Oracle® Application Integration Architecture

Oracle Design to Release Integration Pack for Agile PLM
Product Lifecycle Management and JD Edwards EnterpriseOne
Implementation Guide

Release 11.1
E27416-02

April 2012

The Oracle Design to Release Integration Pack for Agile PLM Product Lifecycle Management and JD Edwards EnterpriseOne implementation guide is a valuable resource for administrators and developers involved in the implementation, administration and deployment of Oracle’s next-generation integrated enterprise PLM processes provided by Oracle Application Integration Architecture (AIA) Pre-Built Integrations Release 11.2.

The first part of this guide focuses on understanding the pre-built integration between Agile PLM and JD Edwards EnterpriseOne. It also lists the various assumptions and constrains, process flows, interfaces and integration services used by Agile PLM Pre-Built integrations.

The second part of this guide discusses the prerequisites, post installation configuration steps, cross-references and National language support (NLS) required for integrating Agile PLM with JD Edwards EnterpriseOne.
## Contents

Preface ........................................................................................................................................................................... xi
    What's New in this Guide ........................................................................................................................................... xi
    Common Oracle AIA Pre-Built Integration Guides ............................................................................................... xi
    Documentation Accessibility ................................................................................................................................... xii
    Additional Resources ................................................................................................................................................... xii

Part I  Understanding the Delivered Integrations

1  Agile Product Lifecycle Management and JD Edwards EnterpriseOne
    1.1  Overview ............................................................................................................................................... 1-1
    1.2  Solution Assumptions and Constraints ................................................................................................. 1-2
    1.3  Architecture of Agile PLM Integration .................................................................................................... 1-2
    1.4  Business Process Flows ............................................................................................................................ 1-4
        1.4.1  Agile PLM to JD Edwards EnterpriseOne Process Flow ................................................................. 1-4
        1.4.2  JD Edwards EnterpriseOne to Agile PLM Process Flow ................................................................. 1-4
        1.4.2.1  JD Edwards EnterpriseOne to Agile PLM Initial Load .................................................................. 1-5
        1.4.2.2  JD Edwards EnterpriseOne to Agile PLM Processes ....................................................................... 1-6
        1.4.2.3  JD Edwards EnterpriseOne Item and Unit Cost Information to Agile PLM (Batch) .................... 1-6
        1.4.2.4  JD Edwards EnterpriseOne Item Balance Information to Agile PLM (Batch) ............................. 1-6
    1.5  Components of Agile PLM Integration Pack for JD Edwards EnterpriseOne Integration .................. 1-6
        1.5.1  Oracle AIA Foundation Pack ................................................................................................................ 1-7
        1.5.2  Agile PLM ............................................................................................................................................. 1-7
        1.5.3  JD Edwards Enterprise One .................................................................................................................. 1-8

2  Process Integration for Initial Load of Items and Bill of Materials
    2.1  Overview ................................................................................................................................................... 2-1
    2.2  Item and Bill of Materials Initial Load Integration Flow ........................................................................ 2-2
        2.2.1  Item Initial Load Flow ....................................................................................................................... 2-2
        2.2.2  BOM Initial Load Flow ...................................................................................................................... 2-3
        2.2.3  Initial Load Orchestration ................................................................................................................ 2-4
    2.3  Solution Assumptions and Constraints .................................................................................................... 2-6
    2.4  JD Edwards EnterpriseOne Interfaces ..................................................................................................... 2-7
3 Process Integration for Item Attributes and Cost Updates

3.1 Overview ....................................................................................................................... 3-1
3.2 Item Attributes and Cost Updates Process Integration .................................................. 3-2
  3.2.1 Update Item Attribute Flow ...................................................................................... 3-2
  3.2.2 Update Item Attribute Orchestration ........................................................................ 3-4
3.3 Solution Assumptions and Constraints .......................................................................... 3-5
3.4 Agile PLM Interfaces ..................................................................................................... 3-6
3.5 JD Edwards EnterpriseOne Interfaces ............................................................................ 3-6
3.6 Core AIA Components ................................................................................................. 3-7
3.7 Agile PLM and JD Edwards EnterpriseOne Components for Update Item Attributes ... 3-7
3.8 Integration Services ....................................................................................................... 3-8
  3.8.1 ItemEBSV2................................................................................................................. 3-8
  3.8.2 UpdateItemListJDEE1ReqABCSImpl ......................................................................... 3-8
  3.8.3 UpdateItemListAgileProvABCSImpl ......................................................................... 3-9

