.

`oracle/wss_saml_token_bearer_over_ssl_client_policy`

2. Users (including Supplier users) can navigate between Agile PLM and PD Cloud using Single Sign-On (SSO).

Given below are the deployment considerations to enable secure integration between on-premise Agile PLM and PD Cloud.

1. Set up Agile PLM with Secure Sockets Layer (SSL) and WSS enabled. Refer to *Agile Product Lifecycle Management Security Guide*, section 4.3.3, "Agile Setup with SSL and WSS Enabled".

Note: Oracle strongly recommends provisioning VPN between Agile PLM and PD Cloud, if Agile PLM is installed within the firewall.

2. Configure Security Assertion Markup Language (SAML)-based WSS using OWSM to propagate user identity. Refer to "[Configuring oracle/wss_saml_token_bearer_over_ssl_client_policy](#)".
3. Set up SSO using appropriate Identity Management system that Oracle Cloud Applications support.

Note: Oracle recommends using Oracle Identity Cloud Service (<https://docs.oracle.com/en/cloud/paas/identity-cloud/index.html>) with WebLogic Server using SAML 2.0. Or, customize SSO solution with Oracle Access Manager (OAM), using federated or synchronized LDAP.

Product Development Cloud Service Policy

The default Service Policy in Product Development Cloud is a group policy that supports multiple client policies, including the default Client Policy offered by Oracle:

`oracle/wss_saml_token_bearer_over_ssl_client_policy`

You are not required to attach this default Service Policy separately.

Configuring CAD for Cloud Client Policy

You must configure the required Client Policy to enable communication between CAD for Cloud and Oracle Applications Cloud service (such as Product Development Cloud).

This section describes the available Client Policy and the tasks that are required to configure it. The Client Policy is:

`oracle/wss_saml_token_bearer_over_ssl_client_policy`

Using the Web Services Security Configurator Tool

On the Agile PLM side of the integration, use the WSS Configurator tool to do the following for CAD for Cloud:

1. Enable or disable the Client Policy for CAD for Cloud.
2. Import or export certificates.

Prerequisites

Before you run the WSS Configurator tool, ensure the following:

1. OWSM is installed in the Agile domain as described in "[Installing Oracle Web Services Manager](#)" on page 2-2.
2. The attribute `CLOUD_CAD_INTEGRATION_ENABLED` in `agileDomain/config/CloudIntegration.properties` is set to `TRUE`.
3. Unzip `wssconfigurator.zip` (in `agileDomain/tools`) into the same location `agileDomain/tools/wssconfigurator`.
4. Grant **Execute** privilege on shell script `wssconfigurator.sh` in folder `wssconfigurator`.

Configuring `oracle/wss_saml_token_bearer_over_ssl_client_policy`

To configure `oracle/wss_saml_token_bearer_over_ssl_client_policy`, first ensure that SSL security is enabled: this is done on the PD Cloud side of the integration.

Once that is confirmed, perform tasks A.-H. in sequence. Before going to the next task, verify that each previous task is successfully completed.

[A. Attach Client Policy and Generate Key-pair using an Alias](#)

[B. Generate and Export Certificate Signing Request](#)

[C. Request Signature for the Certificate](#)

[D. Import Certifying Authority-signed Certificate into Agile PLM Key-store](#)

[E. Configure SAML Issuer Name to Enhance Security](#)

[F. Obtain Distinguished Name from the CA-Signed Certificate](#)

[G. Import Certificate Chain and Configure SAML Issuer and DN List in Oracle Applications Cloud](#)

[H. Validate Secure Web Services Connection](#)

A. Attach Client Policy and Generate Key-pair using an Alias

Run the WSS Configurator tool to attach the Client Policy and generate a key-pair. The certificate generated here - **CADforCloudSAMLSignKey** - is displayed in the key-store under stripe OWSM in the WebLogic console.

It is also exported to the folder where the WSS Configurator tool is unzipped.

By default, it is issued by "Demo CA" ('Demo' Certifying Authority).

1. Run WSS Configurator tool under agileDomain/tools/wssconfigurator/wssconfigurator.cmd for Windows and wssconfigurator.sh for Unix.
2. Select option 1 to enable Web Services Security.
3. Enter server URL, username, and password to connect to the WebLogic Server.
4. Select option 3 from the CAD for Cloud policy.
5. From the 3 options, select option 2 in the CAD-for-Cloud Client Policy: oracle/wss_saml_token_bearer_over_ssl_client_policy.
6. Enter the subject name and key-bit size, and press **Enter**.
 - a. Subject name should be as follows: CN=MyCommonname, OU=OracleUnit, O=MyOrganization, L=MyCity, ST=MyState, C=MyCountry

For example, CN=slc09pnr.us.oracle.com, OU=OracleUnit, O=Oracle, L=Redwood City, ST=California, C=US
 - b. Oracle recommends using **2048** for key-bit size for the certificate.
 - c. A certificate with an alias - **CADforCloudSAMLSignKey** - is generated and exported to the folder where the WSS Configurator tool is unzipped, usually in agileDomain/tools/wssconfigurator.
 - d. The client policy is configured successfully. Two entries are added to agile.properties in agileDomain/config/ (near the end of the file).

```
WSS.ClientPolicy.CADforCloud=oracle/wss_saml_token_bearer_over_ssl_client_policy
WSS.CADforCloudSAMLSignKey.alias=CADforCloudSAMLSignKey
```

B. Generate and Export Certificate Signing Request

Run WSS Configurator tool to generate and export a Certificate Signing Request (CSR) based on the certificate that was generated in task A.

1. Launch WSS Configurator tool on Agile Server.
2. Select option 4 to manage OPSS OWSM key-store, and press **Enter**.
3. Enter server URL, username, and password to connect to WebLogic Server.
4. For the second prompt, select option 3 to export the certificate, and press **Enter**.
5. For the third prompt, select option 2 (Certificate Signing Request).

- a. Fill in the alias to be exported. Enter CADforCloudSAMLSignKey, which was generated in task A.
- b. For the CSR to be exported, fill in the absolute path and the CSR file name (and extension). For example: C:\Certificate\C4CSAMLSignKey.csr.

C. Request Signature for the Certificate

To request a signature for the certificate:

1. Send the generated CSR to your Certifying Authority (CA) to request a CA-signed certificate.

The generated certificates:

1. Root Certifying Authority certificate
2. Intermediate Certifying Authority certificate and the CA-signed certificate are then used by the Cloud Operations team, who must import the chain of certificates into the OWSM key-store of Product Development.

D. Import Certifying Authority-signed Certificate into Agile PLM Key-store

By default, the certificate (the alias is CADforCloudSAMLSignKey) generated earlier (in task A) is issued by "Demo CA". Import the CA-signed certificate as a certificate with the same alias - CADforCloudSAMLSignKey. Then its certificate type changes from "Demo CA-signed Certificate" to "Public CA-signed Certificate".

First, copy the CA-signed certificate (that was returned from the certifying authority) to the local file system of Agile Server. Run WSS Configurator tool to import the CA-signed certificate, as in this procedure.

To import the CA-signed certificate:

1. Launch WSS Configurator tool on Agile Server.
2. Select option 4 to manage OPSS OWSM key-store, and press **Enter**.
3. Enter the server URL, username and password to connect to WebLogic Server.
4. For the second prompt, select option 2: **Import Certificate**.
5. For the third prompt, select option 1: **Certificate**.
 - a. Fill in the alias to be imported. Enter CADforCloudSAMLSignKey which was generated earlier, and press **Enter**.
 - b. Fill in the absolute path and the CA signed certificate to be imported.
For example: /opt/Certificate/slc07die.crt
6. Enter option 6: **Exit**.

E. Configure SAML Issuer Name to Enhance Security

For security reasons, you must set up a separate SAML issuer to overwrite the default value www.oracle.com. SAML issuer name can be any meaningful string, typically the host name of the client instance or company-specific information.

To configure the SAML Issuer Name:

1. Go to agileDomain/config/ and open the CloudIntegration.properties file.
2. Edit the attribute **SAML_ISSUER_NAME** to overwrite the default value.
3. Restart the Agile PLM Server.

Note: SAML_ISSUER_NAME cannot be left empty. The value is case-sensitive.

F. Obtain Distinguished Name from the CA-Signed Certificate

You must obtain the Distinguished Name (DN) from the Certifying Authority (CA)-signed certificate using the JDK tool. Provide the DN in the service request for the Cloud Operations team, who will add it to the configured SAML issuer.

To obtain the Distinguished Name:

1. Copy the CA-signed certificate to a local drive.
2. Open a command line and run the following command:

`keytool -printcert -file <full name with certificate path>`

Example: `keytool -printcert -file /opt/certs/slc07die.crt`

Note: if "keytool" cannot be recognized, go to \$JAVA_HOME/bin to run it again.

3. Copy and save the **Owner** value.

Example:

`keytool -printcert -file /opt/certs/slc07die.crt`

Owner: CN=slc07die.us.oracle.com, OU=Oracle PLMA FOR TESTING PURPOSES ONLY, O=Oracle Corporation, L=Redwood City, ST=California, C=US

Issuer: CN=DigiCert SHA2 Secure Server CA, O=DigiCert Inc, C=US

Here the DN for the certificate is CN=slc07die.us.oracle.com, OU=Oracle PLMA FOR TESTING PURPOSES ONLY, O=Oracle Corporation, L=Redwood City, ST=California, C=US.

G. Import Certificate Chain and Configure SAML Issuer and DN List in Oracle

Applications Cloud Based on the details provided in your service request, using WebLogic Scripting Tool (WLST), the Cloud Operations team performs these tasks:

Import the signed certificates as trusted certificates into the OWSM key-store of Fusion Applications Cloud

Configure the SAML issuer along with the DN list.

Raise a service request and include the following:

1. Signed certificates.
2. SAML issuer name and DN list.
3. Point of Delivery (POD) details.
4. My Oracle Support note IDs:
 - a. **Note 2404754.1** for certificates.
 - b. **Note 2427496.1** for configuring SAML issuer and DN List

Signed certificates to be attached contains Root CA, Intermediate CA, and CA-signed certificate you received earlier. (See [C. Request Signature for the Certificate.](#)) SAML issuer name is the value you configured in CloudIntegration.properties file.

Tip: If there are multiple Distinguished Names (DN), separate each one with a semicolon (;).

Be sure to provide the complete DN list. If the SAML issuer name and DN was added previously, WLST will overwrite the DN list of the SAML issuer.

For example, if SAML issuer CADforCloud already has "J" as DN and you want to add "K" as the DN, provide "J;K" as DN list, not just "K".

Note: Oracle recommends that you enter the DN list in a single line within the file, then upload it to the SR. This is to avoid any data loss during copying and pasting.

H. Validate Secure Web Services Connection

Test to make sure web-services calls are working without any failures, by executing the CAD for Cloud functions in Agile PLM.

If the access denied ("oracle.wsm.security.WSIdentityPermission" "resource=AgilePLM" "assert") exception shows while testing the CAD for Cloud functions with this client policy, follow these steps to fix it in Agile PLM.

1. Set WLST_EXT_CLASSPATH
 - a. Example for Windows: `c:/> set WLST_EXT_CLASSPATH=<AgileHome>\agileDomain\lib\wlsauth.jar`
 - b. Example for Unix: `$ export WLST_EXT_CLASSPATH=<AgileHome>/agileDomain/lib/wlsauth.jar`
2. Open the WebLogic Scripting Tool, go to `<weblogic_home>\oracle_common\common\bin\`, run `wlst.cmd` for windows and `wlst.sh` for Unix.
3. Connect to the WebLogic domain using the following command: `wls:/offline> connect (username, password, url)`. Enter appropriate values for the *username*, *password* and *URL* fields.

username: user to log in to WebLogic Server

password: password of the user

url: weblogic server url, the format is `t3(s)://<server>:<port>`.

4. Execute the following two commands to add permission:

```
grantPermission(permClass="oracle.wsm.security.WSIdentityPermission",permTarget="resource=AgilePLM", permActions="assert")

grantPermission(codeBaseUrl="file:${common.components.home}/modules/oracle.wsm.common_${jrf.version}/wsm-agent-core.jar",
permClass="oracle.wsm.security.WSIdentityPermission",
permTarget="resource=AgilePLM", permActions="assert")
```

Disabling CAD for Cloud Client Policy

You can disable CAD for Cloud Client Policy after enabling it.

To disable CAD for Cloud Policy:

1. Run WSS Configurator tool on Agile Server.
2. Select option 2 to **Disable** web-services security.

3. Enter the server URL, username and password to connect WebLogic Server.
4. Choose option 3 to **Disable** client policy in CAD for Cloud.
5. Enter 'y' or 'Y' to confirm.

The client policy is disabled successfully. Entry or entries added to agile.properties in agileDomain\config\ at the end will be removed.

Cluster Support in CAD for Cloud Using JMS and MDB

Agile PLM takes advantage of the clustering capability provided by the application server. See Chapter 5 of the *Agile PLM 9.3.6 Installation Guide*. CAD for Cloud is fully supported in this deployment configuration.

Java Message Service (JMS) provides a reliable, asynchronous, and loosely coupled communication between the nodes in the cluster. The Message-Driven Bean (MDB) processes the message asynchronously. The messaging load is distributed across the available nodes in the cluster and the traffic is re-directed to other available nodes.

CAD for Cloud's Publish-to-PD-Cloud feature makes use of JMS and MDB to ensure scalability and automatic fail-over protection in the event of a node failure.

Configuring Target C4CJMSServer and C4CStore in config.xml

To configure the target C4CJMSServer and C4CStore, do the following:

For Clustered Environments:

1. Get the Agile cluster name \$ClusterName from the <cluster> tag in file \$AGILE_HOME/agileDomain/config/config.xml (Note that the installer default value is 'AgileCluster'):

```
<cluster>
  <name>$ClusterName</name>
  ...
</cluster>
```

2. Fill the \$ClusterName in the two empty <target> tags in <jms-server> and <jdbc-store> located in file \$AGILE_HOME/agileDomain/config/config.xml

```
<jms-server>
  <name>C4cJMSServer</name>
  <target>$ClusterName</target>
  <persistent-store>C4cStore</persistent-store>
</jms-server>

<jdbc-store>
  <name>C4cStore</name>
  <prefix-name>C4C_</prefix-name>
  <data-source>AgileHaDataSource</data-source>
  <distribution-policy>Distributed</distribution-policy>
  <target>$ClusterName</target>
  <migration-policy>Always</migration-policy>
</jdbc-store>
```

For stand-alone environments:

1. Get the Agile server name \$StandaloneServerName from within the <server> tag in file \$AGILE_HOME/agileDomain/config/config.xml:

```
<domain>
  ...
```

```
<server>
  <name>${StandaloneServerName} </name>
</server>
...
</domain>
```

2. Fill `StandaloneServerName` in the two empty `<target>` tags in `<jms-server>` and `<jdbc-store>` located in file `$AGILE_HOME/agileDomain/config/config.xml` as follows:

```
<jms-server>
  <name>C4cJMSServer</name>
  <target>${StandaloneServerName}</target>
  <persistent-store>C4cStore</persistent-store>
</jms-server>

<jdbc-store>
  <name>C4cStore</name>
  <prefix-name>C4C_</prefix-name>
  <data-source>AgileHaDataSource</data-source>
  <target>${StandaloneServerName}</target>
</jdbc-store>
```

Agile PLM Configurations

To set up the Agile PLM environment for integration with Product Development, you must configure the following in Agile PLM Administrator:

Reference Objects:

- Define the Product Development application as a reference object and enable it
- Create subclasses for the reference object
- Define reference object mappings for items and changes

Roles and Privileges:

- Create a role for the CAD integration user.
- Associate privileges to this role to limit user access to the following business objects only: designs, DFCOs, and reference objects. See ["Define Agile PLM Roles and Privileges"](#) for more information.

Define Reference Objects in CAD for Cloud

The Reference Objects feature allows you to search for objects in external applications and add them as references. In Agile Administrator, once you enable the Cloud application from the Applications node under Reference Objects Management, you must create subclasses for the reference object.

For example, first enable an Application called: Oracle Cloud

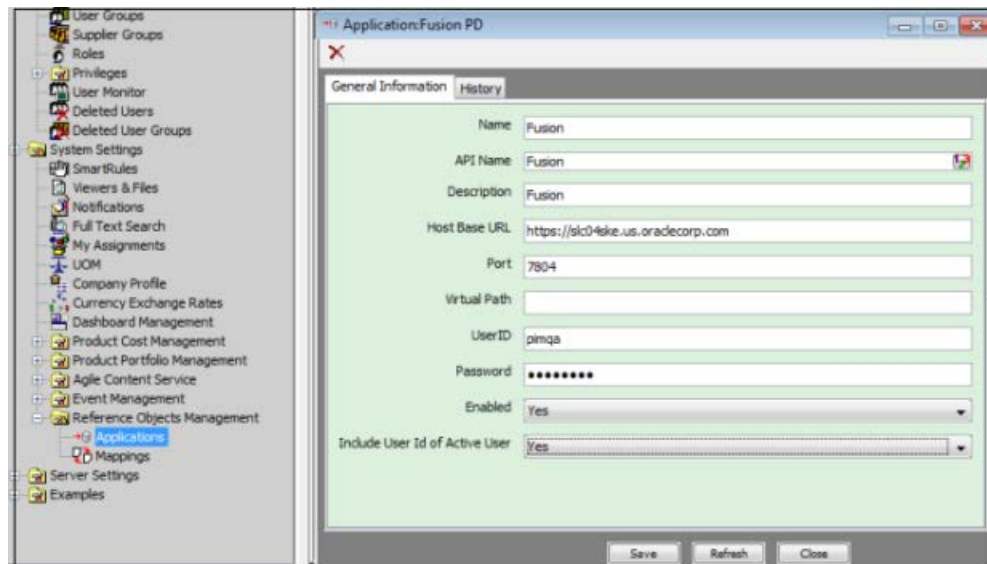
Then create two Reference Object subclasses: PD Item and PD Change.

Using the existing Reference Object Management settings, the administrator must configure the Reference Object in Java Client to connect to the Product Development Cloud environment. You have to configure the Reference object application before the mappings.

The procedure to create a reference object and its subclasses is explained here. The Reference Objects feature is not automatically enabled in Agile PLM.

