

Agile Product Lifecycle Management

EC Solution Pack User's Guide

V1.0



Part No. E42115-02
December 2014

Copyright and Trademarks

Copyright © 1995, 2014, Oracle and/or its affiliates. All rights reserved.

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 software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, 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 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle and Java are registered trademarks of Oracle Corporation 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 on 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. 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. The RMW product includes software developed by the Visigoth Software Society.

CONTENTS

Copyright and Trademarks	2
EC Solution Pack Overview	7
Design to Item Mapping	7
Item to Design Mapping	7
Design to Design Mapping	7
Change Mapping.....	8
Design Revision Control.....	8
Enabling the EC Solution Pack with the Java Client	8
Configuring the Property File.....	11
Defining Source and Target Attributes	11
Defining Relationships	12
Item to Design.....	12
Design to Item.....	12
Configuring Criteria	12
Configuring Mapping	13
Configuring Events.....	15
Configuring Design Revision Control	16

Preface

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) <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.

Note To read the PDF files, you must use the free Adobe Acrobat Reader version 9.0 or later. This program can be downloaded from the [Adobe web site](http://www.adobe.com) <http://www.adobe.com>.

The [Oracle Technology Network \(OTN\) web site](http://www.oracle.com/technetwork/documentation/agile-085940.html) <http://www.oracle.com/technetwork/documentation/agile-085940.html> can be accessed through Help > Manuals in both Agile web Client and Agile Java Client. If you need additional assistance or information, please contact My Oracle Support (<https://support.oracle.com>) for assistance.

Note Oracle customers 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. Before calling Oracle Support about a problem with an Agile PLM manual, please have the full part number, which is located on the title page.

TTY Access to Oracle Support Services

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>. Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Readme

Any last-minute information about Agile PLM can be found in the Readme file on the [Oracle Technology Network \(OTN\) web site](http://www.oracle.com/technetwork/documentation/agile-085940.html) <http://www.oracle.com/technetwork/documentation/agile-085940.html>.

Agile Training Aids

Go to the [Oracle University web page](http://www.oracle.com/education/chooser/selectcountry_new.html) http://www.oracle.com/education/chooser/selectcountry_new.html for more information on Agile Training offerings.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

This documentation may contain links to Web sites of other companies or organizations that Oracle

does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these web sites.

EC Solution Pack Overview

This chapter includes the following:

▪ Design to Item Mapping	7
▪ Item to Design Mapping	7
▪ Design to Design Mapping	7
▪ Change Mapping	8
▪ Design Revision Control.....	8

The Agile EC Solution Pack allows Agile PLM customers to automate the following business processes across the entire PLM suite:

- Mapping Design attributes to Item attributes
- Mapping Item attributes to Design attributes
- Mapping Design attributes to Design attributes
- Change Mapping
- Design Revision Control

Using Java Process Extensions deployed on the Agile application server, a customer can use the EC Solution pack to perform these specific functions through event management in the Agile PLM Java Client or by configuring an XML file and the supporting events.

Design to Item Mapping

With the EC Solution pack, you can automatically copy values entered on a Design object to a Part object. You can map Design attributes to change controlled Item attributes. You can also map different attribute types, including numeric attributes.

These events are triggered by updates to the Design Title Block, Page Two and Page Three, Check In, Check Out, and Relationships.

Item to Design Mapping

You can automatically copy values entered on a Part object to the related Design object. You can map different attribute types, like Money to Text, as well as change controlled Item attributes to Design attributes.

These events are triggered by updates to the Item Title Block, Page Two and Page Three, Attachments, or Relationships.

Design to Design Mapping

You can automatically copy values from a Model design to the parent Drawing design object and from a Drawing design to a child Model design which keeps information consistent between related

Models and Drawings.

These events are triggered by updates to the Design Title Block, Check In, or Check Out.

Change Mapping

Change mapping starts with a Change object and allows mapping of Change or related Item attributes to Design objects. The supported event triggers are Change Status and Change Affected Items update.

Design Revision Control

After check in or check out of a Design, the revision number of the Design and its related Items is automatically increased.

Enabling the EC Solution Pack with the Java Client

You can enable the EC Solution Pack through the event management framework of the Agile PLM Java Client. Not all events are supported through the solution pack, but you can create events for those that are supported.