4 Process Integration for Item Balance Updates

4.1 Overview ....................................................................................................................... 4-1
4.2 Item Balance Updates Process Integration ..................................................................... 4-2
  4.2.1 Item Balance Update Orchestration .......................................................................... 4-3
4.3 Solution Assumptions and Constraints .......................................................................... 4-5
4.4 Agile PLM Interfaces ..................................................................................................... 4-5
4.5 JD Edwards EnterpriseOne Interfaces ............................................................................ 4-6
4.6 Core AIA Components ................................................................................................. 4-6
4.6.1 Agile PLM and JD Edwards EnterpriseOne Components for Update Item Balance ... 4-7
4.7 Integration Services ....................................................................................................... 4-7
  4.7.1 ItemBalanceEBS ....................................................................................................... 4-7
  4.7.2 UpdateItemBalanceListJDEE1ReqABCSImpl ............................................................. 4-8
  4.7.3 UpdateItemBalanceListAgileProvABCSImpl ............................................................. 4-8

5 Process Integration for Product Design Changes

5.1 Overview ....................................................................................................................... 5-1
5.2 Process Flow for Product Design Changes ..................................................................... 5-2
  5.2.1 Creating ECO Flow .................................................................................................. 5-2
  5.2.1.1 Monitoring ECO Processes .................................................................................. 5-3
  5.2.2 Orchestrating ECO Flow ........................................................................................ 5-4
5.3 Solution Assumptions and Constraints .......................................................................... 5-6
5.4 Agile PLM Interfaces ..................................................................................................... 5-7
5.5 JD Edwards EnterpriseOne Interfaces ............................................................................ 5-8
5.6 Core AIA Components ................................................................................................. 5-8
5.6.1 Agile PLM and JD Edwards EnterpriseOne Components for Creating ECO ......... 5-8
5.7 Integration Services ....................................................................................................... 5-9
  5.7.1 EngineeringChangeOrderEBS ............................................................................... 5-9
  5.7.2 ProcessEngineeringChangeOrderAgileReqABCSImpl .............................................. 5-10
  5.7.3 CreateEngineeringChangeOrderListJDEE1ProvABCSImpl ...................................... 5-10
6 Process Integration Flow for ECO Implementation

6.1 Overview ............................................................... 6-1
6.2 ECO Implementation Process Integration ................. 6-2
6.2.1 Updating ECO Flow ........................................ 6-2
6.2.2 Changing Status Operation ............................. 6-3
6.2.3 Updating ECO Orchestration ............................ 6-5
6.3 Solution Assumptions and Constraints ..................... 6-6
6.4 Agile PLM Interfaces ............................................ 6-7
6.5 JD Edwards EnterpriseOne Interfaces ...................... 6-7
6.6 Core AIA Components ........................................ 6-7
6.7 Agile PLM and JD Edwards EnterpriseOne Components for Update ECO ..... 6-8
6.8 Integration Services ............................................ 6-9
6.8.1 EngineeringChangeOrderEBS .......................... 6-9
6.8.2 UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl .......... 6-9
6.8.3 UpdateEngineeringChangeOrderListAgileProvABCSImpl .......... 6-10