To configure the Product Development external application:

1. In Java Client Administrator, navigate to **System Settings > Reference Objects Management**.
2. Double-click the **Applications** node and, in the window that appears, click **New**.
3. Fill in the required fields.



Example values for the required fields:

Name: Oracle Cloud

API Name: This API Name is also used later as the value for CLOUD_INTEGRATION_APPLICATION_APINAME property in CloudIntegration.properties file.

Host Base URL: Use the Domain URL of the Cloud Instance. Ensure that the Host Base URL includes the protocol and not the context path or port.

Port: Enter port number of the Cloud Instance.

Virtual Path: Enter the virtual path.

User ID: If SAML is configured to use Agile login user authentication, leave this field blank. If you are using common user authentication to access PD Cloud, provide that user name.

Password: Use the active user password

Enabled: Select Yes in the drop-down list to enable the application.

Include User ID of active user: Select Yes in the drop-down list.

4. Click Save.

To enable reference objects classes in Java Client:

In Java Client Administrator, navigate to **Data Settings > Classes**, double-click **Classes**, and double-click the **Reference Object** class.

In the window that appears, select Yes in the drop-down list to enable the Reference Object class.

Click Save.

With the Reference Object Class, each external application must have its own subclasses. For detailed information about Agile PLM object classes, see 9.3.6 *Agile Product Lifecycle Management Administrator Guide*, “Classes” chapter.

To create reference object subclass:

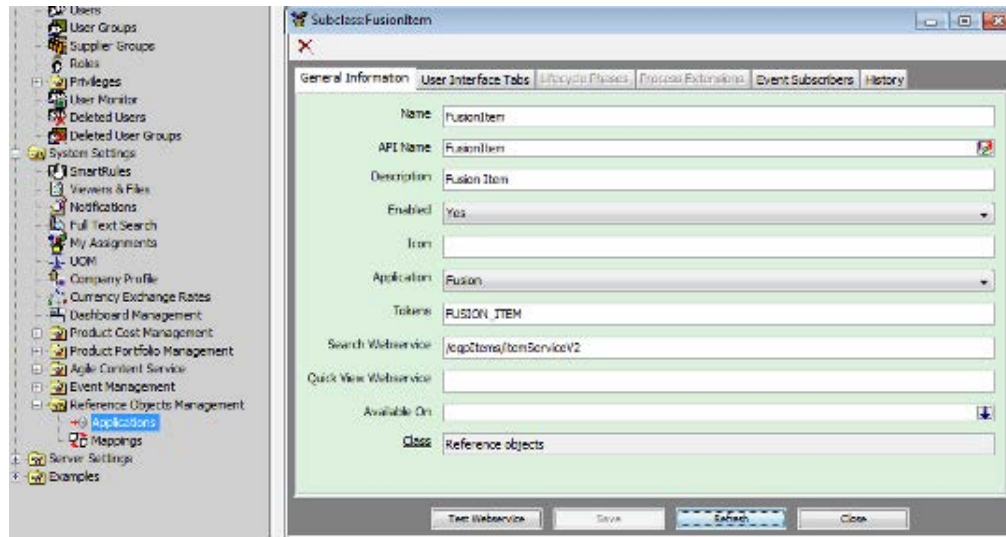
In this procedure, you can create subclasses called "PD Items" and "PD Changes".

In Administrator, navigate to **Data Settings > Classes**. Select and double-click **Classes**.

Scroll to select and double-click **Reference Objects** class name.

In the class properties window, click the **Subclasses** tab and click the **New Subclass** button. Enter suitable names for two subclasses to identify PD items and PD changes.

Fill in the required fields.



This table lists the sample values for the required fields of the Item subclass:

Field Names	Description
Name	PD Item
API Name	Use the exact API Name in the CloudIntegration.properties' CLOUD_PD_ITEM_REF_OBJECT_SUBCLASS_APINAME property.
Description	-Description of the item.
Enabled	Select Yes in the drop-down list.
Icon	Enter the location of a custom icon
Application	Oracle Cloud
Tokens	FUSION_ITEM
Search web services	'/fscmService/ItemServiceV2'
Quick View web services	N/A
Available On	Design; DFCO
Class	Reference Objects

Similarly, the values for Changes need to be listed to create a Changes subclass.

This table lists the sample values for the required fields of the Changes subclass.

Field Name	Description
Name	PD Change
API Name	Use the exact API Name in the CloudIntegration.properties'CLOUD_PD_CHANGE_REF_OBJECT_SUBCLASS_APINAME' property.
Description	-Description of the change.
Enabled	Select Yes in the drop-down list.
Icon	Enter the location of a custom icon.
Application	Oracle Cloud
Tokens	FUSION_CHANGE
Search web services	/fscmService/ProductDesignChangeOrderService'
Quick View web services	N/A
Available On	Design;DFCO
Class	Reference Objects

Click Save.

For each Reference Object subclass, you can verify that the web services connection to PD Cloud works correctly by using the Test Webservice function. Once you click the **Test Webservice** button, a message is returned that reports connection success or failure.

Define Reference Object Mappings for Items and Changes

Reference Objects Management > Mapping node enables you to map attributes from an external application to attributes on a configured Reference Objects subclass. You can create Reference Object Mappings for PD items and changes.

To define attribute mappings:

Attribute mappings are automatically created when the subclass is created.

Navigate to **Settings > System Settings > Reference Objects Management**.

Double-click the **Mapping** node to display the **Mappings** window. The defined **Reference Object** subclasses are displayed in a table.

Double-click the table row of the subclass you want to modify. The **Mappings** window appears.

Under **Attribute Mappings**, you can modify the **Web Services** attribute.

When you are finished in the **Mapping** dialog, click **OK**.

In Administrator, you can modify the existing Reference Object Attributes (Name, Description, Current Status) and save the changes. When a user logs in as a CAD Engineer (Design Engineer) and searches for PD Items on **Publish Workspace**, the modified Reference Object Attributes are visible.

For more details on reference objects and mappings, refer to Agile Product Lifecycle Management Administrator Guide, Rel. 9.3.6, in the chapter “Administering Reference Objects”.

Define Agile PLM Roles and Privileges

Agile PLM Roles provide a way to allocate a common set of Privileges to multiple users who have common functions in the development and integration processes.

In Administrator, for the CAD to Cloud integration, you will create Roles for the Engineering coordinator(s) and CAD engineer(s), per your company's needs, and to those roles add Privileges that allow access to designs, DFCOs, and reference objects. Generally, the role would not include privileges that allow access to the larger arena of Agile PLM parts, structures, or change orders (that is, to non-CAD-related sections of the company's products or services).

Note: Refer to Agile Product Lifecycle Management Administrator Guide, Release 9.3.6. The chapter "Roles and Privileges" provides much greater detail than herein for the construction of tailored Roles and how PLM Privilege Masks work. This is of particular importance for setting a consistent and rigorous access and security policy across your company's Agile PLM users.

After you publish a design from Agile PLM to PD Cloud, the updated item revision for the design is displayed in the Title Page's Revision attribute. You cannot perform Un-relate, Publish, Search, or Relate actions if you do not have the **Modify** privilege on any of these: Design.P2.Tex12, Design Revision, or Design.P2.Text30. An error message - "Modify privilege is required. Please check with the Agile administrator." - will appear in the information area of the **Publish Workspace** tab.

Role	Applies to these base classes	Description
Example role: CAD for Cloud	Reference Objects, Designs	CAD for Cloud user

This table lists the privileges that you must assign to the role that you created.

Name	Description	Privilege
Add to Table All Design File Changes	Add to Table All Design File Changes	Add to Table
Approve / Reject File Folder Versions	Approve / Reject File Folder Versions	Approve / Reject
Attachment Redline for File Folders for Others	Attachment Redline for File Folders for Others	Attachment Redline for Others
Attachment Redline for File Folders for Self	Attachment Redline for File Folders for Self	Attachment Redline for Self
Cancel Checkout All Design File Changes	Cancel Checkout All Design File Changes	Cancel Checkout
Cancel Checkout Designs	Cancel Checkout Designs	Cancel Checkout
Checkin All Design File Changes	Checkin All Design File Changes	Checkin
Checkin All Markups	Checkin All Markups	Checkin
Checkin Designs	Checkin Designs	Checkin
Checkin for File Folders	Checkin for File Folders	Checkin
Checkout All Design File Changes	Checkout All Design File Changes	Checkout

Name	Description	Privilege
Checkout All Markups	Checkout All Markups	Checkout
Checkout Designs	Checkout Designs	Checkout
Checkout Designs by Folder Owner Only	Checkout Designs by Folder Owner Only	Checkout
Checkout for all Items	Analyst Checkout for Items	Checkout
Checkout for File Folders	Analyst Checkout for File Folders	Checkout
Checkout for My User Groups	Checkout for My User Groups	Checkout
Checkout for Myself	Checkout for Myself	Checkout
Checkout My Markups	Checkout My Markups	Checkout
Comment for File Folders	Comment for File Folders	Comment
Create Designs	Create Designs	Create
Create DFCO	Create Design File Change Orders	Create
Create File Folders	Create File Folders	Create
CS - Cancel DesFileChgOrder	CS - Cancel DesFileChgOrder	Change Status
CS - Hold DesFileChgOrder	CS - Hold DesFileChgOrder	Change Status
CS - Implement DesFileChgOrder	CS - Implement DesFileChgOrder	Change Status
CS - Release DesFileChgOrder	CS - Release DesFileChgOrder	Change Status
CS - Return DesFileChgOrder	CS - Return DesFileChgOrder	Change Status
CS - Route DesFileChgOrder	CS - Route DesFileChgOrder	Change Status
CS - Submit DesFileChgOrder	CS - Submit DesFileChgOrder	Change Status
Delete Designs	Delete Designs	Delete
Delete File Folders	Delete File Folders	Delete
Delete for all File Folders	Delete for all File Folders	Delete
Discover Designs	Discover Designs	Discovery
Discover Designs I have Created	Discover Designs I have Created	Discovery
Discover DFCO	Discover DFCO	Discovery
Discover File Folders	Discover File Folders	Discovery
Discover File folders I have Created	Discover File folders I have Created	Discovery
Discover My File Folders	Discover My File Folders	Discovery
discover reference objects	NA	Discovery
Discover Users	Discover Users	Discovery
Enforce Field Level Read	Enforce Field Level Read	Enforce Field Level Read
GET Designs	GET Designs	GetFiles
GetFile All Design File Changes	GetFile All Design File Changes	GetFiles
GetFile for File Folders	GetFile for File Folders	GetFiles
Incorporate Items	Incorporate Items	Incorporate
Modify All Markups	Modify All Markups	Modify
Modify Design File Changes	Modify Design File Changes	Modify

Name	Description	Privilege
Modify Designs	Modify Designs	Modify
Modify File Folders	Modify File Folders	Modify
Print Designs	Print Designs	Print Designs
Publish to External Item	Publish to External Item Buttons on Publish Workspace tab disabled users without privilege.	Publish to External Item
Read All Markups	Read All Markups	Read
Read Changes	Read Engineering Changes	Read
Read Design File Changes	Read Design File Changes	Read
Read Designs	Read Designs	Read
Read File folders I have Created	Read File folders I have Created	Read
Read My File Folders	Read My File Folders	Read
Read My Markups	Read My Markups	Read
Read RO-Change	Read RO-Changes	Read
Read RO-Item	Read RO-Item	Read
Read Users	Read Users	Read
SaveAs DFCO	SaveAs DFCO	Save As
Unincorporate Items	Unincorporate Items	Unincorporate
View Designs	View Designs	View Designs

Map PLM Design/Change TB / P2 Attributes to PD Item EFF / Change DFF Attributes

You can map PLM Design/Change Title Block and Page Two attributes to PD Item Extensible Flex-Field (EFF) / PD Change DFF attributes for CAD Data Publish.

Also, PLM Release 9.3.6.9 adds the capability to map PLM Design Title Block and Page Two **List** attributes to PD Item EFF Single List / Multi rows attributes for CAD Data Publish.

Note: Mapping to Extensible Flex_Field (EFF) on Root Item Class is supported.

CAD for Cloud does not support mapping to ItemRevision and ItemSupplier context usage.

Template of C4C_Mapping.xml Configuration File

An understanding is required of Product Development Cloud's Item Class, EFF CategoryContext (Attribute Group), EFF Attributes, Change Class, DFF Context Segment, and DFF attributes.

PD Item Class: 1. Class Name; 2. Internal API name

EFF CategoryContext: 1. Display Name; 2. Code; 3. API Name

EFF Attribute in a CategoryContext: 1. Name; 2. Code; 3. API Name

PD Change Class: 1. Class Name; 2. Internal API name

DFF Context Segment: 1. Context Type

DFF Attribute in a Context Type: 1. Name; 2. Code; 3. API Name

The PD Cloud administrator must:

Complete configuration of Product Development Item EFF and Change DFF attributes if enabled;

Deploy the EFF/DFF configuration in PD Cloud;

Regenerate the WS Java Client stub jars; and,

Package them in Agile PLM's **application.ear** file.

For more details on re-generating jar files, refer to [Generating PD Cloud WS Java Client Stub Jars](#).

The default mapping file **C4C_Mapping.xml** is shipped with Agile PLM installation with CAD to Cloud mapping disabled.

Notes to **C4C_Mapping.xml** default template file:

If **<MappingEnabled>** is configured to be False, the C4C_Mapping is disabled: the loading and validation of C4C_Mapping.xml configuration will be skipped and ignored.

<PDItemClass> defines a Fusion Item Class and has attributes and one or more **<CategoryContext>** elements.

Attribute (classname = "CADItem1"): "CADItem1" is Item Class name

Attribute (classinternalapiname = "CADItem1"): "CADItem1" is Item Class Internale API name

Attribute (uid = "caditem1clz"): "caditem1clz" is user specified UID for PD Item in mapping;

One or more **<CategoryContext>** element configured for this Item class.

<CategoryContext> defines EFF Attribute Context that associates with Item class

Attribute (categorytype = "Caditem1"): "Caditem1" is the generated WS Item EFF category java classname, and can be found in package

"com\oracle\xmlns\apps\scm\productcatalogmanagement\advanceditems\flex\egoitemeff\item\categories" of WS stub jar file.

Attribute (contexttype = "Caditemcontext1APIName"): "Caditemcontext1APIName" is EFF Attribute context API name, and can be found in package

"com\oracle\xmlns\apps\scm\productcatalogmanagement\advanceditems\flex\egoitemeff\item\contexts" of WS Stub jar file.

Attribute (uid = "caditem1.caditemcontext1apiname"): UID for CategoryContext in format of category + "." + context.

<DesignClass> defines Agile Design subclasses and it's table supported. Currently only "TitleBlock" and "PageTwo" tables are supported.

Attribute (subclassname = "C4CDesign"): Design Subclass API name

Attribute (uid = "c4cdesign"): UID for Design Subclass

One or more **<Table>** elements;

<**Table**> defines Design subclass table supported in Design class.

Attribute (tablename="TitleBlock"): either "TitleBlock" or "PageTwo"

Attribute (uid="tb"): UID for table in <DesignClass>

< **PDItemAttribute**> defines Fusion Item attributes that are used in mapping configuration. It uses the following attributes.

uid: UID for specified Item attribute. It is recommended to use syntax "caditem1clz.caditem1.caditemcontext1apiname.cadattr1apiname" for EFF attribute. They are corresponding UID of Item class, CategoryContext, and attribute API name; or "caditem1clz.itemnumber" for non EFF attribute (not used in current release) that are corresponding UID of Item Class, and attribute API name;

type: Data type of Item Attribute. String, Numeric, and SingleList attributes are supported for mapping.

name: attribute API name of Item (case sensitive). Recommend no special character and low case for first letter in EFF attribute API name

eff: to specify if this is EFF attribute (true/false).

< **PDItemAttributes**> contains one or more < PDItemAttribute> elements in this configuration file for mapping.

< **DesignAttribute**> defines Agile Design subclass attributes that are used in mapping configuration, and has the following attributes:

uid: UID for specified Agile Design subclass. It is recommended to use syntax "c4cdesign.tb.description" that are corresponding to UID of defined DesignClass, Table and attribute API name;

type: Data type of Design attribute. String, Numeric, and List attributes are supported for mapping.

name: attribute API name of Design (case sensitive)

< **DesignAttributes**> contains one or more < DesignAttribute> elements in this configuration file for mapping.

<**ChangeClass**> defines Agile Change object subclasses and it's table supported. Currently only Cover Page and "PageTwo" tables are supported.

Attribute (subclassname = "DFCO"): Change Object Subclass API name

Attribute (uid ="dfco"): UID for Change Subclass

One or more <Table> elements;

<Table> defines Change object subclass table supported in Change class.

Attribute (tablename="TitleBlock"): either "TitleBlock" or "PageTwo"

Attribute (uid="cp"): UID for table in <ChangeClass>

<ChangeAttribute> defines Agile Change subclass attributes that are used in mapping configuration, and has the following attributes:

uid: UID for specified Agile Change subclass. It is recommended to use syntax "dfco.cp.number" that are corresponding to UID of defined ChangeClass, Table and attribute API name;

type: Data type of Design attribute. String, Numeric, and List attributes are supported for mapping.

name: attribute API name of Design (case sensitive)

<ChangeAttributes> contains one or more <ChangeAttribute> elements in the configuration file for mapping

<PDChangeClass> defines a Fusion Change Class and has attributes and one or more <ContextSegmentDFF>

Attribute (classname="C4C_PD_CO_TYPE"): "C4C_PD_CO_TYPE" is Change Class Type names

Attribute (classinternalapiname = "C4C_PD_CO_TYPE"): "C4C_PD_CO_TYPE" is Change Class Internal API names

Attribute (uid = "c4cpdcotypeclz"): "c4cpdcotypeclz" is user specified UI for PD Change in mapping

<ContextSegmentDFF>

Attribute (contexttype = "C4cPdCoType"): "C4cPdCoType" is the API name for context in Context Sensitive Segments for Change Order Type - C4C_PD_CO_TYPE

Attribute (uid = "ctxdff"): "ctxdff" is user specified UI for context in mapping

<PDChangeDFFAttributes>

uid: UID for specified PD Change DFF attribute. It is recommended to use syntax "c4cpdcotypeclz.ctxdff.c4cgsattrtext1" for DFF attribute. They are corresponding UID of Change class, ContextSegmentDFF, and DFF attribute API name;

type: Data type of Change DFF Attribute. String, Numeric, and SingleList attributes are supported for mapping.

name: attribute API name of Change DFF (case sensitive). Recommend no special character and low case for first letter in EFF attribute API name

dff: to specify if this is EFF attribute (true/false), default: true.