The basic steps to enable the EC Solution Pack through the Java Client are as follows:

1. Create an event.
2. Create the corresponding event handler from the event action.

There are five event actions supported by the EC Solution Pack:

1. **Com.agile.ec.ChangeMapping:** Starts with a Change object and allows mapping change or related item attributes to Design objects. The supported event triggers are Change Status and Change affected items update.
2. **Com.agile.ec.ItemMapping:** Starts with an Item object and allows mapping to item and pending changes attributes to Design objects. The supported event triggers are Item Title Block, Page Two, Page Three update or redline, Attachment, and Relationships tab update.
3. **Com.agile.ec.DesignMapping:** Starts with a Design object and allows mapping of Design attributes to Item objects or its related Change objects. The supported event triggers are Design Title Block, Page Two and Page Three update, Check In, Check Out, and Design Relationships tab update.
4. **Com.agile.ec.DesignSync:** Starts with a Design object and allows sync attributes between Design objects. This helps to keep information consistent between this Design and its parents or children. The supported event triggers are Design Title Block update, Check In, and Check Out.
5. **Com.agile.ec.SyncRevision:** Starts with a Design object and allows Design Revision to be tied to Workflow Released status change. The supported event triggers are Design Check In, Design Check Out, and Workflow status change.

3. Create a subscriber to bind the handler to an event. The EC Solution pack does not support configuring Pre-events or Asynchronous events.

After configuring the event, start the action from the Web Client, Java Client, SDK, or EC connector.

For example, when updating the title block of an Item, you want to copy the Item description to a Design. When you update the Item title block in the Web Client and after the edit action is complete, the event you created is run.

Configuring the Property File

This chapter includes the following:

▪ Defining Source and Target Attributes	11
▪ Defining Relationships	12
▪ Configuring Criteria	12
▪ Configuring Mapping	13
▪ Configuring Events	15
▪ Configuring Design Revision Control	16

The XML file maintains all of the information related to attribute mapping. After configuration, save the file in a Design or File Folder object named MCAD_SOLUTION_PACK that can be checked in and out.

The configuration property file, ec-solution-mapping-config.xml, is used to define the rules for mapping actions and is stored in MCAD_SOLUTION_PACK. With the configuration file, you can:

- Identify the attribute to map from and to (source and target)
- Identify which trigger event applies to mapping
- Allow additional criteria to control mapping
- Allow string modification of attribute values

The following XML example shows how attributes are mapped from Item to Design when a change is released:

```
<MappingEntry type="changeMapping" name = "changeStatusMapping">
  <EventTrigger eventType="changeStatus" workFlow="Default Change
Orders" statusFrom="CCB" statusTo="Released" />
  <AttributeMapGroup>
    <AttributeMap type = "setValue">
      <SourceValue>[Change.Affected Items.Lifecycle Phase]
Rev [Change.Affected Items.New Rev] </SourceValue>
      <TargetAttribute>Design.Title
Block.Label</TargetAttribute>
    </AttributeMap>
  </AttributeMapGroup>
</MappingEntry>
```

Defining Source and Target Attributes

The source value specifies from where you want to copy to the target attribute. It can be dynamic, constant, or a combination of both. A dynamic value is flagged [Type.Table.Attribute Name], such as <SourceValue>[Change.Affected Items.Lifecycle Phase] Rev [Change.Affected Items.New Rev]</SourceValue>.

The target attribute specifies the attribute you are copying to. The target attribute is also flagged

Type.Table.Attribute Name, such as <TargetAttribute>Item.Page Two.Design Latest Version</TargetAttribute>.

Defining Relationships

The first step in configuring the property file is to define the relationships which will apply to all of the subsequent mapping actions. You can define an Item to Design or Design to Item relationship.