Part II Implementing the Delivered Integrations

7 Configuring Design to Release: Agile PLM - JDE E1

7.1 Prerequisites .......................................................... 7-1
7.1.1 Prerequisites for Initial Load of Items and BOM .......... 7-1
7.1.2 Prerequisites for Product Design Changes .............. 7-1
7.1.3 Prerequisites for ECO Implementation .................. 7-2
7.1.4 Prerequisites for Item Attributes and Cost Updates .... 7-2
7.1.5 Prerequisites for Item Balance Updates ................ 7-2
7.2 Setting Up Participating Applications ....................... 7-2
7.2.1 Setting Up Agile Content Services ....................... 7-2
7.2.1.1 Create New Destinations ............................... 7-2
7.2.1.2 Create New Events for ECO and SCO ............... 7-3
7.2.1.3 Define Filters .............................................. 7-4
7.2.1.4 Create New Subscribers for ECO and SCO .......... 7-4
7.2.1.5 Set ECO and SCO Privileges .......................... 7-5
7.2.1.6 Set up Agile Item Quantity Attributes ............... 7-5
7.2.1.7 Create a New Workflow for Initial Load Change Orders 7-6
7.2.1.8 Set ECO Workflow Privileges ........................ 7-7
7.2.1.9 Setup Change_Status Table for Workflow .............. 7-7
7.2.1.10 Set Up Agile Notifications ............................. 7-9
7.2.2 Setting Up JD Edwards EnterpriseOne .................. 7-9
7.2.2.1 Setting Up Batch Programs ............................ 7-9
7.2.2.2 Setting Up UDCs ......................................... 7-14
7.2.2.3 Setting Up JD Edwards EnterpriseOne Web Services .... 7-16
7.2.2.4 Setting Up BSSVs ....................................... 7-18
7.2.2.5 Configuration Parameters for Creating ECO Flow .... 7-19
7.2.3 Setting Up OWSM ............................................. 7-19
7.2.3.1 Creating Credential Maps and Credential Keys .... 7-20
7.2.4 Setting Up AIA ............................................... 7-21
A  Queue Management
A.1  Queue Management Solution ................................................................. A-1
  A.1.1  Queue Controller ............................................................................. A-2
  A.1.2  Queue Schema .................................................................................. A-2
  A.1.3  Queue Monitor .................................................................................. A-2
  A.2  User Interface ....................................................................................... A-2
    A.2.1  Accessing the Process Queue Monitor ........................................... A-3
    A.2.2  Fields and Attributes ...................................................................... A-3
      A.2.2.1  Change Order Queue Monitor ................................................ A-3
      A.2.2.2  Queue Operators ..................................................................... A-4
      A.2.3  Filters ............................................................................................ A-4
        A.2.3.1  Filter 1: All Change Orders ..................................................... A-5
        A.2.3.2  Filter 2: Errored Change Orders Only ..................................... A-5
        A.2.3.3  Filter 3: Pending Changes Only .............................................. A-5
        A.2.3.4  Filter 4: Completed Change Orders Only ............................... A-6
        A.2.3.5  Filter 5: Changes Errored Within Last Week ....................... A-6
        A.2.3.6  Filter 6: Unprocessed Change Orders .................................... A-6
        A.2.3.7  Filter 7: All Delete Flags ....................................................... A-7
        A.2.3.8  Advance Query Abilities ....................................................... A-7
  A.3  Queue Manager Services ..................................................................... A-8
    A.3.1  Transformations .............................................................................. A-9
    A.3.2  Implementation Details ................................................................... A-10
    A.3.3  Error Management ........................................................................ A-10

B  Mapping Page 2 and Page 3 Attributes
B.1  Inserting Code to Map Page 2 and Page 3 Attributes ........................ B-1
C  Troubleshooting

C.1  Resolving Common Issues ....................................................................................................... C-1
List of Figures