<AttributeMap> defines the mapping between Design attribute and Item Attribute, and has the following attributes:

designattribute/changeattribute: one of UID defined in
<DesignAttributes>/<ChangeAttributes>;

pditemattribute/pdchangedffattribute: one of UID defined in <
PDItemAttributes>/<PDChangeDFFAttributes>

< **MappingEntry**> defines mapping entry. It contains one or more <AttributeMap> element.
 type: mapping type. Currently "RedlineEFF" and "MapChangeDFF" types are supported.
 name: user specifies mapping names

Loading, Parsing, and Validating C4C_Mapping.xml File

The C4C_Mapping.xml file can be found in the folder AGILE_DOMAIN\config, and is loaded, parsed, and validated during Agile Server startup. The statuses of loading, parsing and validating are logged in a log file in the folder

AGILE_DOMAIN\servers\YOURAGILESERVERNAME\logs\c4cmapping.

The log file name is in the format of status-MM-DD-YYYY.log and the example here is status-06-22-2018.log, where:

DD - day (digits) of Agile PLM server startup day

MM - month (digits) of Agile PLM server startup day

YYYY - years (four digits) of Agile PLM server startup day.

The log file name and its folder is printed out in WebLogic Server console:

Agile log filename: C:\builds\agile9367\wls\apcm\agileDomain\servers\
 SLC15ALJ-AgileServer\logs\c4cmapping\status-06-22-18.log

There are NO error messages while performing C4C mapping, file loading, parsing, and validating, as this is only for Agile Administrator configuration task.

If CAD integration is not enabled (CLOUD_CAD_INTEGRATION_ENABLED is FALSE in CloudIntegration.properties), no loading, parsing, or validation for C4C_Mapping.xml will be performed, and no logging status.

If CAD integration is enabled (CLOUD_CAD_INTEGRATION_ENABLED is TRUE in CloudIntegration.properties) and MappingEnabled in C4C_Mapping.xml is FALSE, no further loading, parsing and validation is performed with log status of "MappingEnabled: disabled"

In WebLogic Server console, while starting Agile PLM, "ERROR: C4C_Mapping.xml configuration file is not valid. Please contact your Agile Administrator!" may be found in the console if mapping configuration file is INVALID. The status logs will be appended to the end of the same log file if Agile server is started, restarted multiple times within the same day.

Here is one sample log file, status-06-22-18.log", for starting and restarting Agile server with the same C4C_Mapping.xml file, with only the difference of WS Java stub jars without EFF configured. Let's briefly go through some of them:

Status of Loading, Parsing, and Validating

CADforCloud Integration: enabled

MappingEnabled: enabled

Configuration file is invalid.

Validation errors will be logged in the log file.

The mapping file will not be loaded and parsed any further if CAD to Cloud integration and MappingEnabled are disabled. Agile Administrator needs to resolve any configuration error for CAD to Cloud publishing, if MappingEnabled is enabled.

During Agile server startup, there is NO validation check for classname and classinternalapiname while validating PDItemClass. The Administrator must make sure that the correct classname and classinternalapiname for PDItemClass are configured and case sensitive. If they are not configured correctly, the error will ONLY be reported during the CAD for Cloud

Publish (EFF redlining at runtime). The names of EFF Category, Context, and attribute will be validated by examining if the corresponding Java classes or attributes can be found in the WS stub jar file. Please note that they are the corresponding API names (case-sensitive) in PD Cloud configuration.

Validation of Designclass, table and its attribute will be performed with Agile live data in DesignClass, DesignAttribute, and DesignAttributes. This will ensure that Design subclass is defined and enabled. The table configured is either Title Block or Page Two, and existing attributes enabled with the correct data type.

Validation for AttributeMap and MappingEntry will check if designattribute and pditemattribute are defined early and if they are valid with the same data type between Design attribute and PD Item EFF attribute.

Objects	Description
PDItemClass Parsing	One or more PDItemClass can be configured in the mapping file, but only first PDItemClass mapping for the same PD Item class name is considered.
DesignClass Parsing	One or more DesignClass can be configured in mapping file, but only first DesignClass for the same Design class name is considered.
PDItemAttributes Parsing	Only first PDItemAttributes is considered in parsing if there are more than one element.
DesignAttributes Parsing	Only first DesignAttributes is considered in parsing if there are more than one element. For the validation to be passed, Design Title Block, Page Two table and attributes need to be enabled.
MappingEntry Parsing	One or more MappingEntry can be configured in mapping file, but only "RedlineEFF" type is considered for now.

Publish the Mapped Design TB / P2 Attributes to PD Item EFF Attributes

If C4C Mapping is enabled in C4C_Mapping.xml and one or more mapping is invalid, this error message appears: "Invalid EFF Mapping configuration. Please contact Agile Administrator." This message appears in the business object's Tab Header area when clicking **Validate** or **Publish** in Publish Workspace. No other validation or publishing will be performed until the error is corrected.

If C4C Mapping is valid, it will continue with other validations. Publish EFF Attributes will be performed after publishing Item Relationship, and will report error in **Details**, if any. If not, it will show Completed in the **Status** of Publish Workspace.

If Change Class mapping is enabled in C4C_Mapping.xml, loading, parsing and validating C4C_Mapping.xml is similar to above and the mapped Change Cover page/P2 attributes will be published to PD Change DFF attributes.

Configuring Rule-Generated Item Numbers

In order to support the Rule-Generated Item Number feature on Product Development Cloud (that has been integrated with CAD for Cloud), you must complete the following procedure.

More complete information and procedures can be found in *Oracle SCM Cloud Implementing Product Management*, Chapter 6 "Defining Product Rules" (for example, section "Item Rule Numeric Functions and Operators").

1. Create an EFF context group meant for EFF ItemNumber attribute in PD Cloud and note its API name (for example, CadItemNumberRuleEffCtx).

2. Create an EFF Attribute within the above context group and note its API Name (for example, CadItemNumberRuleEffAtt). This is the attribute used during rules generation.

Note: An existing context can also be used for this purpose, but an exclusive EFF attribute should be made available for Item Numbers.

3. Create a new Rule Set by opening the **Navigator** and going to **Setup and Maintenance**. In the drop-down menu next to **Setup**, choose **Product Management**. Select **Product Rules**, then click **Management Item Rule Sets**. Click + ("plus") to add a new Rule Set: using the **Manage Rule Sets** task, configure two sequential rules, one rule for matching the value of EFF Attribute to a fixed constant (for example, 'ItemSequence'), and a second rule for not matching.

- a. If EFF attribute does not match a fixed constant (such as ItemSequence), the Rule should return the value contained in the EFF Attribute itself for Item Number. This takes care of creating Item Objects from the CAD Publish function, where Item Numbers are predefined (retrieved AutoNumber from Agile sources).

Example:

The screenshot displays the Oracle Product Rules configuration interface. The top section, 'General Information', shows the rule set 'CAD Item Number Rule Set' with a description: 'if (EFF attribute == "PIMSequence") Item Number = Value from PIM sequence Generator else Item Number = value of EFF attribute'. Below this, the 'Rules' tab is active, showing a table of rules. The table has columns: Sequence, Name, Description, Business Entity, Return Type, Target Business Entity, and Target Group. Two rules are listed: '10 ItemNumber from CAD' and '20 ItemNumber from PIM'. Below the table, the 'ItemNumber from CAD: Details' section shows the 'Primary IF Expression' set to 'true'. The 'Secondary IF Expression' is '[Item].[CAD CTX Item Number].[CAD EFF Item Number] != "PIMSequence"'. The 'Return Value' is '[Item].[CAD CTX Item Number].[CAD EFF Item Number]'.

Sequence	Name	Description	Business Entity	Return Type	Target Business Entity	Target Group
10	ItemNumber from CAD	Item Number sourced by CAD Integration	Item	Item number		
20	ItemNumber from PIM	Item Number sourced by PD Sequence	Item	Item number		

ItemNumber from CAD: Details

Primary IF Expression: true

Secondary IF Expression: [Item].[CAD CTX Item Number].[CAD EFF Item Number] != "PIMSequence"

Return Value: [Item].[CAD CTX Item Number].[CAD EFF Item Number]

- b. If the EFF attribute matches a fixed constant (for example, 'ItemSequence'), the Rule should return Item Number by processing the equation or method defined in the 'Return Value' of the rule. This rule also takes care of creating Item objects using Oracle Application (Fusion) UI or Import.

Example:

Secondary IF Expression	Return Value
* THEN Expression	[Item].[CAD CTX Item Number].[CAD EFF Item Number] * PART + auto_sequence([CAD_Item_Number], 1000, 1) == PIMSequence

4. Update these two properties with the API Name values (case-sensitive) in CloudIntegration.properties file, for example:

```
ITEM_EFF_CONTEXT_FOR_ITEM_NUMBER_RULE =
CadItemNumberRuleEffCtx
```

```
ITEM_EFF_ATTRIBUTE_FOR_ITEM_NUMBER_RULE =
CadItemNumberRuleEffAtt
```

Adding the Publish Workspace Tab in Java Client

The Change Order class in Java Client has a new Publish Workspace tab which is not visible by default. Set the CLOUD_CAD_INTEGRATION_ENABLED property to true for the Publish Workspace tab at the subclass level.

To enable the Publish Workspace tab in a DFCO subclass:

1. In Java Client > Classes > Change Order class, create a subclass for Design File Change Orders (DFCO) - as always, how you name it is per your company's setup.
2. Open the User Interface tab, and set `<?xml version="1.0" encoding="utf-8"?>`
3. Select the Publish Workspace tab.

You can see that the Visible property of the Publish Workspace tab is set to No.

4. To enable the tab, double-click the Publish Workspace row and set Visible to Yes. A message, "Publish Workspace tab requires the Affected Files tab. Please set Affected Files tab to Visible=Yes first" appears.
5. Set the Affected Files tab to Yes (enabled); set the Publish Workspace tab to Yes. Note that with this subclass, you cannot enable both the Affected Files tab and the Affected Items tab at the same time; you must enable only one of these two tabs and not both at the same time.

Now you can successfully enable the Publish Workspace tab. The tab can only be enabled from the subclass level and not from the class level. If you enable it at a class level, this message appears: "You cannot do this action on <e.g., Change Orders>." All the attributes appear in the class level.

WebLogic Server Configurations

The administrator must set up WebLogic Server to enable CAD for Cloud.

Configuring WebLogic Server

Since we connect to PD using HTTPS, host name verification can be turned off in this case. Use the following credentials to log in to the WebLogic Server console.

Once you have successfully logged in to the Administration Console, perform the following steps to turn off WebLogic Server hostname verification:

1. Click Servers under Domain Structure > agileDomain > Environment.
2. Click the server name in the Summary of Servers pane.
3. In the Settings pane for the server, click the SSL tab.
4. Expand the Advanced Section.
5. Look for Hostname Verification and verify that it is set to None. In the Cluster server, set the Hostname Verification to None for admin server and all managed nodes. If Hostname Verification is not set to None, click the Lock and Edit button and change the value to None.
6. Save your changes, and click the Release Configuration button.

Setting Up the CloudIntegration.properties File

Once you have set up the key-store and imported the security certificates, you must confirm that the default properties in the **CloudIntegration.properties** file are set properly. Make changes as needed. The CloudIntegration.properties file is in the same location as the agile.properties file in the "../agileDomain/config" folder.

To set up the CloudIntegration.properties file:

1. Go to the AGILE_HOME\agileDomain\config folder and open the CloudIntegration.properties file.
2. Modify property values as needed.

The following table lists the main properties in the CloudIntegration.properties file and provides brief descriptions of each property.

Table 2–1 Properties in CloudIntegration.properties File

Property	Description
DESIGN_ATTRIBUTE_FOR_BOM_PUBLISH_OPTION	This property is Yes/No on every Design's P2 List attribute. Set to No disables the Design or a structure to be published to PD Cloud. Administrator can set for selective publish of a Design on a structure, or an assembly (structure) on a BOM (structure). See "Selective Publish with DESIGN_ATTRIBUTE_FOR_BOM_PUBLISH_OPTION" on page 3-14.
PUBLISH_FILETYPES	All CAD files will be added as "smart URLs" if there is no PUBLISH_FILETYPES property in Cloud-Integration.properties. A single CAD file will be added as a "smart URL" if its file type is listed in PUBLISH_FILETYPES property. A CAD file will not be added if its file type is blank (not listed) in PUBLISH_FILETYPES. See "Selective CAD File Types with PUBLISH_FILETYPES" on page 3-15.
CLOUD_CAD_INTEGRATION_ENABLED	This property takes a TRUE or FALSE value and controls the integration between Agile PLM and PD Cloud. Default is set to FALSE.
CLOUD_INTEGRATION_APPLICATION_NAME	Using Java Client Administrator, create an Application for Reference Objects Management and provide that name here for Cloud Integration. An example value is Oracle Cloud.

Table 2–1 (Cont.) Properties in CloudIntegration.properties File

Property	Description
CLOUD_PD_ITEM_REF_OBJECT_SUBCLASS_APINAME	<p>The Agile PLM administrator must set up the Reference Object subclass in Java Client. This property identifies the API Name of the Reference Object subclass in Agile under CAD for Cloud Integration that is configured to reference PD Items.</p> <p>An example value is PDItem.</p>
CLOUD_PD_CHANGE_REF_OBJECT_SUBCLASS_APINAME	<p>The Agile PLM administrator must set up the Reference Object subclass in Java Client. This property identifies the API Name of the Reference Object subclass in Agile under CAD for Cloud Integration that is configured to reference PD Changes (ECOs).</p> <p>An example value is PDChange.</p>
CLOUD_PD_ITEM_MASTER_ORG_CODE	<p>This property identifies the Master Organization that has been configured on the Cloud side for PD Items (an internal attribute). This property value is required to be set depending upon Cloud configurations for PD Items. The search and create of PD Items is limited by this Organization Code.</p> <p>If it is not set, an error message is displayed in the search results stating that the Master Organization has not been set.</p>
CLOUD_PD_MAX_SEARCH_RESULT_SIZE	<p>The maximum number of results to be returned in the Search window that is launched from Publish Workspace. The default is 50.</p> <p>A warning message appears to notify the user if, and when, the number of search results exceeds the maximum set by this property.</p>
INNOVATION_MANAGEMENT_RELATIONSHIP_SERVICE_PATH	Innovation Management web services End Point
SCMCOMMON_CHANGE_DEEPLINK_URL	PD Change deeplink template URL from Agile PLM to PD for Changes.
SCMCOMMON_ITEM_DEEPLINK_URL	PD Item deeplink template URL from Agile PLM to PD for Items.
CadToCloud.LinkType	<p>The API Name of Attribute on the Relationship table to Identify Link Type between Design and PD Item. The default is TEXT01.</p> <p>The system uses the Design.Relationships.TEXT01 attribute if CadToCloud.LinkType is not configured or configured attribute API Name does not exist.</p>
CLOUD_CAD_PUBLISH_USE_JMSJOBQUEUE	<p>This property controls the CAD for Cloud integration publish by using JMS or non-JMS job queue. The default is TRUE, which selects for using JMS job queue.</p> <p>If the property is set to FALSE, the C4C publish will use local job queue with the Publish Thread pool property below.</p>
CLOUD_CAD_PUBLISH_THREADS	<p>The number of Cloud CAD Publishing threads. The default is 5.</p> <p>No publishing threads will be started if CLOUD_CAD_INTEGRATION_ENABLED is not enabled.</p>
CLOUD_CAD_PUBLISH_THREAD_TIMEOUT	<p>Timeout (minutes) for CAD Publish thread since last updated, default 120 minutes.</p> <p>This timeout setting is not for the whole job publishing, and is for no more updated since task last updated. For example, last updated time is recorded at beginning of files upload to UCM and will be updated again when finishing upload files. Publish thread is considered to be stuck if no update for configured timeout since last update.</p>
CLOUD_PD_BATCH_SIZE_FOR_BULK_API	The number of objects used in PD Webservice by default=25.