Item to Design

Items with Related Designs will have a Relationship connection. You can define a criteria within the element `RelatedDesignCriteria` to configure this relationship. The following example shows a related design from item relationship where the attribute on the Item Relationships tab Link type is equal to CAX-ITEM:EMPTY.

```
<RelatedDesignCriteria>
  <Condition attributeName = 'Item.Relationships.Link Type'
operator='equal to'>CAX-ITEM:EMPTY</Condition>
</RelatedDesignCriteria>
```

Design to Item

Designs with Related Items have a Relationship connection. You can define a criteria within the element `RelatedItemCriteria` to configure this relationship. The following example shows a related items from design relationship where the attribute on the Design Relationships tab Link type is equal to CAX-ITEM:EMPTY:

```
<RelatedItemCriteria>
  <Condition attributeName = 'Design.Relationships.Link Type'
operator='equal to'>CAX-ITEM:EMPTY</Condition>
</RelatedItemCriteria>
```

Configuring Criteria

Criteria is a global setting that defines how to find the related items from a specific design, or the related designs from a specific item. The criteria must be either in a `Condition` or `ConditionGroup` node. the condition is a final element which cannot include any child nodes. The `ConditionGroup` is a collection of `Condition` or `ConditionGroup` nodes combined with `Or` or `And` operators. The following criteria example specifies that the Change Cover Page Number is not null:

```
<Condition attributeName = "Change.Cover Page.Number" operator="is not
null"></Condition>
```

An example of a combination of multiple conditions specifying a change whose part category is not null and a lifecycle phase not equal to Inactive:

```
<ConditionGroup join = "and">
  <Condition attributeName = "Change.Cover Page.Part Category"
operator=" is not null "> </Condition>
  <Condition attributeName = "Change.Cover Page.Lifecycle Phase "
operator=" not equal to "> Inactive </Condition>
</ConditionGroup>
```

An example of a ConditionGroup that includes one ConditionGroup and one Condition:

```
<ConditionGroup join = "or">
  <ConditionGroup join = "and">
    <Condition attributeName = "Change.Cover Page.Part Category"
operator=" is not null "> </Condition>
    <Condition attributeName = "Change.Cover Page.Lifecycle
Phase " operator=" not equal to "> Inactive </Condition>
  </ConditionGroup>
  <Condition attributeName = "Change.Page Two.Notes" operator="start
with">TNS</Condition>
</ConditionGroup>
```

The attribute name of Condition specifies the left value of the condition which is read from the Agile data object. The value of this Condition can be constant or dynamic.

The following table contains a list of supported operators for each attribute type:

Attribute Type	Operator
String	is null, is not null, equal to, not equal to, contains, not contains, start with, does not start with, end with, does not end with
Date	is null, is not null, equal to, not equal to, greater than, greater than or equal to, less than, less than or equal to
Money	is null, is not null, equal to, not equal to, greater than, greater than or equal to, less than, less than or equal to
Numeric (Integer/Double)	is null, is not null, equal to, not equal to, greater than, greater than or equal to, less than, less than or equal to
List/Multilist	is null, is not null, equal to, not equal to, in, not in

Configuring Mapping

The mapping configuration tells what the source value and target attribute are. The element MappingEntries is a collection of MappingEntry elements, so you can define more than one mapping entry. The available values of attribute type for MappingEntry are as follows:

1. changeMapping

Change mapping starts with a Change object and allows mapping of Change or related Item attributes to Design objects. The supported event triggers are Change Status and Change Affected Items update.

You can copy attributes between Changes (Cover Page, Page Two, and Page Three), Change Affected Items rows, Items (Title Block, Page Two and Page Three, or the Redline tab), Item Relationship rows, and Design Objects.

2. itemMapping

Item mapping starts with an Item object and allows the mapping of Item and pending changes attributes to Design objects. The supported event triggers are Item Title Block, Page Two, Page Three update or redline, Attachment and Relationships tab update.

When you are updating or creating an Item Title Block, you can map attributes between Items, Pending Changes, Change Affected Items rows, Item Relationship rows, and Design objects.

When you are updating the Item Attachment tab, you can map attributes between Pending Changes, Change Affected Items rows, Items, Attachment rows, and the current File Folder.

When you are updating the item Relationship tab, you can map attributes between Pending Changes, Change Affected Item rows, Items, Relationships and the Design object.

You can configure a designMapping post-event of event type "Checkin" on a Design object in addition to (or instead of) the event-type "Update". See the designMapping attribute for an example on how to ensure Item attribute values are mapped to the Design object even for first time saved objects.

3. designMapping

Design mapping allows you to automatically copy from a Design object to an Item object or Pending Change. The supported event triggers are Design Title Block, Page Two and Page Three update, Design Create, Check-in, Check-out and Design Relationships tab update.

When updating a Design Title Block, creating a Design or checking in or checking out a Design, you can map attributes between Designs, Design Relationships, Items, Pending Changes, and Change Affected Item rows.