1–1 Agile PLM to JD Edwards EnterpriseOne integration architecture ........................................ 1-3
1–2 Agile PLM to JD Edwards EnterpriseOne integration architecture ........................................ 1-4
1–3 Initial load of items and BOMs from JD Edwards EnterpriseOne........................................ 1-5
2–1 Integration flow for initial load.............................................................................................. 2-3
2–2 Orchestration of initial load from JD Edwards EnterpriseOne to Agile PLM ........................ 2-4
3–1 Update Item Attribute process flow..................................................................................... 3-4
4–1 Update Item balance integration flow ................................................................................... 4-3
5–1 Create ECO sequence diagram .......................................................................................... 5-3
5–2 ECO created in a sequence ............................................................................................... 5-4
6–1 Update ECO flow from JD Edwards EnterpriseOne to Agile PLM .................................... 6-3
7–1 Example of data selection for the batch program ............................................................. 7-10
7–2 Example of data selection................................................................................................... 7-11
7–3 Work with User Defined Code Page................................................................................ 7-14
7–4 User Defined Codes........................................................................................................... 7-15
7–5 Entering Code for UDC.................................................................................................... 7-15
7–6 Saving the UDC values ..................................................................................................... 7-16
7–7 Work with Business Property Page.................................................................................. 7-19
7–8 Selecting security credentials ............................................................................................ 7-20
7–9 Create map........................................................................................................................ 7-20
7–10 Creating a key.................................................................................................................. 7-21
7–11 System Mbean browser .................................................................................................. 7-24
7–12 Resequencer attribute ..................................................................................................... 7-25
7–13 Change Order Queue ...................................................................................................... 7-51
7–14 Example of sample error message ................................................................................... 7-52
7–15 Path to navigate to faulted instance ................................................................................ 7-54
7–16 Fault Instance................................................................................................................ 7-54
7–17 Flow Trace..................................................................................................................... 7-55
7–18 Audit Trial....................................................................................................................... 7-56
7–19 Flow Tab ........................................................................................................................ 7-57
A–1 Process Queue Monitor ...................................................................................................... A-3
A–2 Process Denoters ............................................................................................................. A-4
A–3 Change Priorities............................................................................................................. A-4
A–4 Filter All Change Orders ................................................................................................. A-5
A–5 All Change Orders .......................................................................................................... A-5
A–6 Change Orders with Errors .............................................................................................. A-5
A–7 Pending Change Orders .................................................................................................. A-6
A–8 Completed Change Orders .............................................................................................. A-6
A–9 ChangesErrored within Last Week .................................................................................. A-6
A–10 Unprocessed Change Orders ......................................................................................... A-7
A–11 All Delete Flags ............................................................................................................ A-7
A–12 Advance Query Abilities ............................................................................................... A-7
A–13 Search Parameters ........................................................................................................ A-8
A–14 QueueProcessorServiceImpl ........................................................................................ A-9
<table>
<thead>
<tr>
<th>List of Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>2–1</td>
</tr>
<tr>
<td>2–2</td>
</tr>
<tr>
<td>2–3</td>
</tr>
<tr>
<td>2–4</td>
</tr>
<tr>
<td>3–1</td>
</tr>
<tr>
<td>3–2</td>
</tr>
<tr>
<td>3–3</td>
</tr>
<tr>
<td>3–4</td>
</tr>
<tr>
<td>3–5</td>
</tr>
<tr>
<td>3–6</td>
</tr>
<tr>
<td>3–7</td>
</tr>
<tr>
<td>3–8</td>
</tr>
<tr>
<td>3–9</td>
</tr>
<tr>
<td>4–1</td>
</tr>
<tr>
<td>4–2</td>
</tr>
<tr>
<td>4–3</td>
</tr>
<tr>
<td>4–4</td>
</tr>
<tr>
<td>4–5</td>
</tr>
<tr>
<td>4–6</td>
</tr>
<tr>
<td>4–7</td>
</tr>
<tr>
<td>4–8</td>
</tr>
<tr>
<td>4–9</td>
</tr>
<tr>
<td>5–1</td>
</tr>
<tr>
<td>5–2</td>
</tr>
<tr>
<td>5–3</td>
</tr>
<tr>
<td>5–4</td>
</tr>
<tr>
<td>5–5</td>
</tr>
<tr>
<td>5–6</td>
</tr>
<tr>
<td>5–7</td>
</tr>
<tr>
<td>5–8</td>
</tr>
<tr>
<td>6–1</td>
</tr>
<tr>
<td>6–2</td>
</tr>
<tr>
<td>6–3</td>
</tr>
<tr>
<td>6–4</td>
</tr>
<tr>
<td>6–5</td>
</tr>
<tr>
<td>6–6</td>
</tr>
<tr>
<td>6–7</td>
</tr>
<tr>
<td>6–8</td>
</tr>
<tr>
<td>6–9</td>
</tr>
<tr>
<td>6–10</td>
</tr>
<tr>
<td>7–1</td>
</tr>
<tr>
<td>7–2</td>
</tr>
<tr>
<td>7–3</td>
</tr>
<tr>
<td>7–4</td>
</tr>
<tr>
<td>7–5</td>
</tr>
<tr>
<td>7–6</td>
</tr>
<tr>
<td>7–7</td>
</tr>
<tr>
<td>7–8</td>
</tr>
<tr>
<td>7–9</td>
</tr>
<tr>
<td>7–10</td>
</tr>
<tr>
<td>7–11</td>
</tr>
<tr>
<td>7–12</td>
</tr>
<tr>
<td>7–13</td>
</tr>
</tbody>
</table>
Web service operations
Values for Key and Group fields
Consumer services properties
Consumer service properties
Cross reference
XRef Table Format
ITEM_ITEMID XRef Table Value Format
Sample Data for XRef ITEM_ITEMID Table
JDEE1_ITEMID XRef Table Value Format
Sample Data for XRef JDEE1_ITEMID Table
Domain Value Mapping
Agile PLM configuration properties
Settings CreateQueueService service property
Settings for QueueProcessorServiceImpl service property
Settings for UpdateEngineeringChangeOrder property
Settings for UpdateItemBalance property
Settings for UpdateItemListAgile Property
Settings for ProcessEngineeringChangeOrder Property
Settings for JDEE1 Property
Settings for CreateEngineeringChangeOrder property
Settings for UpdateEngineeringChangeOrder property
Settings for UpdateItemBalance property
Settings for UpdateItemList property
Settings for InitialLoadBillofMaterial property
Settings for InitialLoadItemList property
Errors generated by PIP services
Errors generated by initial load
Preface