Table 2–1 (Cont.) Properties in CloudIntegration.properties File

Property	Description
CLOUD_CREATE_PD_CHANGE_SUPPORTED	<p>The CAD for Cloud system can automatically create a PD Change Order if configured.</p> <p>This property takes a TRUE or FALSE value so that the system is configurable to enable/disable operation of creating changes on PD. Upon integration/installation, the default value for this property shall be set to 'FALSE'.</p>
CLOUD_CREATE_PD_ITEM_SUPPORTED	<p>This property takes a TRUE or FALSE value so that the system is configurable to enable/disable operation of creating items on PD. Upon integration/installation, the default value for this property shall be set to 'FALSE'.</p>
DESIGN_ATTRIBUTE_FOR_PD_ITEM_NUMBER	<p>The API Name of Attribute (Default is Blank) on Design Page Two table to define PD Item Number for auto-relating and auto-creating.</p> <p>The feature is not supported if it is not configured.</p>
DESIGN_ATTRIBUTE_FOR_PD_ITEM_CLASS	<p>The API Name of Attribute on Design Page Two to define list of 'Item Classes' for auto-create.</p> <p>Auto-Create is not enabled if the property value is blank or set to an invalid value.</p>
DESIGN_ATTRIBUTE_FOR_CREATE_PD_ITEM_FLAG	<p>The API Name of Attribute on Design Page Two to define list of 'Yes/'No' for object level Item create selective option.</p> <p>Auto-Create is not enabled if the property value is blank or set to an invalid value.</p>
DESIGN_ATTRIBUTE_FOR_PD_ITEM_UOM	<p>API Name of Attribute on Design Page Two to define 'Units of Measure' List for Auto-Create.</p> <p>Auto-Create is not enabled if the property value is blank or set to an invalid value.</p>
DESIGN_ATTRIBUTE_FOR_DISABLE_OBJECT_LEVEL_ATTACHMENT_PUBLISH	<p>Administrator can set to allow selective Attachment publish of Designs.</p> <p>The Default value false.</p> <p>If this property is True, (Design P2 List attribute. Set to No) disables the files from this Design to be published to PD Cloud.</p>
DFCO_ATTRIBUTE_FOR_PD_CHANGE_NUMBER	<p>API Name of Text Attribute on DFCO Page Two table to define PD Change Number for auto relate or auto create. Default is blank.</p>
DFCO_ATTRIBUTE_FOR_PD_CHANGE_CLASS	<p>API Name of Attribute on DFCO Page Two to define list of 'Change Order Types' for auto create. Default is blank.</p>
DFCO_ATTRIBUTE_FOR_CREATE_PD_CHANGE_FLAG	<p>API Name of Attribute on DFCO Page Two to define list of 'Yes/'No' for object level Change create selective option.</p>

Table 2–1 (Cont.) Properties in CloudIntegration.properties File

Property	Description
DESIGN_ATTRIBUTE_FOR_PD_ITEM_DIMENSIONS_UNIT	The Page Two attributes of Design object can be published the PD Item OOTB UOM attributes - Dimensions, Weight, and Volume during publish if being configured correctly. For example, PD Item UOM of "Weight" contains its type (Grams, KG...) and its numeric value, and needs two design.p2 attributes (list and numeric). Similarly, Dimensions contain its type (meter, foot...) and 3 numeric values (length, depth, width).
DESIGN_ATTRIBUTE_FOR_PD_ITEM_DIMENSIONS_WIDTH	
DESIGN_ATTRIBUTE_FOR_PD_ITEM_DIMENSIONS_LENGTH	
DESIGN_ATTRIBUTE_FOR_PD_ITEM_DIMENSIONS_HEIGHT	
DESIGN_ATTRIBUTE_FOR_PD_ITEM_WEIGHT_UNIT	
DESIGN_ATTRIBUTE_FOR_PD_ITEM_WEIGHT_WEIGHT	
DESIGN_ATTRIBUTE_FOR_PD_ITEM_VOLUME_UNIT	
DESIGN_ATTRIBUTE_FOR_PD_ITEM_VOLUME_VOLUME	
SAML_ISSUER_NAME	SAML token issuer to override the default value. This property is only used for SAML token based client policies.
ITEM_EFF_CONTEXT_FOR_ITEM_NUMBER_RULE	System Level support for Rule Generated Item Numbering by EFF Attribute Value. API Name of the EFF Context (Example: CadItemNumberRuleCtx) meant for EFF ItemNumber attribute.
ITEM_EFF_ATTRIBUTE_FOR_ITEM_NUMBER_RULE	API Name of the EFF Attribute (Example: CadItemNumberRuleAtt) on above Context configured for Rule Generated ItemNumber - This is the attribute used during Rules Creation.
CLOUD_ATTACHMENT_CATEGORY_CAD_2D_VIEWABLES	Attachment category for 2D viewable specific to CAD for Cloud integration with Fusion PD.
CLOUD_ATTACHMENT_CATEGORY_CAD_3D_VIEWABLES	Attachment category for 3D viewable specific to CAD for Cloud integration with Fusion PD.
CLOUD_ATTACHMENT_CATEGORY_CAD_DRAWINGS	Attachment category for Drawing specific to CAD for Cloud integration with Fusion PD.
CLOUD_ATTACHMENT_CATEGORY_CAD_MODELS	Attachment category for Model specific to CAD for Cloud integration with Fusion PD.
DESIGN_ATTRIBUTE_FOR_CLOUD_ATTACHMENT_CATEGORY =	<p>API Name of Attribute on Design Page Two to define 'Attachment Category' List configured in PD</p> <ol style="list-style-type: none"> Create a simple list in the Java Client having the values of Attachment Category that are configured in PD. Configure a Design Page Two list attribute to use the newly created list and rename it to PD Attachment Category (Optional) Set this property to the API name or the ID of the list attribute. <p>Override CLOUD_ATTACHMENT_CATEGORY_CAD_*** if this property is configured with a valid P2 List Attribute API Name/(ID) and attribute value of Design Object is NOT blank.</p>

Table 2–1 (Cont.) Properties in CloudIntegration.properties File

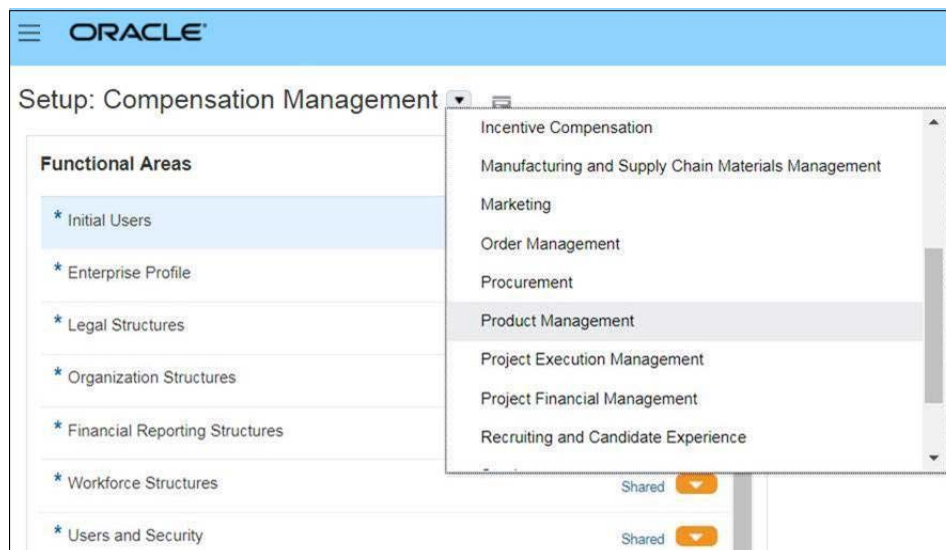
Property	Description
PUBLISH_UPLOAD_FILETYPES	This property allows the Agile PLM Administrator to configure the specific sets of file types to be uploaded to UCM. This property can be used in conjunction with PUBLISH_FILETYPES to provide fine grain control on the publishing process.
PUBLISH_UPLOAD_MAX_FILESIZE	The default of this property is set to 100MB. The administrator can adjust it based on the business requirement
PUBLISH_UPLOAD_MAX_FILES_INBULK	Max number of files allowed to be uploaded to Content Cloud in bulk The default value 25.
PUBLISH_UPLOAD_MAX_TOTALFILESIZE_INBULK	Property to configure allowable maximum total file size(MBs) of files uploaded to Content Cloud in bulk. The default value 100 MBs.
USE_DESIGN_SUBTYPE_FOR_DRAWING	System Level configuration parameter - to use Design Subtype for Drawing. USE_DESIGN_SUBTYPE_FOR_DRAWING = Yes in order to support NX data. This setting will use the Design.P2.SubType value to determine if the file is a Drawing or a Model. The MCAD Connectors set this value automatically for all supported Cad Systems.

Setting Up Attachment Categories in Root Item Classes

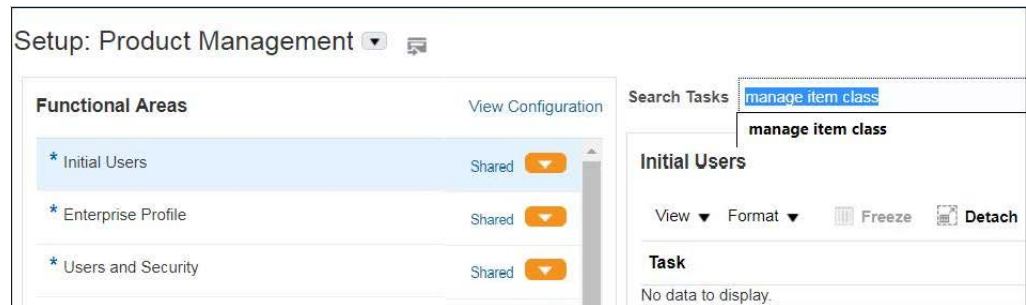
For the success of the redlineItemAttachments web services (ProductDesignChangeOrder) call, you must configure the attachment categories to have the value of Product Development in the Root Item Class level.

To configure the attachment categories:

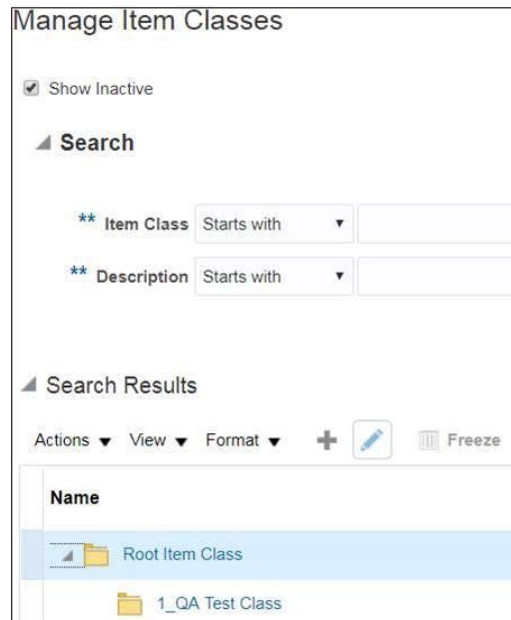
1. Log in to Oracle Fusion Applications Cloud with administrator credentials and navigate to the Setup and Maintenance work area.
2. Click Setup and Maintenance.
3. Select Product Management from the Setup drop-down list.



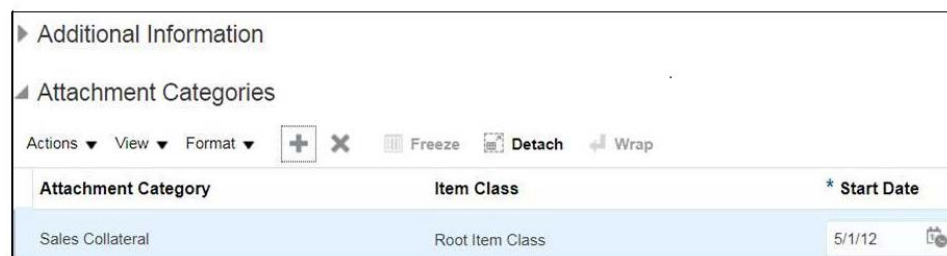
4. Search for Manage Item Classes in the Search Tasks.
5. Click Manage Item Classes from the search result.



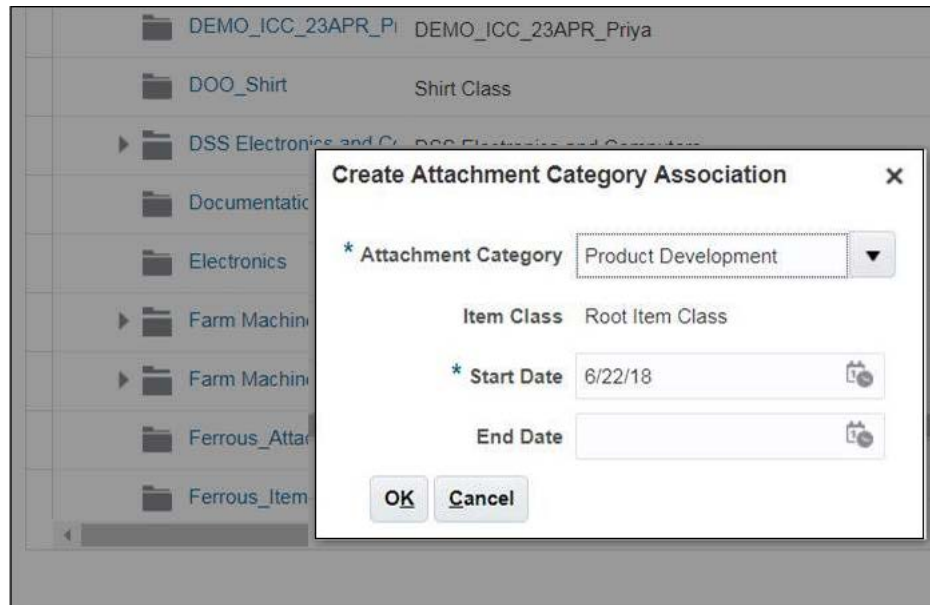
6. Select the row of Root Item Class and click Edit.



7. Scroll down to the bottom to find the Attachment Categories panel.



8. From the Actions menu, click Create.
9. In the Create Association window, select Product Development from the drop-down menu and click OK.



10. Click Save.

Setting Up Attachment Categories in CAD for Cloud

CAD for Cloud uses the following Fusion Product Development Attachment Categories when publishing from Agile. Ensure that they exist in Fusion Product Development. If they have not been created, please do so at this time before proceeding to the next section.

2D Viewables

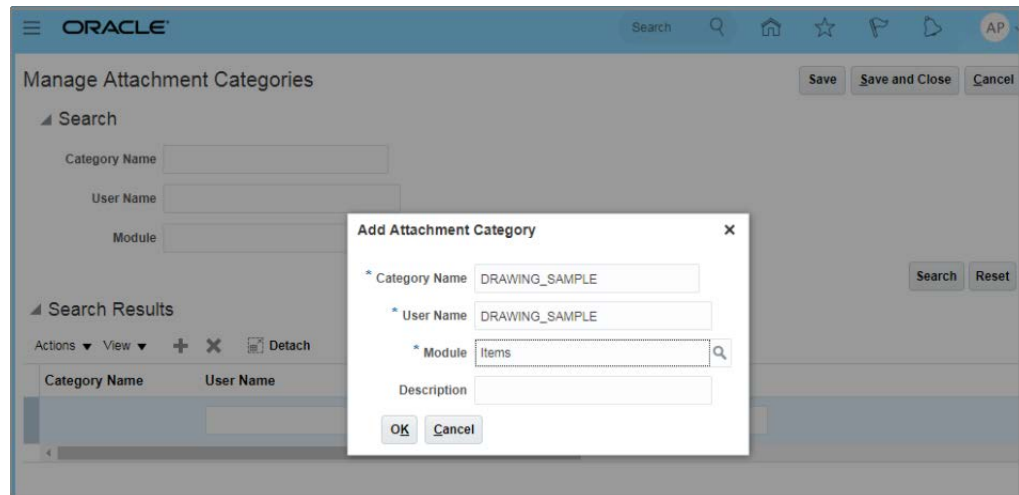
3D Viewables

Models

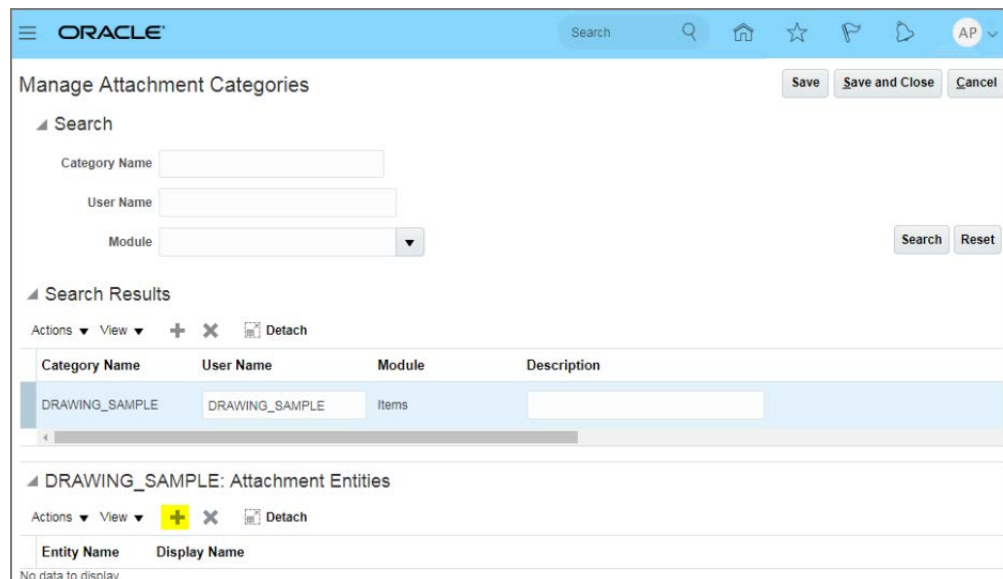
Drawings.

To configure the attachment categories in drawings:

1. Log in to Oracle Fusion Applications Cloud with administrator credentials and navigate to the Setup and Maintenance work area.
2. Click Setup and Maintenance.
3. Search for Manage Attachment Categories for Product Management in the Search Tasks.
4. Click Manage Attachment Categories for Product Management in the search results.
5. Click the + icon in the Search Results section, to create a new category.
6. Enter Items and click the search icon in the Modules field.



7. From the search results, select Acallitems.
8. In the Attachment Entities section, click the + icon to create the new entities.



9. For each category you create, you have to search and add the following entities:
 - ITEM_ENTITY
 - ITEM_REVISION_ENTITY
10. Click Save and Close after you have selected the search results.

Manage Attachment Categories

Search

Category Name

User Name

Module

Search Reset

Search Results

Actions View + X Detach

Category Name	User Name	Module	Description
DRAWING_SAMPLE	DRAWING_SAMPLE	Items	

DRAWING_SAMPLE: Attachment Entities

Actions View + X Detach

Entity Name	Display Name
ITEM_ENTITY	ITEM_ENTITY
ITEM_REVISION_ENTITY	ITEM_REVISION_ENTITY

11. Click Save and Close to complete the task.

Repeat the same steps for MODELS, 2D VIEWABLES, and 3D VIEWABLES.

Creating Attachment Category Association

To create attachment category association:

1. Log in to Oracle Fusion Applications Cloud with administrator credentials and navigate to the Setup and Maintenance work area.
2. Click Setup and Maintenance.
3. Select Product Management from the Setup drop-down list.