When updating a Design Relationship tab, you can map attributes between the current Design, Design Relationship rows, Items, Pending Changes and Change Affected Items rows.

You can configure a designMapping post-event of event type "Checkin" on a Design object in addition to (or instead of) the event-type "Update". For example, you can ensure Design attribute values are mapped to the Design object even for first time saved objects with the following Mapping Entry:

```
<MappingEntry type="designMapping" name = "mapping Design related attribute to Item">
  <EventTrigger eventType="checkin" objectType="Designs" tableName="Title Block" />
  <AttributeMapGroup>
    <AttributeMap type = "setValue">
      <SourceValue>[Design.Title Block.Description]</SourceValue>
      <TargetAttribute>Item.Title Block.description</TargetAttribute>
    </AttributeMap>
  </AttributeMapGroup>
</MappingEntry>
```

4. designSync

Design Sync mapping starts with a Design object and allows sync attributes between Design objects which helps to keep information consistent between this design and its parents or children. The supported event triggers are Design Title Block update, Check-in and Check-out.

When updating a Design Title Block, creating a Design or checking in or checking out a Design, you can sync attributes between the current Design, parent Design, or child Design.

You can configure a designSync mapping post-event of event type "Checkin" on a Design

object in addition to (or instead of) the event-type "Update". For example, you can ensure attributes are synchronized between parent and child even for first time saved objects like this:

```
<MappingEntry type="designSync" name="designSyncCase">
    <EventTrigger eventType="checkIn" objectType="Designs" tableName="Title Block" />

    <AttributeMapGroup>
        <AttributeMap type="setValue">
            <SourceValue>[Design.Title Block.Description]</SourceValue>
            <TargetAttribute>ChildDesign.Title Block.Description</TargetAttribute>
        </AttributeMap>
    </AttributeMapGroup>
</MappingEntry>
```

For a single MappingEntry, it contains an AttributeMapGroup element and a set of EventTriggers. The AttributeMapGroup is a set of AttributeMaps. The available attribute types of AttributeMap are `setValue` and `clearValue`. The `setValue` specifies setting the source value to the target attribute. The `clearValue` is only supported for a list attribute and removes the source value from the target attribute.

Configuring Events

EventTrigger is used to identify which trigger event applies to the mapping. EventTrigger is a direct inner element of MappingEntry. So, only when the expected event trigger is triggered, then mapping happens. In the following example, mapping happens only when updating the Item Title Block:

```
<MappingEntry type="itemMapping" >
    <EventTrigger eventType="update" objectType="Items"
    tableName="Title Block" />
    More elements
</ MappingEntry>
```

The attribute `eventType` indicates what event type is expected. The available values are as follows:

- `checkIn`
- `checkOut`
- `changeStatus`
- `add`
- `remove`
- `update`

The object type is the base class name, class name or subclass name of Item, Design, or Change. It is case sensitive and can be obtained from the Java Client.

The `EventTrigger` must correspond to the mapping type of `MappingEntry`. For example, with a `changeStatus` event, the mapping type can only be `changeMapping`. The following example shows a `changeStatus` event:

```
EventTrigger eventType="changeStatus" workflow="Default Change Orders"
statusFrom="CCB" statusTo="Released" />
```

The `statusFrom` and `statusTo` can be a single status or a combination of more than one status. If you do not indicate a particular status, you can specify the value as `Any`.

Configuring Design Revision Control

After check in or check out of a Design, the revision number of the Design and its related Items is automatically increased.

The `ec-solution-revision-config.xml` file is used to configure revision control:

```
<RevisionMappingConfig
xmlns="http://www.oracle.com/Agile/RevisionMappingConfig">
  <!--Global Settings-->
  <ItemInitialRevision>Introductory</ItemInitialRevision>
  <RevisionSequences>
    <RevisionSequence>
      <Name>RevisionSequence</Name>
      <Value>Introductory,A,B,C,D,E,F,G,H,I,J,K</Value>
    </RevisionSequence>
  </RevisionSequences>
  <DesignRevisonLogic>
    <increment>true</increment>
  </DesignRevisonLogic>
  <VersionSeparator>.</VersionSeparator>
  <DesignInitialValue>1</DesignInitialValue>
</RevisionMappingConfig>
```