What's New in this Guide

- The Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations is restructured into a general installation chapter with an individual configuration and deployment chapter for each pre-built integration.
- The term *process integration pack* is replaced with the term *pre-built integrations*.
- The implementation guides are restructured into two parts: design and set up.
  - Part I - Design: This part provides functional overviews, activity diagrams, assumptions and constraints, and technical sequence diagrams and steps.
  - Part II - Set up: This part provides prerequisites, data requirements, and configuration steps.
- Starting with this release, these integrations are no longer available:
  - Oracle CRM On Demand Integration Pack for JD Edwards EnterpriseOne: Lead to Order
  - Oracle Workforce Administration Integration Pack for PeopleSoft Human Resources

Common Oracle AIA Pre-Built Integration Guides

Oracle Application Integration Architecture Pre-Built Integrations 11.1 includes the following guides shared by all products delivered with this release:

- Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations Release 11.1
  This guide provides an overview of the installation process, including how to install, configure, and deploy your pre-built integrations. The steps required to upgrade your pre-built integrations to the latest release are also provided.
- Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide
  This guide describes:
  - How to work with and configure Session Pool Manager (SPM), which is a service in the Oracle SOA Suite web server whose primary function is to manage a pool of web server session tokens that can be reused by BPEL flows.
How to deploy and configure the AIACompositeScheduler. This is a utility component that is used by pre-built integrations to schedule a service-oriented architecture (SOA) composite to be invoked at the specified time interval.

Oracle Application Integration Architecture Pre-Built Integrations 11.1: Product-to-Guide Index

The Product-to-Guide index lists the guides that provide information for each product delivered in this release.

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 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.

Additional Resources

The following resources are also available:

Oracle Application Integration Architecture Foundation Pack:

Oracle AIA Pre-Built integrations require Foundation Pack 11.1.1.5.0 to be installed. Refer to the Foundation Pack documentation library on OTN to download the Foundation Pack guides at http://download.oracle.com/docs/cd/E21764_01/aia.htm.

Oracle Application Integration Architecture: Product-to-Guide Index:

Oracle Technology Network:
http://www.oracle.com/technetwork/index.html

Known Issues and Workarounds:

My Oracle Support: https://support.oracle.com/

Release Notes:

Oracle Technology Network:
http://www.oracle.com/technetwork/index.html

Documentation updates:

Oracle Technology Network:
http://www.oracle.com/technetwork/index.html
Part I
Understanding the Delivered Integrations

Part I contains the following chapters:

- Chapter 1, "Agile Product Lifecycle Management and JD Edwards EnterpriseOne"
- Chapter 2, "Process Integration for Initial Load of Items and Bill of Materials"
- Chapter 3, "Process Integration for Item Attributes and Cost Updates"
- Chapter 4, "Process Integration for Item Balance Updates"
- Chapter 5, "Process Integration for Product Design Changes"
- Chapter 6, "Process Integration Flow for ECO Implementation"
This chapter provides an overview of the Oracle Design to Release: Agile Product Lifecycle Management - JD Edwards EnterpriseOne integration and discusses:

- Solution assumptions and constraints
- Architecture of Agile Product Lifecycle Management (PLM) integration
- Business process task flows
- Components of Agile PLM pre-built integration

1.1 Overview

Agile PLM enables companies to manage individual product life cycles and to complete product portfolios and programs tied to product conception, design, launch, maturity, and phase-out. Primary users of Agile PLM are involved directly or indirectly in the product design. These users manage content in Agile PLM that often directly affects processes managed in other enterprise systems, such as JD Edwards EnterpriseOne, Manufacturing Execution Systems (MES), Customer Relationship Management (CRM), and so forth.

Companies must propagate timely and accurate product design information from the PLM system to the manufacturing system to ensure that products are built to the correct specifications, thereby enabling a low time to market and eliminating excess and obsolete inventory. Any failure in this crucial integration can lead to products being manufactured to incorrect designs, which can lead to these consequences:

- Delays in product launch (thereby compromising market share and profit margins)
- Compromised product quality, leading to higher service costs and further loss of market share
- Expensive inventory write-offs that directly affect the bottom line

The integration of Agile PLM and JD Edwards EnterpriseOne is designed to synchronize product content information between Agile Product Collaboration and JD Edwards EnterpriseOne.

This integration is a first new generation integration solution between Agile PLM and JD Edwards EnterpriseOne Manufacturing.