Setup: Product Management

Functional Areas

- * Initial Users Shared
- * Enterprise Profile Shared
- * Users and Security Shared
- Transactional Business Intelligence Shared
- Application Extensions Shared
- * Item Organizations
- * Items Shared
- Inventory Organizations
- Catalogs Shared
- * Structures Shared
- * Suppliers for Product Management
- * Item Mass Update

Search Tasks: MANAGE ITEM CLASSES

Items

View Format Freeze Detach Wrap Show All Tasks

Task	Scope
Manage Item Classes	Manage Item Class Descriptive Flexfields
Deploy Item Extensible Flexfields	
Manage Item Statuses	
Manage Item Types	
Manage Cross Reference Types	
Manage Item Descriptive Flexfields	
Download Import Template from Oracle Enterprise Repository	
Upload Item Data	
Load Interface File through Scheduled Process	

Columns Hidden 4

4. Click the Tasks icon and click Search in the pop-up.
5. Click Manage Item Classes from the search result.

6. Select the row of Root Item Class and click Edit.

Manage Item Classes

☒ Show Inactive

Search

Item Class Starts with:

Description Starts with:

Item Creation Allowed: Equals

New Item Request Enabled: Equals

Search Reset Save... Add Fields Reorder

Search Results

Name	Description	Enabled	Item Creation Allowed	New Item Request Enabled
Root Item Class	Root Item Class	✓	✓	—
ALLQA	All Item Classes for QA	✓	✓	—
ALM Item Class	Only to be used by ALM team	✓	✓	—
BAT_PD_NIRITEMCLASS	Create NIR item class	✓	✓	✓
BAT_PD_NONNIRITEMCLASS	Create NON-NIR item class	✓	✓	—

7. Scroll down to the bottom to the Attachment Categories section.

Attachment Categories

Actions View Format + X Freeze Detach Wrap

Attachment Category	Item Class	* Start Date	End Date
Sales Collateral	Root Item Class	5/1/12	
Product Development	Root Item Class	7/28/18	
C4C_2D_VIEWABL	Root Item Class	8/7/18	
3D Viewables	Root Item Class	8/13/18	
DRW	Root Item Class	8/7/18	
MODEL	Root Item Class	8/7/18	
2D Viewables	Root Item Class	8/13/18	
C4C_3D_VIEWABLE	Root Item Class	8/7/18	

8. From the Actions menu, click Create.

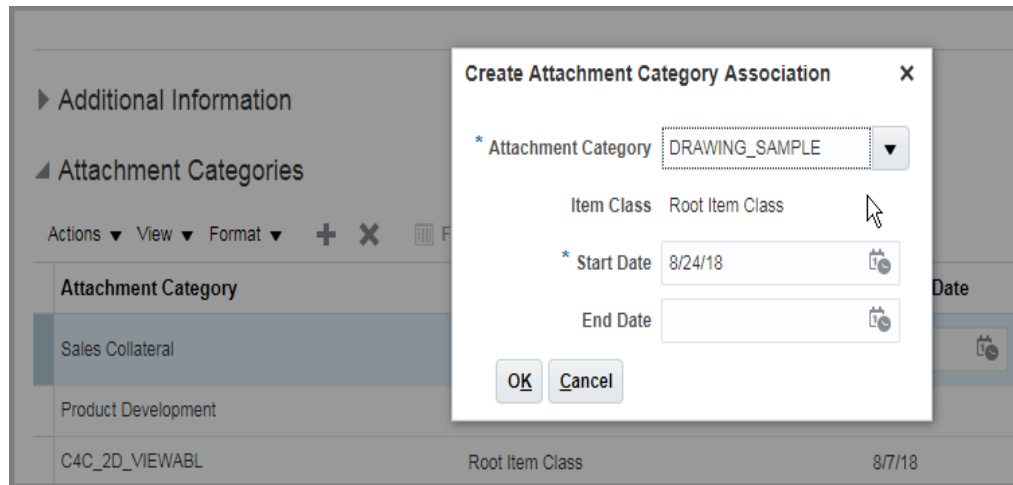
Attachment Categories

Actions View Format + X Freeze Detach Wrap

Create Delete

Attachment Category	Item Class	* Start Date
Sales Collateral	Root Item Class	5/1/12
Product Development	Root Item Class	7/28/18
C4C_2D_VIEWABL	Root Item Class	8/7/18
3D Viewables	Root Item Class	8/13/18
DRW	Root Item Class	8/7/18
MODEL	Root Item Class	8/7/18
2D Viewables	Root Item Class	8/13/18

9. Add the DRAWING_SAMPLE you have created earlier to the association.



10. Click OK.

Attachment Categories			
Actions View Format + X Freeze Detach Wrap			
Attachment Category	Item Class	* Start Date	End Date
DRAWING_SAMPLE	Root Item Class	8/24/18	
Sales Collateral	Root Item Class	5/1/12	
Product Development	Root Item Class	7/28/18	
C4C_2D_VIEWABL	Root Item Class	8/7/18	
3D Viewables	Root Item Class	8/13/18	
DRW	Root Item Class	8/7/18	
MODEL	Root Item Class	8/7/18	
2D Viewables	Root Item Class	8/13/18	

Repeat the same process for MODELS, 2D VIEWABLES, and 3D VIEWABLES.

Add the Category name to the corresponding properties in the CloudIntegration.properties file:

CLOUD_ATTACHMENT_CATEGORY_CAD_DRAWINGS = DRAWING_SAMPLE

CLOUD_ATTACHMENT_CATEGORY_CAD_MODELS = MODELS_SAMPLE

CLOUD_ATTACHMENT_CATEGORY_CAD_2D_VIEWABLES = 2D VIEWABLES_SAMPLE

CLOUD_ATTACHMENT_CATEGORY_CAD_3D_VIEWABLES = 3D VIEWABLES_SAMPLE

CAD File Types Supported in CAD for Cloud

The following are the file types supported in CAD for CLOUD:

CREO:

MODEL	DRAWING	2D VIEWABLE	3D Viewable
ASM, PRT	DRW	PDF, TIF	STP, IGS, X_T

SOLIDWORKS:

MODEL	DRAWING	2D VIEWABLE	3D Viewable
SLDASM, SLDPRT	SLDDRW	PDF, TIF	STEP, IGS, X_T

Administrator can configure a Design Page Two non-version-specific List Attribute to map to a Cloud PD Item Attachment Category

If the business requirement is such that the design needs to be mapped to different category based on what they are for, then you can set up the DESIGN_ATTRIBUTE_FOR_CLOUD_ATTACHMENT_CATEGORY to point to an Agile 9 Design Page Two list of attachment categories mirroring those configured in Cloud PD.

Here are the sample steps the administrator should follow to set up this feature.

Configure the Attachment Categories in Fusion, ex: CAD_SOURCE, CAD_VIEWABLE, CAD_SUPPLIER...etc

Create a list in JavaClient as "PD Attachment Category List" with entries like CAD_SOURCE, CAD_VIEWABLE, CAD_SUPPLIER...etc

Choose one of Design Page Two non version specific list attribute (list1 to list25), ex: list16 (optionally rename its name and API name to what you desire), and assign the list "PD Attachment Category List" to this attribute

Configure the parameter - DESIGN_ATTRIBUTE_FOR_CLOUD_ATTACHMENT_CATEGORY in CloudIntegration.properties to have the API name (ex: list16) or attribute ID (ex: 1276) of above Page Two list attribute

Restart the Agile server.

This feature is not enabled if anyone of the following conditions is true.

DESIGN_ATTRIBUTE_FOR_CLOUD_ATTACHMENT_CATEGORY is missing in the CloudIntegration.property

The value for DESIGN_ATTRIBUTE_FOR_CLOUD_ATTACHMENT_CATEGORY attribute is blank

The DESIGN_ATTRIBUTE_FOR_CLOUD_ATTACHMENT_CATEGORY is not set to one of the non-version specific list attributes (list1 - list25)

If the DESIGN_ATTRIBUTE_FOR_CLOUD_ATTACHMENT_CATEGORY is blank, and the administrator has set up the CLOUD_ATTACHMENT_CATEGORY_CAD_XXX, then the system will use the configured information to publish.

Registering with Agile PLM

Open Design and DFCO objects must register with Agile PLM. The Design/DFCO object that appears in the PD Item/ECO's relationship tab is a link that takes you to Agile PLM. Enter the values for protocol, host, port and context root in the corresponding fields to configure server details of the target system (Agile PLM). Log in to Product Development Cloud. Go to Setup and Maintenance > Product Management > Concept Design Management > Register Agile PLM and configure as below:

Values	Description
Server Protocol	Select from the menu options (http or https)
External Server Host	Enter the Agile PLM server name
External Server Port	Example: 7001

The following table mentions the Associated Module and its Context Root Value:

Associated Module	Context Root Value
AgilePLMWebClient	Agile

Generating PD Cloud WS Java Client Stub Jars

Note: We need to generate stub jars only when there are changes on PD web services. The below topic is optional. If required, follow the procedures mentioned below to generate stub jars.

In the CAD for Cloud integration, Agile PLM relies on three PD Cloud web services to collaborate with Product Development. They are web services of:

Item - ItemServiceV2

Change - ProductDesignChangeOrderService

Relationship - InnovationManagementRelationshipService

In many instances, you are required to update PD Cloud or change the EFF configuration, after the initial installation. Due to changes in Product Development Cloud WS API, the Product Development Cloud WS Java client stub jar files must be regenerated and deployed to Agile PLM again. This topic describes how you can generate them and the procedure to deploy them in Agile PLM's WebLogic Server.

Note: If you only have access to the HTTPS WSDL url to generate WS Java Client stub jar files, you must save the SSL certificates and import them to your JDK trust store that is used to launch wsimport.

Exporting the Certificate from the Browser:

To export the certificate from the browser:

1. Open the HTTPS WSDL url in the Internet Explorer browser.
2. Install the certificate by following online instructions.
3. Go to Tools > Internet Options > Content > Certificates.
4. In the Certificates dialog, switch to the "Immediate Certificate Authorities" tab and look for the recently installed certificate from this site.
5. Select the certificate and click the Export button.
6. Follow instructions and save it as OracleDevCert.cer (default option – DER encoded binary X.509 (.CER)) at a specified location.

Note: It is preferred to use JDK's wsimport in the Linux system to generate Java Client jars, even though the Windows Command is used here.

Certificates to the JDK Trust Store:

To import certificates to the JDK Trust store, perform the following steps in the Windows Command:

1. Go to the security folder of your JDK. For example, C:\jdk1.8.0_144\jre\lib\security.

2. View the certificate list with window command: `keytool -list -keystore cacerts`. If you do not find the certificate from wsdl in the list, click Next to add it.

Note: The default password for cacerts is changeit

3. Add the exported certificate into the JDK trust store: `keytool -import -alias OracleDevCert -keystore cacerts -file C:\wsimport\OracleDevCert.cer`

Generating the Change Service Jar:

To generate the change service jar, perform the following steps in the Windows Command prompt:

1. Create a new folder (c:\>md changes) and change to this folder (c:\>cd changes).
2. Make two sub folders "src" and "classes" by executing two commands (c:\changes>md src & c:\changes>md classes).
3. Execute the following command to generate the Change WS jar file:
`c:\changes>C:/jdk1.8.0_144/bin/wsimport.exe -B-XautoNameResolution -clientjar fusionPDChangeOrder_Service-11.13.17.11.0-12-01-2017.jar -s src -keep -d classes -verbose http(s)://<host_name>:<port_number>/fscmService/ProductDesignChangeOrderService?wsdl.`

In the example, HTTP WSDL URL is used. If https is used, you must import the certificate to the JDK trust store as described earlier.

Generating the Item Service Jar:

To generate the item service jar, perform the following steps in the Windows Command prompt:

1. Create a new folder (c:\>md items) and change to this folder (c:\>cd items).
2. Make two sub folders "src" and "classes" by executing two commands (c:\items>md src & c:\items>md classes).
3. Execute the following command to generate the Item Service WS jar file:
`C:\items>C:/jdk1.8.0_144/bin/wsimport.exe -clientjar fusionItem_Service-11.13.17.11.0-12-01-2017.jar -s src -keep -d classes -verbose http(s)://<host_name>:<port_number>/fscm/Service/ItemServiceV2?wsdl`

In the example, HTTP WSDL URL is used. If https is used, you must import the certificate to the JDK trust store, as described earlier.

Generating the Relationship Service Jar:

To generate the relationship service jar, perform the following steps in the Windows Command prompt:

1. Create a folder (c:\>md rels) and change to this folder (c:\>cd rels).
2. Make two sub folders "src" and "classes" by executing two commands (c:\rels>md src & c:\rels>md classes).
3. Execute the following command to generate the Relationship Service WS jar file:
`c:\rels>C:/jdk1.8.0_144/bin/wsimport.exe -clientjar fusionInnoMgmRelationship_Service-11.13.17.11.0-12-01-2017.jar -p com.agile.integration.fusion.relationship.ws -s src -keep -d classes -verbose http(s)://<host_name>:<port_number>/fscmService/InnovationManagementRelationshipService?wsdl`

Patching and Redeploying the .ear File:

After generating the three jar files, you are ready to patch the Agile PLM system. Remember that the jar files are in their corresponding "classes" folder.

1. You must stop the Agile PLM application.
2. Backup the original application file - "application.ear" in to any outside agileDomain folders.
3. Delete the folder under "<AGILE_HOME>\agileDomain\servers\"
4. Un-zip file "application.ear" and replace the 3 jar files of "fusionPDChangeOrder_Service-11.13.17.11.0-12-01-2017.jar", "fusionItem_Service-11.13.17.11.0-12-01-2017.jar" and "fusionInnoMgmRelationship_Service-11.13.17.11.0-12-01-2017.jar" located in the APP-INF/lib folder with the ones you generated, and repack them back to "application.ear"
5. Move this new "application.ear" file to "<AGILE_HOME>\agileDomain\applications\"
6. Start Agile to redeploy the patched .ear file.

Logging Request and Response Payloads

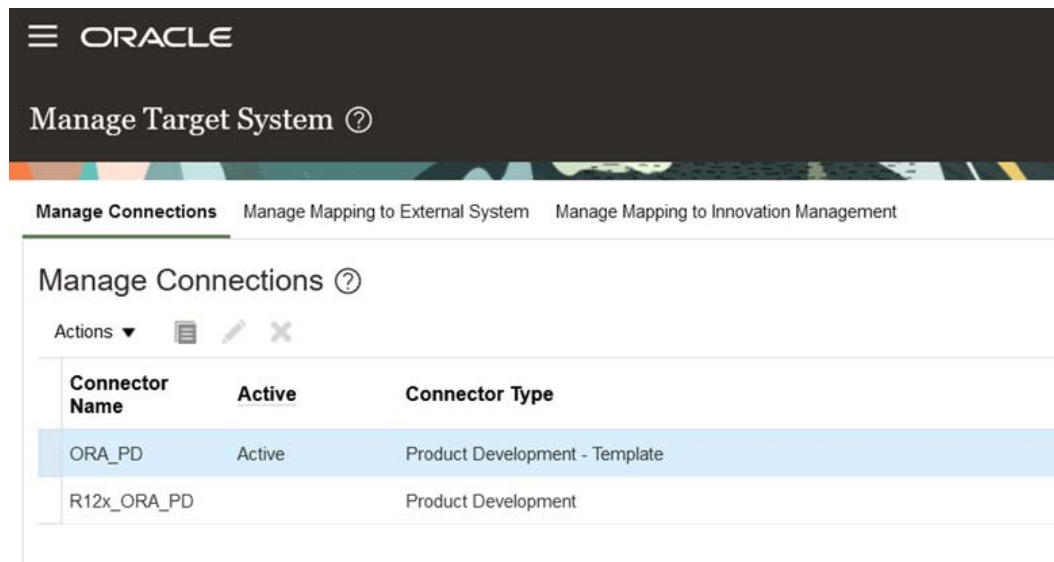
If functionality issues appear in the applications while working in the cloud environments, it is not easy to debug customer deployments because of the high restrictions to accesses. A detailed SOAP payload logging of client requests and server responses helps you to quickly locate the root cause of the issue.

With the Logging ON and Logging OFF feature, you can turn on (enable) or turn off (disable) logging SOAP payload of web-services requests and responses, through **Agile PLM Web Client > Tools and Settings > Administration > Logging Configuration**, set com.agile.integration.fusion.FusionServiceConfig to DEBUG level, while the application is up and running, without re-start.

All CAD for Cloud integration-related web-services requests and responses are logged to an exclusive CAD2Cloud_Integration.log file.

Manage Target System in Fusion

From FA 23B (11.13.23.04.0) PD connector is default in "Manage Target Systems" and A9 connector is hidden.



Using the CAD for Cloud Integration

CAD for Cloud functionality allows functional users in the CAD development process to coordinate and relate designs in Agile PLM to items in Product Development. This chapter must be read by the PLM administrator; and, it is addressed to users such as Engineering Coordinators and CAD Engineers.

Introducing Publish Workspace

In Agile PLM, the CAD for Cloud integration is supported primarily by the **Publish Workspace** tab that is available on a Design File Change Order (DFCO). As a CAD Design user, you can do the following from the Publish Workspace tab:

- Relate and unrelate a design in Agile PLM to a PD Cloud Engineering item (PD item, or item).

- Navigate from a design on **Publish Workspace** to its related PD item.

- Relate and unrelate an Agile PLM DFCO to a PD Cloud ECO.

- Navigate from a DFCO on **Publish Workspace** to its related PD Cloud ECO.

- Validate and publish a design (or design structure) to Product Development.

These tasks are covered in more detail in the following sections.

Using the Publish Workspace

Once the CAD for Cloud administration steps from the previous chapter are completed, the CAD for Cloud development process can be used to allow for an integration between Agile PLM and Product Development.

The following are some basic information about the Publish Workspace functionality:

- Designs that are designated as HelperPart are not visible on the Publish Workspace. They cannot be directly related to a PD item and cannot be published using the Publish Workspace.

- You can add generic and instance designs which are on the DFCO's affected files table to the Publish workspace. Once published to PD, Generic and Instance design components appear in the item's structure.

- The buttons of search functionalities on the **Publish Workspace** tab are disabled for any DFCO that is already Implemented, Released, or Canceled.

- If you select designs and click the **Publish** button, Publish Workspace displays the details of the Job ID and the Published By columns. When the ECO is related to the DFCO, clicking the **Unrelate ECO** button removes the data in the Job ID and Published By

columns. Likewise, if the item is related to the design and you click the **Unrelate Items** button, the data in these columns are removed.

When you hover over the Job ID a Quick View widget appears, which upon clicking displays the publish job details. The details available in the widget are:

- Job ID: The ID of the job you published.
- Published By: Name of the person who published the job.
- No. of design(s) being published: The number of designs you publish.
- Submitted On: The Date and time you submitted the publish job.
- Started On: The date and time you started the publish.
- Updated On: The date and time you updated the publish.
- Finished On: The date and time you finished the publish.
- Job Status: The status of the publish job: Submitted, In Process, Completed, Failed
- Current Process Step: The status the publish is in: Completed, Not Started, Auto-relating Item, Validating, Adding PD ECO Relationship, Adding PD ECO Affected Items, Adding PD Item Attachments, Redlining PD Item Structure, Adding PD Item Relationship.

When you select and publish a row from Agile PLM to PD Cloud or relate a PD item to a design in the 'In Process' status, until the publish of the selected row is completed you cannot perform other functions such as unrelate item, unrelate ECO or validate, on it. An error message "Cannot relate item for the design(s) that is being published to PD Cloud. Please try again when publish is completed." appears in the information area of Publish Workspace tab.

You cannot perform Publish in the Publish Workspace if a PD ECO is not related to a DFCO. An error message, "Cannot publish without a related PD ECO. Please relate a PD ECO first and then try again." appears in the information area of the Publish Workspace tab.

When the design(s) you select is in the process of being published from Agile PLM to PD Cloud, and you click the Un-incorporate button in the Affected Files tab, an error message, "The selected design(s) cannot be un-incorporated because it is being published to PD Cloud. Selected design(s) that is not being published will be un-incorporated." appears in the information area of the Publish Workspace tab.

For each design row on the Publish Workspace, the following information is available:

Has been checked out - A check mark is shown if the design is currently checked out

Incorporated - A lock symbol is shown if the design is currently incorporated

Folder Number - The number of the design.

Folder Description - The description of the design.

Folder Version - The version number of the design.

Item Number - When the design is related to an item in PD, this field is filled with the Item Number that links to the item in PD.

Item Description - The item's description.

Item Lifecycle Phase - The lifecycle state of the item.

Item Revision - The current revision of the item.

ECO Number - When the DFCO is related to an ECO in PD, the ECO Number that links to the ECO in PD appears in this field.

Job ID - Displays the ID- the same Job ID for designs that have same publish job.

Published By - Displays the name of the person who adds designs to Publish job

Status - This field displays status information, including the current publish status of the design or auto-relate status.

Details - Provides error messaging during validation and publishing.

Relating and Publishing Designs in Agile PLM to Items in PD Cloud

You can relate and publish a design in the CAD for Cloud system (that is, Agile PLM) to an engineering item in PD Cloud from a DFCO's Publish Workspace (PW). There are two ways to do this:

Manual relate by Search

Auto-relate by Publish (this function is an in-built subtask of Publish process)

Business Rules for Relating a PLM Design with a PD Item

Listed below are the rules for relating an Agile PLM design with a PD Cloud item:

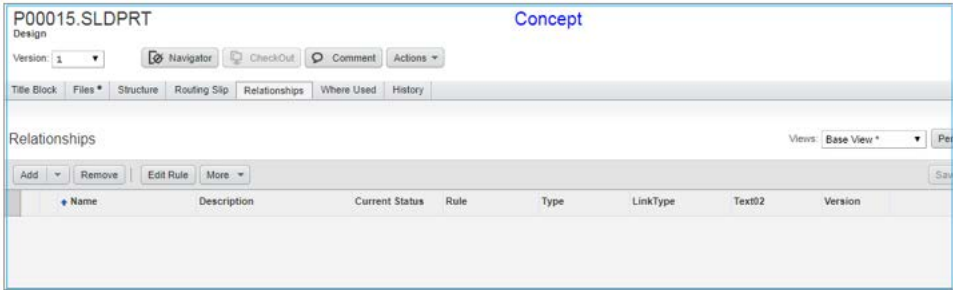
1. A Design (Model/Drawing) can relate to only one PD item, across all pending DFCOs.
2. An item can relate to only one Model and to its corresponding Drawing, across all pending DFCOs.
3. A Drawing and its associated Model can relate to a single item.
4. If configured (auto-relate enabled), a Drawing auto-relates to the item that its Model is related to (a Model's Item Number is automatically applied to its Drawing when the Drawings exclusive Item Number is not available).

For the CAD for Cloud integration to work seamlessly, a Design must have a CAD-specific relationship with a PD Item and similarly a DFCO must have a CAD specific relationship with a PD ECO on Cloud. The CAD specific relationship implies:

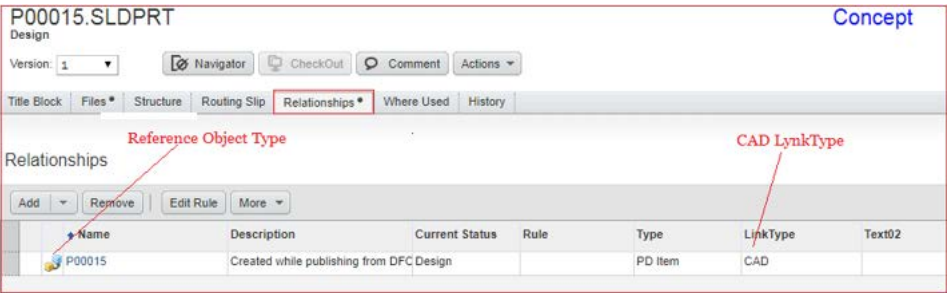
1. Creating a Reference Object for a remote Cloud PD business object that it is relating to
2. Adding the above created Reference Object to the local object's Relationship table
3. Updating the text attribute of the above Relationship row with value 'CAD' which is configured by the property `CadToCloud.LinkType`.

Note: Default setting '`CadToCloud.LinkType=Text01`' entry in `CloudIntegration.properties` can be customized with available text attribute on the Relationship tab only during the initial CAD for Cloud installation/integration. Once the system is in production, updating this property will break the functioning of the system.

This image shows a Design (P00015.SLDPRT) Relationship tab before relate function:



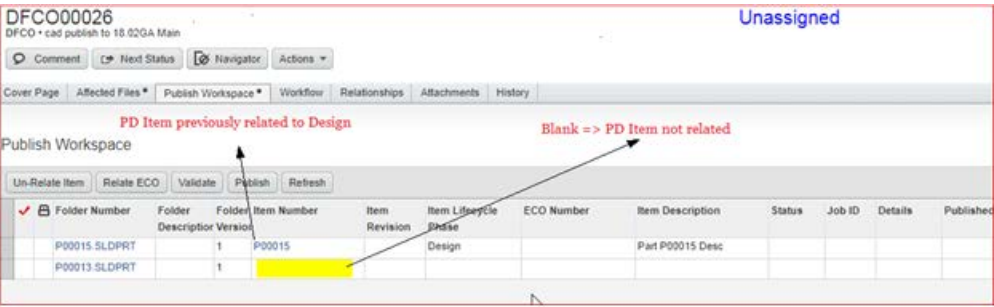
This image shows the Design (P00015.SLDPRT) after the relate function - CAD specific Relationship.



A design engineer would create new designs and/or modify existing ones by incrementing versions as necessary, to meet the day to day design requirements of engineering modeling. Designs that are newly created require establishing a relationship with a PD item, while the pre-existing designs may already be related.

As part of their business process, design engineers would add newly created and/or modified designs onto a DFCO's **Affected File** table. On the DFCO's **PW** table, all those designs that are previously related appear with already related PD item in the corresponding design's Item Number column, while the newly created ones are left with blank columns.

Below is an image showing the **PW** tab on DFCO (DFCO00026) showing one design (P00015.SLDPRT) previously related to item (P00015) and another design (P00013.SLDPRT) not related.



You can either keep the existing relationship and continue with the publish process or change with different items, as required. In such a case, you have to unrelate the existing relationship first, then relate to a different item.

Support NX CAD File Types

As stated early, CAD to Cloud integration only allows one model (part or assembly) to be related to a PD Item. A PD Item can have a Drawing and a model related but not two models.

This rule is dependent on being able to properly identify models and Drawings for each supported CAD system.

RUP16 provides the ability to switch the method used to identify Models and Drawings from the file extension to Design.p2.2010 (SubType).

Configuration changed in CloudIntegration.properties:

System Level configuration parameter - to use Design Subtype for Drawing

If this property is not set (or missing), it defaults to 'FALSE'. Its value can be one of TRUE/YES, FALSE/NO

Design will be considered to a Drawing if this is enabled and Design P2.TEXT04 has value of "Drawing"

USE_DESIGN_SUBTYPE_FOR_DRAWING =

Manual relate by search

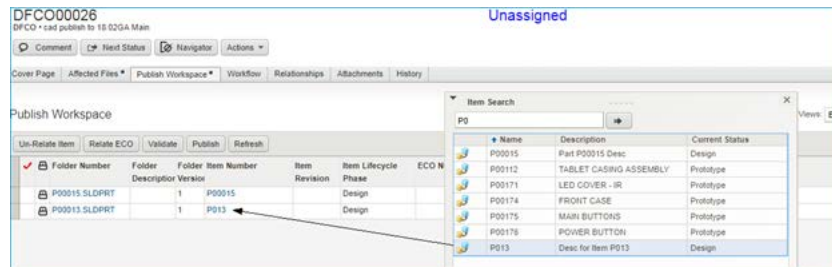
1. Open the DFCO that contains the design.
2. Go to the **Publish Workspace** tab.
3. Select the design part row to which you will relate a PD Item and then click **Relate Item**.

OR

Double-click in the empty cell of the **Item Number** column. The **Search** window appears.

Note: If the Design is already related to a PD Item, select the row, and click the **Unrelate** button. Once the Number cell is cleared, click again to bring up the **Search** window.

4. Enter the Search criteria in the search field and press **Enter**. Only Engineering Items are returned in the result set. Only the current released revision for each Item is listed in the search results.



Note: The Search of an item is controlled by Opt In "Leverage consistent item search across Product Management" with Manage Administrator Profile Values [Code: ORA_EGP_ITEM_SEARCH_CHAR_LIMIT]

The default search criteria is 3 characters when enabled.

5. Double-click to select the PD Item. Once a selection is made, a link to the related Item appears on the Publish Workspace.

The Item Search window remains open even after selection. You can continue to add more Item-Design relationships, but you **must** select the Item Number cell on **Publish**

Workspace first or Item selection will not work. Click the X, or press **Esc**, to close the Item Search window.

Note: The DFCO.PW.ItemRevision is blank until publish returns Completed Status. It is a new pending revision after adding/updating PD Item into PD ECO as Affected Item.

To open the related Item in PD, click the link in the Item Number column on the Publish Workspace. The Item opens in PD to the current revision.

Once a Design object is successfully related to a PD Item, the relationship between the two objects is included on the Relationships tab of the Design object.

Note: The design object is added to the PD Item's relationships table only after publish is complete.

As long as the user has the appropriate privileges, on the Design object's Relationships tab, the related PD Item's Item number, Description, and Current Status (Lifecycle Phase) fields are visible. The Link Type is updated to be CAD. Click the link in the Name column on the Publish Workspace to open the PD Item to the latest released revision. If the publish result is completed, the PD Item opens to the pending revision. If you open the PD Item from the Relationships tab it always opens the latest released revision of the related Item.

Once the DFCO is released, any relationships that were made between a Design and a PD Item remain in tact and are carried over into any subsequent DFCOs in which the Designs may be included.

Auto-relate by Publish

The CAD for Cloud system can perform Auto-relate for designs that need to be related to PD Items automatically, during the Publish process, as a sub-task.

To turn this feature on, you (here, the system administrator) must identify a suitable Design class' Page Two Text Attribute and use its 'API Name' to set the property 'DESIGN_ATTRIBUTE_FOR_PD_ITEM_NUMBER' in the CloudIntegration.properties file to store the PD Item number.

The DESIGN_ATTRIBUTE_FOR_PD_ITEM_NUMBER is set to TEXT01~TEXT25.

In general, customers upgrading from Agile PLM may already have configured a Page2 Text Attribute of Design class to store the 'Part Number'. You need to check if such an attribute is configured in the Java Client. If the configuration exists, use the API Name of that attribute. Otherwise, enable another text attribute on P2 and use its API Name.

The following is an example (you are free to choose other text attributes from TEXT01~TEXT25):

```
DESIGN_ATTRIBUTE_FOR_PD_ITEM_NUMBER = TEXT12
```

Note: This property can be configured/alterd any time during production but requires a system re-start to make it effective.

During the publish process, if the Auto-relate feature is enabled, the Designs selected for publish goes through the auto relate process in batches. The size of the batches depends on the value of the configurable property 'CLOUD_PD_BATCH_SIZE_FOR_BULK_API' in

CloudIntegration.properties. By default, this value is set to 25. As an administrator, you can change or tune its value based on the performance of the system.

During the auto-relate process, the following steps are executed for each Design in a batch:

1. Checks whether the design is related to the item or not. If it is already related it excludes the Design from the process. If not, it considers the design for the auto-relate process;
2. Reads the value for item number from the configured P2 Text Attribute on Design;
3. If the Item Number is available on P2.Text attribute, the Design auto-relate process continues;
4. If the Item Number is not available on P2.Text attribute and the Design is a Model, the process reports error 'Needs Related Item' and excludes the Design from the auto-relate process;
5. If the Item Number is not available on P2.Text attribute and the Design is a Drawing, the auto-relate process further pursues for the Item Number through its associated Model - the Model's Item Number is automatically applied to its Drawing when its exclusive Item Number is not provided. When unable to determine the Item Number, reports error 'Needs Related Item' and excludes the Design from the auto-relate process;
6. Validates the Item Number for the following business rules. Reports corresponding error and excludes it from auto-relate if they are not met:
 - a. If the Design is a Model, the same Item Number is not related to any other Models except to its associated Drawing, across pending DFCO's;
 - b. If the Design is a Drawing, the same Item Number is not related to any other Drawings except to its associated Model, across pending DFCO's.
7. Once all conditions are met, the auto-relate process ascertains the Item Number for Design and checks if that number exists in PD Cloud through web services;
8. If the Item is not found, reports error 'Mapped PD Item for Design does not exist in Cloud system' and excludes the Design from the auto-relate process;
9. If the Item is found, the auto-related process creates the corresponding reference object to PD Item, relates it to the corresponding Design and updates LinkType with the value 'CAD'.

If the Design fails during any of the above auto-relate process steps, the system reports corresponding error in the 'Details' column and sets 'Status' column to 'Failed'. The failed Design is completely excluded from further processing of Publish sub-tasks.

Auto-relating a Design Drawing during Publish

During the publish process, if the Design is a Drawing that has not yet been related to PD Item, but its associated Model is already related to an Item or set to auto-related, the system determines the Model's related Item Number as Drawing's Item Number to be related.

That is, the Model's Item Number automatically applies to its Drawing, if it is not provided with exclusive Item Number.

When unavailable or unable to determine, the system reports error 'Needs Related Item' and excludes such Drawings from the auto-relate process.

Auto-create by Publish

The CAD for Cloud system can automatically create PD Items that need to be related when they do not exist on Cloud during the Publish process as a subtask. The Auto-Create feature is designed as an internal function and is turned off, by default.

Note: The PD Cloud administrator must set the PD item UOM to "Each" to be able to auto-create PD item during publish from PLM to PD. Currently, this is the only supported UOM in CAD for Cloud's auto-create item function.

The prerequisite configurations or setup to turn on the auto-create feature are:

1. Enable the Auto-relate feature so that the Item Number to be created and related is ascertained:
 - a. Configure 'DESIGN_ATTRIBUTE_FOR_PD_ITEM_NUMBER' property to source Item Number;
2. Enable system level Auto-Create feature by setting 'CLOUD_CREATE_PD_ITEM_SUPPORTED' property value to 'TRUE'. Out-of-the-box, its default value is set to 'FALSE'.
3. Configure PD Item class so that the created Item belongs to it:
 - a. Create a list of Engineering PD Item's sub class names;
 - b. Associate the above list to a suitable Page2 List attribute on Design class;
 - c. Configure API name of above attribute to the 'DESIGN_ATTRIBUTE_FOR_PD_ITEM_CLASS' property in 'CloudIntegration.properties' file. The Auto-Create feature is not enabled if this property is blank or set to an invalid attribute;
4. Configure 'Create' Yes/No selection switch so that you can opt in/out of creating an Item at each Design object level:
 - a. Create a list with entries 'Yes' and 'No', if such a list does not already exist;
 - b. Associate above list to a suitable List Attribute on Page2 tab of Design class;
 - c. Configure API name of the above attribute to the 'DESIGN_ATTRIBUTE_FOR_CREATE_PD_ITEM_FLAG' property in 'CloudIntegration.properties' file. The Auto-Create feature is not enabled if this property is blank or set to an invalid attribute.

Listed below are a few examples:

```
DESIGN_ATTRIBUTE_FOR_PD_ITEM_NUMBER =  
DESIGN_ATTRIBUTE_FOR_PD_ITEM_CLASS =  
DESIGN_ATTRIBUTE_FOR_CREATE_PD_ITEM_FLAG =  
DESIGN_ATTRIBUTE_FOR_PD_ITEM_UOM =
```

Note: The above property settings can be altered any time during production, but require a system re-start to make it effective.

During the publish process, if the Auto-Create feature is enabled, the Designs selected for publish goes through the auto create process in batches. The size of the batches depends on the value of the configurable property 'CLOUD_PD_BATCH_SIZE_FOR_BULK_API' in CloudIntegration.properties. By default, its value is set to 25 but you (as an administrator) can change/tune its value based on the performance of the system.

The Auto Create process does the following additional tasks within Auto-relate for each Design in batch:

1. If the Item search process does not find the Item Number and the create Item option is set to blank or 'No' on the corresponding Design, the auto-create process reports error 'Mapped PD Item for Design does not exist in Cloud system' and excludes the Design from the auto-create/relate process;
2. If the Item search process does not find the Item Number and the create Item option is set to 'Yes' on the corresponding Design, the auto-create process includes the Item Number for bulk creation through PD Item web services.
3. All the Items meant for creation will be processed in batches using the Item Service bulk API.
4. The auto-relate process will continue for the Designs corresponding to successfully created Items. The failed items will report error 'Failed to create PD Item Object on Cloud PD' and be excluded from the auto-relate and publish process.

Note: Auto-Create reports error 'Create PD Item failed due to Invalid PD Item Class Name' if the PD Item Class is set to 'Root Item Class' or is left blank.

Rule-Generated Item Numbers

During Item class creation, the System administrator has to choose/configure a suitable Item Number generation method from the available methods, based on business requirement needs. Item Classes in Oracle Product Development Cloud allows you to configure the following types of Item Number generation methods:

Rule-Generated

Sequence-Generated

User-Defined

Product Management Rules are similar to pre-event tasks and they process attributes data and/or execute built-in or external functions including web services. When rules are configured and associated to an item class, they get executed during item object creation of such class. If a PD item class is configured to Rule-Generated, it takes Item Number from the associated Rule, which makes the system block direct input of Item Number through UI, Import, and Web Services. When Items of that class are created, value for the Item Number is read-only from Rule.