This integration consists of the following integration flows:
1. Initial load of items and Bills of Material (BOMs) from JD Edwards EnterpriseOne to Agile PLM
2. Synchronization of engineering change order release from Agile PLM to JD Edwards EnterpriseOne through the Engineering Change Order (ECO) process
3. Synchronization of ECO status from JD Edwards EnterpriseOne to Agile PLM
4. Synchronization of item attributes and cost updates from JD Edwards EnterpriseOne to Agile PLM
5. Synchronization of item balance updates from JD Edwards EnterpriseOne to Agile PLM

1.2 Solution Assumptions and Constraints

Design Assumptions:
1. Use Agile Content Server for events to trigger the payload from Agile PLM to the integration.
2. This design assumes that these statements are true:
   ■ Predefined blank templates are available for custom fields.
   ■ Transformation logic for classification elements are pre-coded in the delivered XSL, but you may need to modify it to suite your Agile PLM implementation.
3. This design leverages AIA error handling framework.
4. We do not integrate Approved List of Manufacturers (AML) information. We do not import any AML information passed to JD Edwards EnterpriseOne in this integration into JD Edwards EnterpriseOne.

Design Constraints:
1. In a few cases, configuration-driven XSLT may not reflect the changes immediately and may require a restart because the main XSL sheet is cached after a successful compilation.
2. Error handling capabilities of this integration are constrained by the capabilities of the AIA framework.

1.3 Architecture of Agile PLM Integration

1. Messages originating in Agile PLM exit Agile Content Service (ACS) in XML format.
   The messages are then queued and the Agile PLM requestor Application Business Connector Services (ABCS), AgileReqABCSImpl, is called. The Agile requestor (ProcessEngineeringChangeOrderAgileReqABCSImpl) invokes EngineeringChangeOrderEBS.

2. EngineeringChangeOrderEBS routes the EBM to the JD Edwards EnterpriseOne provider (CreateEngineeringChangeOrderListJDEE1ProvABCSImpl).
   The JD Edwards EnterpriseOne provider transforms the enterprise business message (EBM) to application business message (ABM) and then invokes a web service (PBSSV - Published Business Service) for creating items and ECOs in JD Edwards EnterpriseOne. The provider receives the response from the PBSSV and passes it to Agile PLM through EngineeringChangeOrderResponseEBS. The Agile requestor receives this response and relays this information to the queue.
For flows originating from JD Edwards EnterpriseOne, report programs in JD Edwards EnterpriseOne generate XML files, and the corresponding requestor ABCS consumes the XML file and passes it to Agile PLM through enterprise business service (EBS).

Figure 1–1 and Figure 1–2 illustrate the Agile PLM to JD Edwards EnterpriseOne integration architecture:
1.4 **Business Process Flows**

This section discusses:

- Agile PLM to JD Edwards EnterpriseOne process flow
- JD Edwards EnterpriseOne to Agile PLM process flow

1.4.1 **Agile PLM to JD Edwards EnterpriseOne Process Flow**

This integration covers the Agile PLM to JD Edwards EnterpriseOne process flows.

1.4.1.1 **Agile PLM to JD Edwards EnterpriseOne Processes**

**Engineering Change Order Release:**

During product design, developers introduce new products and parts, and modify the design of existing parts. When they complete the authoring of parts attributes and design information, and are ready to publish to the manufacturing system, the product design is released using the ECO process. The change order release process consists of new Part/Product Release (PREL) and Product Design Modification (PDM) flows of Agile PLM.

The release of a change order in Agile PLM acts as a trigger for the synchronization of product design within JD Edwards EnterpriseOne. Because, Agile PLM is a system of
records for product design data, the synchronization process involves transferring the released revision of ECO from Agile PLM to the manufacturing system.

Both these processes use the same integration sequence.

For more information, see Chapter 5: Process Integration for Product Design Changes.

1.4.2 JD Edwards EnterpriseOne to Agile PLM Process Flow

This integration covers the following business flows:

- JD Edwards EnterpriseOne to Agile PLM initial load
- JD Edwards EnterpriseOne to Agile PLM processes
- JD Edwards EnterpriseOne Item and Unit Cost Information to Agile PLM (Batch)
- JD Edwards EnterpriseOne Balance Information to Agile PLM (Batch)

1.4.2.1 JD Edwards EnterpriseOne to Agile PLM Initial Load

Initial load is a process to synchronize items and BOMs from JD Edwards EnterpriseOne to Agile PLM. This process synchronizes data from an existing JD Edwards EnterpriseOne install to a new Agile PLM install. The assumption is that no items or BOMs exist within Agile PLM at the time of the initial load. By following the initial load, items and BOMs are maintained in Agile PLM. When the items and BOMs are modified in Agile PLM, the system sends these changes to JD Edwards EnterpriseOne by using the engineering change order release integration flow:

Figure 1–3 Initial load of items and BOMs from JD Edwards EnterpriseOne
To load the data from JD Edwards EnterpriseOne to Agile PLM, run the extract batch programs in JD Edwards EnterpriseOne. These programs generate XML files that contain the data that is uploaded into Agile PLM. After the files are written to the Enterprise Server, the appropriate consumer service retrieves and debatches the XML file. The routing service then routes each XML file to a BPEL service.