C4C Auto-Create feature supports only User-Defined and Rule Generated Item Number generation methods. So, Item classes meant for the purpose of auto-create from Designs should be configured to use only these two methods. Items created directly on PD system do not apply this restriction and they work well for Auto-relate or Search and Relate features.

You need an additional configuration or setup for a C4C integration instance, that allows you to create PD items from classes configured to use both User-Defined and Rule-Generated, or just Rule-Generated, Item Number generation method.

A C4C integration instance allows to create PD Items only from classes configured to use User-Defined Item Number generation method, no additional setup is required even though the remaining Item classes uses different Item Number generation methods.

Setting and Propagating Primary UOM in PD Items

A released DFCO in Agile PLM has a relationship with an ECO in PD Cloud, but the new DFCO does not carry the relationship with the ECO forward.

Design engineers can set, modify, correct, and propagate to PD Cloud the proper Unit of Measure (UOM) in their designs. This lets the CAD for Cloud publish process pull the selected value during item creation in PD Cloud.

The Agile administrator works with a new list called UOM Names, and a new property in CloudIntegration.properties called DESIGN_ATTRIBUTE_FOR_PD_ITEM_UOM. The administrator adds the value to the Page Two list being used for UOM in a design's attributes.

When setting DESIGN_ATTRIBUTE_FOR_PD_ITEM_UOM in CloudIntegration.properties, the administrator sets it to the API name of the list attribute as it appears in Java Client, for example, DESIGN_ATTRIBUTE_FOR_PD_ITEM_UOM=list20. The list values are added to the associated list for the list20 attribute.

A default value may be set in Java Client for commonly used UOMs. Design engineers can choose from the UOM list and set the value accordingly in each design.

Setting Design Revision to Push to Item Revision

The administrator wants to specify a Design Revision in Agile PLM to push to an Item Revision in PD Cloud. Herein, the word revision (or rev) is understood to be the revision of the item in PD Cloud; if that revision number was pushed to PD from a design in PLM, still, on the design side, it is always called design revision (or design rev).

So, more specifically, the administrator wants to specify the PD item's revision to be the same as the PLM design that is going to be published in PD using CAD for Cloud. To achieve this, a new property, PUSH_DESIGN_REVISION, is in the CloudIntegration.properties file.

If PUSH_DESIGN_REVISION is set to False, the existing behavior of the item - in this case, its revision - is copied to Design.Revision during Publish, whether or not the design's revision has been assigned a value.

If PUSH_DESIGN_REVISION is set to True, follow the table's first two rows below for proper configuration of pushing design revision to the item revision.

Table 3–1 Push Design Revision to Item Revision

Push Design Revision	Design.Rev before publish	PD.Item new Rev after publish	Design.Rev after publish
TRUE	No value	Keep item revision.	Update from the PD item's revision on the change (new rev).
TRUE	Has a value	Push design revision to Item revision.	Keep design revision.
FALSE	No value	Keep item revision.	Update from the PD item's revision on the change (new rev).
FALSE	Has a value	Keep item revision.	Update from the PD item's revision on the change (new rev).

Steps to configure Push design rev capability:

1. Open the configuration file CloudIntegration.properties to set PUSH_DESIGN_REVISION value to TRUE and save it back as below:

Keep the existing Item.Revision behavior if the property is set to FALSE.
 Design.Revision reflects the Item.Revision after a successful publish.

 # Push the value set in the Design Revision to the Item Revision if the attribute value is not blank, otherwise, the behavior is the same as setting this property to FALSE.

2. Restart the Agile Application Server.
3. Populate the values of the Revision attribute of those designs that are to be published.
4. Publish designs as usual in the DFCO PublishWorkspace.

If the publish operation fails, an error message appears in Publish Workspace Details column from PD Cloud.

Note: The design files are either uploaded to UCM or created as smart URL in the PD Item based on the properties values (PUBLISH_UPLOAD_FILETYPES and PUBLISH_UPLOAD_MAX_FILESIZE) set by the administrator in the CloudIntegration.properties. The uploaded files are added to the PD Item's attachment table based on revision. This allows for fine grain control on the file types and file size for uploading to UCM.

Unrelating a Design in Agile PLM from an Item in PD Cloud

You can remove the relation between a Design that was previously related to an Item.

To unrelate a PLM Design from a PD item:

1. Open the DFCO that contains the Design.
2. Go to the Publish Workspace tab.
3. Select the row(s) of the Design(s) that you want to unrelate from a PD item(s).
4. Click the unrelate Item button.

If the selected row had a related PD Item, the relationship is removed and all Item-related and Status information is cleared from the Publish Workspace. If, however, the relationship between the Design and the Item exists for a released revision, the relationship will not be removed.

You can also unrelate an Item from a design that has been published to PD. When the relationship is removed, the PD Item's information is cleared from the Publish Workspace and the relationship is no longer listed on the Relationships tab of the Design. After the unrelate action, the Design is deleted from the PD Item's Relationships tab. This however, will not remove previously published Item attachments or Item structures.

Relating a DFCO in Agile PLM to an ECO in PD Cloud

You can relate a DFCO in Agile PLM to an ECO in PD Cloud from the Publish Workspace tab on a DFCO's Affected Files tab.

To relate a DFCO in Agile PLM to an ECO in PD Cloud:

1. Open the DFCO.
2. Go to the Publish Workspace tab.
3. Click the Relate ECO button. The Relate ECO Search window appears. The Relate ECO button is grayed out, if there is no row in the Publish Workspace tab.
4. In the Search window, enter the search criteria and click Enter.
5. Double-click to select a pending ECO. The Search window closes. The ECO and the DFCO are now related. The relationship is applied to the entire DFCO and its related Items. The ECO Number is automatically filled in for every Design row on the Publish Workspace window.

To open the related ECO in PD, click the link in the ECO Number column on the Publish Workspace. The ECO is opened in PD. After the ECO is related, and before publish, only the

ECO is added onto the DFCO's Relationships tab. After publish, the DFCO is added into PD on the ECO's relationships table.

When the DFCO is Released or put on Hold, the relationships between the DFCO and ECO are maintained.

Unrelating a DFCO in Agile PLM to an ECO in PD Cloud

You can unrelate a DFCO in Agile to an ECO in PD from the Publish Workspace tab of a DFCO, only if the DFCO is related previously.

To unrelate a DFCO in Agile to an ECO in PD:

1. Open the DFCO.
2. Go to the Publish Workspace tab.
3. If the ECO is not related yet, click the Relate ECO button to relate it.
4. The Relate ECO button name changes to Unrelate ECO.
5. Click this button to unrelate the ECO.

You see that before Publish, when you relate a DFCO to an ECO in the Publish Workspace, the DFCO creates a relationship to the PD ECO. Click the Unrelate button to delete the existing Agile DFCO relationship.

After you relate a DFCO to a PD ECO and perform a Publish, both the Agile DFCO to PD ECO and the PD ECO to Agile DFCO relationships are created. BOM and Attachment redlines are also created. Clicking the Unrelate button deletes both the relationships. The unrelate function deletes only the relationships and not the BOM and Attachment redlines from the ECO.

If the DFCOs are in the Open, Draft, or Canceled status and you click the Unrelate ECO button, it will unrelate the ECO. If the ECO is released in PD, the error message: "ECO has been released and cannot be unrelated." appears in the information area of the Publish Workspace tab.

Validating the Publish Readiness of a Design

A design user can validate or check whether a design is ready to be published. To qualify for "Publish Ready" Status, a design must first have a related PD Item, a related pending ECO, and must be incorporated. If you fill in the Design P2 Text 12 attribute in PLM Web Client, you don't have to go to the publish workspace to search and relate the design to a PD item.

To validate a Design's publish readiness:

1. Open the DFCO.
2. Go to the Publish Workspace tab.
3. Select the row of the unpublished Design that you want to validate and click the Validate button.

Check the Details column for any corresponding error messages describing which criteria has not been met yet. If the design validates successfully, the Details column will contain the Publish Ready message.

Note: Configure the maximum number of change lines that are allowed in a change order is controlled by optin feature "Improve Change Order Data Loading with a Maximum Limit Setting for Affected Objects" with Manage Administrator Profile Values [Code:EGO_MAX_LINES_PER_CO]

Publishing a Design or Design Structure in Agile PLM to PD Cloud

You can publish native files, 2D, and 3D files to an Engineering Item in PD from the Publish Workspace. A publish results in the following:

Design files are published to the PD Item's Attachments tab as a smart URL under the related pending PD ECO revision.

The Design structure is published to the PD Item's BOM tab under the related pending PD ECO's revision

The Design is added into the PD Item's Relationships tab under the related pending PD ECO's revision

The PD Item is added into the PD Change's Affected Item's tab, if it is not already an Affected Item.

Redlined BOM/Attachment tabs are updated.

The DFCO is added into the PD Change's Relationships tab.

Note: Restarting the application moves pending publish jobs to the Completed status.

Note: The design engineer should not checkout or checkin the designs that are being published to Cloud PD.

The current publish status of a design is available in the Status and Details columns for the affected design.

If a design has not been manually related to a PD Item using the Publish Workspace, but a valid PD Item number has been assigned through the Page Two attribute, the Design is auto-related during the publish process.

If a design drawing has not been manually related to a PD Item using the Publish Workspace or the Page Two attribute, auto relation is attempted. See ["Auto-relating a Design Drawing during Publish"](#) for more information.

When publishing a design to PD, if the related PD Item is already on the related ECO's Affected Items table, once the publish process successfully completes, the Item Revision is updated on the Publish Workspace. Otherwise, if the related PD Item is not already on the related ECO's Affected Items table, the PD Item is added to the ECO's Affected Items as part of the publishing process. The Item Revision is then updated on the Publish Workspace.

To publish a Design to PD:

1. Open the DFCO.
2. Go to the Publish Workspace tab.
3. You can publish one or multiple rows to PD. Select the row(s) of the unpublished Design(s) that you want to publish to PD and click the Publish button.

The Status column reflects the status of the attempt to Publish. The following are the possible publish status options in the Status column:

Blank - design has not been published yet

In-Process - design is in the process of being published

Failed - design failed to publish to PD

Completed - design has been successfully published to PD

If the design was successfully published to PD, the Status column for the row states Completed. If the Status column contains a Failed message, the Details column has a corresponding error message describing why the failure occurred. You can also see status on the Title Block tab of the Design object itself.

Once a design is successfully published to PD, to open the pending revision of the Item in PD, click the link in the Item Number column on the Publish Workspace tab.

If you check out a design after it is published or failed a publish attempt, the Status is reset and returns to being blank. The Status changes are reflected in the Affected Files table, as well as the Publish Workspace.

Publishing while some Designs are in Process

The Design Engineer can publish all the related affected designs in the Publish Workspace, or a subset of the related Affected Designs, to PD Cloud. For those designs that are In Process, selecting them and clicking on the **Un-Relate Item**, **Validate**, or **Publish** buttons will result in an Error message that the action cannot be completed while the design is being published. For selecting design that is not being published, Un-Relate Item, Validate or Publish could be performed correctly.

Clicking on the **Un-Relate ECO** button while any publish is in progress will result in an Error message, and the ECO will not be unrelated.

Publishing CAD Design Structures

CAD structures can be successfully published, as well. During the publish process, if a selected Design row has a child component, and if the child has a related PD Item, the child is added as a structure member to the parent PD Item, which is related to the parent Design.

The PD Item structure attribute Source System is set as CAD to identify that this is a row controlled by CAD. The PD Item's structure attribute Quantity is also updated with the Quantity from the related child Designs.

For a structure that was originally created in PD, during a publish operation with a Quantity update, the Source field is set as CAD and CAD takes ownership of the published structure components.

Selective Publish with DESIGN_ATTRIBUTE_FOR_BOM_PUBLISH_OPTION

This property, when configured with an eligible Design Page Two list, lets the Design Engineer decide whether the PD item BOM structure is updated when publishing the design. If the value is set to Yes, then the PD item BOM structure is updated with the information from the design. If the value is set to No, then only the design's attributes and attachments are updated in the PD item, and the BOM structure will not be modified.

The Engineering Coordinator needs to establish the guideline on when the PD Item BOM structure should be updated based on the company's business requirements and communicate the guideline to the CAD Engineers.

This lists all possible configuration behaviors of the selective publish function:

1. Publish BOMs for all designs (and design structures) when this property does not exist in CloudIntegration.properties file.
2. Publish BOMs for all designs (and design structures) when this property exists in CloudIntegration.properties file but is left "blank" (List attribute no value).

3. Publish BOMs for all designs (and design structures) when this property is set to an invalid list or disabled list attribute.
4. Publish BOMs for all designs (and design structures) when this property is set to a valid list attribute, but no entries in list (empty list entries)
5. Publish BOMs for all designs (and design structures) when this property is set to a valid list attribute, with Yes and No as the value, but no default value set.
6. Don't Publish BOM when this property is set to a valid **List** attribute having valid entries, but the administrator choose No or it is default to No.
7. Publish BOM when this property is set to a valid **List** attribute having valid entries, and the administrator choose Yes or it is default to Yes.

Selective CAD File Types with PUBLISH_FILETYPES

With PUBLISH_FILETYPES added to CloudIntegration.properties, the administrator defines the specific file types of design files - or None or All - that can be added to PD Items as "smart URLs" in the **Publish** action.

If the design's extension matches the file types listed in PUBLISH_FILETYPES, then a "smart URL" of the design is published to the PD item's Attachments tab.

If the Agile administrator has left the PUBLISH_FILETYPES property Blank, no files will be uploaded as "smart URLs" to PD Items.

If there is no PUBLISH_FILETYPES property in CloudIntegration.properties (either your version of CAD for Cloud precedes ver. 9.3.6.9, or your administrator has removed this property from the properties file), all types of design files are uploaded as "smart URLs" to PD Items.

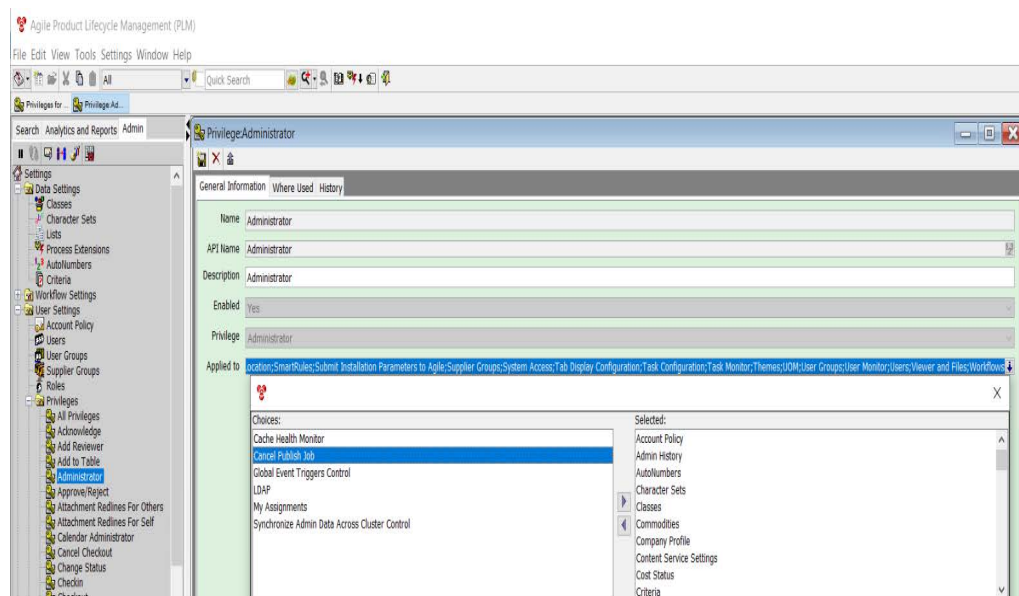
Administor with Admin Privilege can cancel a C4C Publish Job

User with Admin Privilege Can Cancel a C4C Publish Job

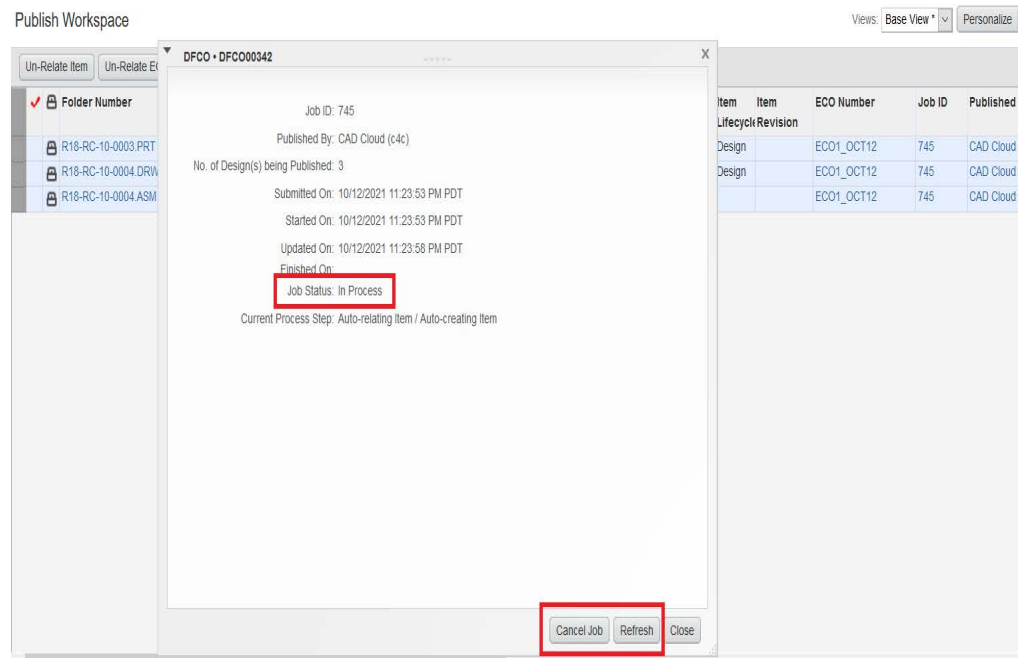
Prior to 9.3.6.14, if a C4C Publish Job became stuck "in process," the user had no way to clear the status and republish. 9.3.6.14 provides two ways to clear stuck C4C publish jobs.

Manual Cancel Publish Job

New Administrator applied to privilege added "Cancel Publish Job."



If a user has this new privilege, they will have access to new Cancel Job button in the Job Status pop-up window.



CLOUD_CAD_PUBLISH_THREAD_TIMEOUT

A new configuration parameter - CLOUD_CAD_PUBLISH_THREAD_TIMEOUT is introduced in CloudIntegration.properties file as following:

Timeout (minutes) for CAD Publish thread since last updated, default 120 minutes

CLOUD_CAD_PUBLISH_THREAD_TIMEOUT = 120

This timeout setting is not for the whole job but is for each publish step. For example, last updated time is recorded at the beginning of files uploaded to UCM and will be updated again when the upload is finished. A publish thread is considered to be stuck if there is no update for configured timeout. The Publish will be aborted and the Publish status will be to "Failed" in Publish Workspace with detailed message "C4C Publish is terminated due to being idle longer than the pre-configured time since last update (publish thread may be stuck)."

Using Save As to Create a New DFCO

In the CAD for Cloud integration, you can use the DFCO's Actions > Save As menu option on a Released or Canceled DFCO to create a new DFCO. When using Save As on a Released or Canceled DFCO, its Designs, and their relationships to Items in PD are carried over to the new DFCO. Since a Released DFCO has Designs that are also related to ECOs, note that the relationships between the Designs and ECOs are not carried over to the new DFCO. The Save As option does not work on Pending DFCOs.

Complete C4C Mapping Default Template File

This is a complete sample "C4C_Mapping.xml" default template file

C4C Mapping Sample File

Example A-1

```
<?xml version="1.0" encoding="utf-8"?>
<MappingConfig>
<MappingEnabled>>false</MappingEnabled>
<MappingEFFEnabled>>false</MappingEFFEnabled>
<MappingDFFEnabled>>false</MappingDFFEnabled>
<!-- output debugging information in WLS console window or stdout.log while
loading, parsing and validating configuration file if debug is enabled.
By default, debug is off. You can turn it on by setting its value to be true
-->
<DebugEnabled>>false</DebugEnabled>

<!--
This is the sample C4C_Mapping.xml that maps Design object's Title Block/Page Two
attribute to PD Item EFF attributes
In this sample configuration, following assumptions are made (make sure both Name
and API name being the same in EFF Context and Attribute configuration):
1. CADItem1/CADItem1 - subclass name of Root Item Class and its internal API name
in Fusion;
1.1 CADItem1S1/CADItem1S1 - subclass name of CADItem1 Class and its internal API
name in Fusion;
2. "CADItemContext1DisplayName" / "CADItemContext1Code" / "Caditemcontext1APIName"
- Display name / Code / API name for a single row EFF attribute Context (ie.
Attribute Group)
2.1 "CCCADItemCtx1MultiList01" / "CCCADItemCtx1MultiList01" /
"Cccaditemctx1multilist01" - Display name / Code / API name for a multi rows EFF
attribute Context (ie. Attribute Group)
2.2 "CADItem1Cxt1" / "CADItem1Cxt1" / "Caditem1Cxt1" - Display name / Code / API
name for a single row EFF attribute Context (ie. Attribute Group)
2.3 No attribute context of its own (add one or more if needed)
3. EFF Attributes defined in Caditemcontext1APIName Context with Name/Code/API
Name (being configured its Behavior as "Single Row")as followings:
CADAttr1Name & CADAttr2Name / CADAttr1Code & CADAttr2Code / cadattr1APIName &
cadattr2APIName
3.1 Add EFF single selected list support and configure a single list EFF attribute
as "CCItemCheckoutStatus", "CCItemCheckoutStatus", "ccitemcheckoutstatus" with
values of 'Checked In' and 'Checked Out' which are identical as its counter part
in Agile
```

```

3.2 Add EFF multi selected list support, and configur a multi list EFF attribute
as "CCMultiList01", "CCMultiList01", "ccmultilist01" with multi list value of
'CCA', 'CCB',... and 'CCG' which are identical
as its counter part in Agile.
3.3 EFF Attributes defined in Caditemlcxt1 Context with Name/Code/API Name (being
configured its Behavior as "Single Row")as followings:
strAttr1 & numAttr2 / strAttr1 & numAttr2 / strattr1 & numttr2
4. EFF Context Caditemcontext1APIName/Cccaditemctx1multilist01 are associated with
ROOT ITEM class and CADitem1Cxt1 is associated
with CADitem1 class and their attributes are exposed in Product Development (PD)
5. Use wsimport to generate 3 web service Java stub jar files; Ex: Item EFF
Category and attribute context - JItemRootIccPrivate/Caditem1 &
Caditemcontext1APIName/Cccaditemctx1multilist01/Caditemlcxt1
-->

<!--
The values of "classinternalapiname ", "categorytype ", "contexttype " and "name"
in PDItemClass, CategoryContext and PDItemAttributes configuration
in C4C_Mapping.xml sample files should be the exact values of corresponding "API
name" (case sensitive)
For Root Item class, its values are "ROOT_ICC", "JItemRootIccPrivate",
"Caditemcontext1APIName"/"Cccaditemctx1multilist01", "Root Item Class"
-->
<!--
<PDItemClass classname = "Root Item Class" classinternalapiname = "ROOT_ICC" uid =
"rtclz">
<CategoryContext categorytype = "JItemRootIccPrivate" contexttype =
"Caditemcontext1APIName" uid = "rtcc.caditemcontext1apiname" />
<CategoryContext categorytype = "JItemRootIccPrivate" contexttype =
"Cccaditemctx1multilist01" uid = "rtcc.cccaditemctx1multilist01" />
</PDItemClass>
<PDItemClass classname = "CADitem1" classinternalapiname = "CADitem1" uid =
"caditem1clz">
<CategoryContext categorytype = "Caditem1" contexttype = "Caditemcontext1APIName"
uid = "caditem1.caditemcontext1apiname" />
<CategoryContext categorytype = "Caditem1" contexttype =
"Cccaditemctx1multilist01" uid = "caditem1.cccaditemctx1multilist01" />
<CategoryContext categorytype = "Caditem1" contexttype = "Caditem1Cxt1" uid =
"caditem1.caditemlcxt1" />
</PDItemClass>
<PDItemClass classname = "CADitem1S1" classinternalapiname = "CADitem1S1" uid =
"caditem1s1clz">
<CategoryContext categorytype = "Caditem1S1" contexttype =
"Caditemcontext1APIName" uid = "caditem1s1.caditemcontext1apiname" />
<CategoryContext categorytype = "Caditem1S1" contexttype = "Caditem1Cxt1" uid =
"caditem1s1.caditemlcxt1" />
</PDItemClass>

<DesignClass subclassname = "C4cDesign" uid ="c4cdesign">
<Table tablename="TitleBlock" uid="tb" />
<Table tablename="PageTwo" uid="p2" />
</DesignClass>
<DesignClass subclassname = "Design" uid ="design">
<Table tablename="TitleBlock" uid="tb" />
<Table tablename="PageTwo" uid="p2" />
</DesignClass>

<PDItemAttributes>
<PDItemAttribute uid="rtclz.rtcc.caditemcontext1apiname.cadattr1apiname"
type="String" name="cadattr1APIName" eff="true" />

```

```

<PDItemAttribute uid="rtclz.rtcc.caditemcontextlapiname.cadattr2apiname"
type="Numeric" name="cadattr2APIName" eff="true" />
<PDItemAttribute uid="rtclz.rtcc.caditemcontextlapiname.ccitemcheckoutstatus"
type="SingleList" name="ccitemcheckoutstatus" eff="true" />
<PDItemAttribute uid="rtclz.rtcc.cccaditemctxlmultilist01.ccmultilist01"
type="MultiList" name="ccmultilist01" eff="true" />
<PDItemAttribute uid="caditemlclz.caditeml.caditemcontextlapiname.cadattrlapiname"
type="String" name="cadattr1APIName" eff="true" />
<PDItemAttribute uid="caditemlclz.caditeml.caditemcontextlapiname.cadattr2apiname"
type="Numeric" name="cadattr2APIName" eff="true" />
<PDItemAttribute
uid="caditemlclz.caditeml.caditemcontextlapiname.ccitemcheckoutstatus"
type="SingleList" name="ccitemcheckoutstatus" eff="true" />
<PDItemAttribute uid="caditemlclz.caditeml.cccaditemctxlmultilist01.ccmultilist01"
type="MultiList" name="ccmultilist01" eff="true" />
<PDItemAttribute uid="caditemlclz.caditeml.caditemlcxtl.strattr1" type="String"
name="strattr1" eff="true" />
<PDItemAttribute uid="caditemlclz.caditeml.caditemlcxtl.numattr1" type="Numeric"
name="numattr1" eff="true" />
<PDItemAttribute
uid="caditemlslclz.caditemlsl.caditemcontextlapiname.cadattrlapiname"
type="String" name="cadattr1APIName" eff="true" />
<PDItemAttribute
uid="caditemlslclz.caditemlsl.caditemcontextlapiname.cadattr2apiname"
type="Numeric" name="cadattr2APIName" eff="true" />
<PDItemAttribute uid="caditemlslclz.caditemlsl.caditemlcxtl.strattr1"
type="String" name="strattr1" eff="true" />
</PDItemAttributes>

<DesignAttributes>
<DesignAttribute uid="c4cdesign.tb.description" type="String" name="description"
/>
<DesignAttribute uid="c4cdesign.tb.revision" type="String" name="revision" />
<DesignAttribute uid="c4cdesign.tb.checkoutstatus" type="SingleList"
name="checkoutStatus" />
<DesignAttribute uid="c4cdesign.tb.number" type="String" name="number" />
<DesignAttribute uid="c4cdesign.tb.label" type="String" name="label" />
<DesignAttribute uid="c4cdesign.tb.publishedchangeorder" type="String"
name="publishedChangeOrder" />
<DesignAttribute uid="c4cdesign.tb.newchangesequences" type="String"
name="newChangeSequence" />
<DesignAttribute uid="c4cdesign.p2.numeric06" type="Numeric" name="numeric06" />
<DesignAttribute uid="c4cdesign.p2.numeric07" type="Numeric" name="numeric07" />
<DesignAttribute uid="design.p2.numeric06" type="Numeric" name="numeric06" />
<DesignAttribute uid="c4cdesign.p2.multilist01" type="MultiList"
name="multiList01" />
<DesignAttribute uid="design.tb.description" type="String" name="description" />
</DesignAttributes>

<MappingEntry type = "RedlineEFF" name = "redlineeff">
<AttributeMap designattribute = "c4cdesign.tb.description" pditemattribute =
"rtclz.rtcc.caditemcontextlapiname.cadattrlapiname" />
<AttributeMap designattribute = "c4cdesign.p2.numeric06" pditemattribute =
"rtclz.rtcc.caditemcontextlapiname.cadattr2apiname" />
<AttributeMap designattribute = "c4cdesign.tb.checkoutstatus" pditemattribute =
"rtclz.rtcc.caditemcontextlapiname.ccitemcheckoutstatus" />
<AttributeMap designattribute = "c4cdesign.p2.multilist01" pditemattribute =
"rtclz.rtcc.cccaditemctxlmultilist01.ccmultilist01" />
<AttributeMap designattribute = "c4cdesign.tb.description" pditemattribute =
"caditemlclz.caditeml.caditemcontextlapiname.cadattrlapiname" />

```

```

<AttributeMap designattribute = "c4cdesign.p2.numeric06" pditemattribute =
"caditemlclz.caditeml.caditemcontextlapiname.cadattr2apiname" />
<AttributeMap designattribute = "c4cdesign.tb.checkoutstatus" pditemattribute =
"caditemlclz.caditeml.caditemcontextlapiname.ccitemcheckoutstatus" />
<AttributeMap designattribute = "c4cdesign.p2.multilist01" pditemattribute =
"caditemlclz.caditeml.cccaditemctxlmultilist01.ccmultilist01" />
<AttributeMap designattribute = "c4cdesign.tb.revision" pditemattribute =
"caditemlclz.caditeml.caditemlctxl.strattr1" />
<AttributeMap designattribute = "c4cdesign.p2.numeric07" pditemattribute =
"caditemlclz.caditeml.caditemlctxl.numattr1" />
<AttributeMap designattribute = "c4cdesign.tb.description" pditemattribute =
"caditemlslclz.caditemlsl.caditemcontextlapiname.cadattrlapiname" />
<AttributeMap designattribute = "c4cdesign.p2.numeric06" pditemattribute =
"caditemlslclz.caditemlsl.caditemcontextlapiname.cadattr2apiname" />
<AttributeMap designattribute = "c4cdesign.tb.revision" pditemattribute =
"caditemlslclz.caditemlsl.caditemlctxl.strattr1" />
</MappingEntry>
-->
<!--
This is the sample C4C_Mapping.xml that maps DFCO object's CoverPage/Page Two
attribute to PD Change DFF attributes
In this sample configuration, following assumptions are made (make sure both Name
and API name being the same in DFF Context and Attribute configuration):
1. C4C_PD_CO_TYPE/C4C_PD_CO_TYPE - name of Change Order Type and its internal API
name in Fusion;
2. C4C_GS_ATTR_TEXT1/C4C_GS_ATTR_TEXT1/c4cGsAttrText1 - name / Code / API name for
a DFF attribute in Global Global Segments;
3. C4C_PD_CO_TYPE/C4C_PD_CO_TYPE/C4cPdCoType - name / Code / API name for a
Context in Context Sensitive Segments for Change Order Type - C4C_PD_CO_TYPE (be
sure to have the code as Change Order Type)
3.1 "C4C_PD_DFF_Str1" / "C4C_PD_DFF_Str1" / "c4cPdDffStr1" - name / Code / API
name for a String DFF attribute in Context - C4C_PD_CO_TYPE
3.2 "C4C_PD_DFF_Str1_Req" / "C4C_PD_DFF_Str1_Req" / "c4cPdDffStr1Req" - name /
Code / API name for a String DFF attribute as Required in Context - C4C_PD_CO_TYPE
3.3 "C4C_PD_DFF_Num1" / "C4C_PD_DFF_Num1" / "c4cPdDffNum1" - name / Code / API
name for a Numeric DFF attribute in Context - C4C_PD_CO_TYPE
3.4 "C4C_PD_DFF_SL1" / "C4C_PD_DFF_SL1" / "c4cPdDffSL1" - name / Code / API name
for a SingleList DFF attribute with its value as Numeric in Context - C4C_PD_CO_
TYPE
3.5 "C4C_PD_DFF_STRSL1" / "C4C_PD_DFF_STRSL1" / "c4cPdDffStrsl1" - name / Code /
API name for a SingleList DFF attribute with its value as String in Context - C4C_
PD_CO_TYPE

Use contexttype of "ChangeObjectDFF" if there is no Change object Context
Sensitive Segment being configured in PD
Subclassname is Agile Change Order subclass API name, and their tablename are
CoverPage or PageTwo
-->
<!--
<PDChangeClass classname = "C4C_PD_CO_TYPE" classinternalapiname = "C4C_PD_CO_
TYPE" uid = "c4cpdcotypeclz">
<ContextSegmentDFF contexttype = "C4cPdCoType" uid = "ctxdff" />
</PDChangeClass>

<ChangeClass subclassname = "DFCO" uid="dfco">
<Table tablename="CoverPage" uid="cp" />
<Table tablename="PageTwo" uid="p2" />
</ChangeClass>

<PDChangeDFFAttributes>

```

```

<PDChangeDFFAttribute uid="c4cpdcotypeclz.ctxdff.c4cgsattrtext1" type="String"
name="c4cGsAttrText1" dff="true" />
<PDChangeDFFAttribute uid="c4cpdcotypeclz.ctxdff.c4cpddffstrlreq" type="String"
name="c4cPdDffStrlReq" dff="true" />
<PDChangeDFFAttribute uid="c4cpdcotypeclz.ctxdff.c4cpddffstr1" type="String"
name="c4cPdDffStr1" dff="true" />
<PDChangeDFFAttribute uid="c4cpdcotypeclz.ctxdff.c4cpddffnum1" type="Numeric"
name="c4cPdDffNum1" dff="true" />
<PDChangeDFFAttribute uid="c4cpdcotypeclz.ctxdff.c4cpddffsl1" type="SingleList"
name="c4cPdDffSl1" dff="true" />
<PDChangeDFFAttribute uid="c4cpdcotypeclz.ctxdff.c4cpddffstrsl1" type="SingleList"
name="c4cPdDffStrsl1" dff="true" />
</PDChangeDFFAttributes>

<ChangeAttributes>
<ChangeAttribute uid="dfco.cp.number" type="String" name="number" />
<ChangeAttribute uid="dfco.cp.changetype" type="SingleList" name="changeType" />
<ChangeAttribute uid="dfco.p2.text01" type="String" name="text01" />
<ChangeAttribute uid="dfco.p2.numeric07" type="Numeric" name="numeric07" />
<ChangeAttribute uid="dfco.p2.numeric06" type="Numeric" name="numeric06" />
<ChangeAttribute uid="dfco.p2.list11" type="SingleList" name="list11" />
<ChangeAttribute uid="dfco.p2.list12" type="SingleList" name="list12" />
</ChangeAttributes>

<MappingEntry type = "MapChangeDFF" name = "changedffmap">
<AttributeMap changeattribute = "dfco.cp.number" pdchangedffattribute =
"c4cpdcotypeclz.ctxdff.c4cgsattrtext1" />
<AttributeMap changeattribute = "dfco.p2.text01" pdchangedffattribute =
"c4cpdcotypeclz.ctxdff.c4cpddffstr1" />
<AttributeMap changeattribute = "dfco.p2.numeric06" pdchangedffattribute =
"c4cpdcotypeclz.ctxdff.c4cpddffnum1" />
<AttributeMap changeattribute = "dfco.p2.list11" pdchangedffattribute =
"c4cpdcotypeclz.ctxdff.c4cpddffstrsl1" />
<AttributeMap changeattribute = "dfco.p2.list12" pdchangedffattribute =
"c4cpdcotypeclz.ctxdff.c4cpddffsl1" />
</MappingEntry>
-->
</MappingConfig>

```