The BPEL service then transforms the list of items into an XML.xsd format and writes the file with the name agile<BPELInstance>.xml. The process then invokes a Java utility, which picks up the agile<BPELInstance>.xml file, zips it, and saves it as agile<BPELInstance>.axml. An ant script invokes the Agile Integration Service (AIS) Java utility to import the data stored in the agile<BPELInstance>.axml file in the form of change orders. These change orders are moved to the released state in Agile PLM by calling a Java utility built with Agile SDK.

1.4.2.2 JD Edwards EnterpriseOne to Agile PLM Processes

Engineering Change Order Update

The engineering change order update process from JD Edwards EnterpriseOne to Agile PLM describes the change in status of the ECO in JD Edwards EnterpriseOne, a part of the manufacturing update business flow. It constitutes a key requirement for keeping users in Agile PLM apprised of the life cycle of an ECO. This process involves communicating the ECO status in JD Edwards EnterpriseOne to a configurable field in the Change Order flex field in Agile PLM; and changing the status of the change order in Agile PLM.

After you synchronize an ECO to JD Edwards EnterpriseOne, certain changes to the ECO in JD Edwards EnterpriseOne must be updated to Agile PLM. For example, if the status of an ECO changes and moves to an implemented state in JD Edwards EnterpriseOne, then you should notify Agile PLM.

1.4.2.3 JD Edwards EnterpriseOne Item and Unit Cost Information to Agile PLM (Batch)

A user can make item and item cost changes in JDE that originate from Agile. When the items updates are done in JDE, the update flow runs from JDE back to Agile PLM. Item and cost universal batch engine (UBE) has a cost method processing option. To extract the unit cost of an item, enable the UBE by selecting the cost method processing option.

1.4.2.4 JD Edwards EnterpriseOne Item Balance Information to Agile PLM (Batch)

The item balance information in JD Edwards EnterpriseOne is stored in three fields: Reserved Quantity, Available Quantity, and On-hand Quantity. An item in JD Edwards EnterpriseOne can exist in multiple branch/plant.

A change to any of the three quantities may affect one or more branch/plants. You make these changes in Agile PLM. Similarly, changes made in the item information may affect one or more branch/plants. For example, changing the cost of an item in JD Edwards EnterpriseOne triggers a similar update in Agile PLM.

1.5 Components of Agile PLM Integration Pack for JD Edwards EnterpriseOne Integration

This integration includes the following components:

- Oracle AIA Foundation Pack
- Agile PLM
■ JD Edwards EnterpriseOne

1.5.1 Oracle AIA Foundation Pack

Pre-built integrations connect participating applications using AIA Foundation Pack and Oracle Fusion Middleware (FMW) components according to the AIA Foundation Pack development methodology.

For more information about the AIA Foundation Pack development methodology, see

1.5.2 Agile PLM

Agile Content Service

Agile Content Service (ACS) is an event-driven XML-based publishing service that makes the product records available to a wide variety of business applications and users, both internally and across the global manufacturing network. In addition to allowing employees and supply chain partners to publish the product record on demand, you can configure ACS to publish automatically the Item Master, BOM, and AML changes during any phase of the product life cycle to multiple destinations, ensuring that everyone is working with up-to-the-minute information.

An ACS module generates output in an XML file or a PDX package.

Agile Integration Service

Agile Integration Services (AIS) is a collection of predefined web services in the Agile Integration Framework that enables communications between the Agile PLM server and disparate systems, including Enterprise Resource Planning (ERP) systems, CRM systems, and Business-to-Business Integration (B2Bi) systems, other Agile PLM systems, and supply chain partners. Using AIS to exchange content with other systems simplifies the process for aggregating raw product content, and makes critical product content available in realtime to other core systems.

AIS web services provide import and export capabilities, which you can use to:

■ Make product content available to Enterprise Application Integration (EAI) systems.

■ Share product content with product design, manufacturing planning, shop floor, ERP, and CRM applications.

■ Make product content available to B2Bi systems that can transfer Agile Application Server data across corporate boundaries to a wide range of external applications.

■ Provide content to custom applications.

■ Import product content data from ERP and other supply chain applications.

Software Development Kit

Agile Software Development Kit (SDK) contains a collection of tools, application programming interfaces (APIs), sample applications, and documentation. You use it to build custom applications that access Agile Application Server functionality. By using the Agile SDK, you can create programs that perform tasks automatically against Agile PLM.

Agile SDK enables the following operations:

■ Integrate Agile PLM with JD Edwards EnterpriseOne or other custom applications.
- Develop applications to process product data.
- Perform batch operations against the Agile Application Server.

Agile SDK has the following modules:
- Agile API: A Java API with interfaces that expose Agile PLM business objects. Use Agile API to create additional Agile PLM clients. You can also use it as part of an extension developed using web service extensions (WSX) or process extensions (PX).
- PX: A framework that allows Agile PLM customers to extend the functionality of Agile PLM clients by adding external reports, user-driven and workflow-driven customized actions, customized tools, and customized automatic number sources.
- WSX: A framework that allows Agile PLM customers to extend the functionality of the Agile PLM server and expose customer-specific solutions using a web service.

1.5.3 JD Edwards Enterprise One

JD Edwards EnterpriseOne provides interoperability with other Oracle applications and third-party systems by natively producing and consuming web services. Web services enable software applications written in various programming languages and running on various platforms to exchange information. JD Edwards EnterpriseOne exposes business services as web services. A web service is a standardized way of integrating web-based applications. JD Edwards EnterpriseOne refers to web services as published business services. Business services enable JD Edwards EnterpriseOne to expose transactions as a basic service that can expose an XML document-based interface.

Published Business Services

A published business service is a JD Edwards EnterpriseOne Object Management Workbench (OMW) object that represents one Java class that publishes multiple business services. When you create a web service, you identify the Java class. The published business service also contains value object classes that make the signature for the published business service.

Business Services

A business service is a JD Edwards EnterpriseOne OMW object that represents one or more classes that expose public methods. Each method performs a business process. A business service also contains internal value object classes that make the signature for the business service methods. These public methods can be called from other business service classes and from published business service classes.

UBEs

You use UBEs for data extraction, transformation, publication, and distribution. You can also use them to generate various outputs, such as operational document creation, customer-facing documentation, ad hoc reporting, financial reporting, and regulatory reporting and analytics.
Process Integration for Initial Load of Items and Bill of Materials

This chapter provides an overview of the integration flow for initial loads and discusses:

- Item and Bill of Materials (BOM) initial load integration flow
- Solution assumptions and constraints
- JD Edwards EnterpriseOne interfaces
- Core Application Integration Architecture (AIA) components

2.1 Overview

The initial load is a one-way process that loads data from JD Edwards EnterpriseOne into Agile PLM in bulk.

We assume that no items or BOMs exist within Agile PLM while the initial load is being performed. After the initial load, items and BOMs are maintained in Agile PLM. When the items and BOMs are modified in Agile PLM, the system moves these changes to JD Edwards EnterpriseOne by using the Engineering Change Order Release integration flow.

**Note:** We strongly recommend that you perform the initial load process only one time. After the successful initial load from JD Edwards EnterpriseOne into Agile PLM, you should not perform this process over the same data. However, after you load the business data initially, you can run the initial load process to load new data that was not previously integrated.

To load data from JD Edwards EnterpriseOne to Agile PLM, run the extract programs in JD Edwards EnterpriseOne. These programs generate XML files that contain the data to be uploaded into Agile PLM. These files are written to the folders on the JD Edwards EnterpriseOne server. The location of these folders is specified in the processing options of the extract programs. The details of the individual extract programs and files are discussed later in this chapter.

After creating the XML files, you can use two methods to pass the data to Agile PLM:

- File Transfer Protocol (FTP)
- Weblogic Server
If you use the FTP method, the initial load process for item and BOMs data requires a configured FTP adapter to monitor the JDEE1In folder for newly created extract files. When the FTP adapter locates a new file, the appropriate Mediator process debatches the files into separate instances. Use debatching to split large XML files into several smaller XML files.

You can also use a Weblogic Server to move the XML files to JD Edwards EnterpriseOne in a folder on the Weblogic Server. If you use this method, the appropriate mediator process detects the file and debatches it into separate instances.

After the XML files are debatched, routing services route each XML file to the appropriate item and BOMs initial load from JD Edwards EnterpriseOne to Agile PLM BPEL Service. This BPEL service then invokes a Java utility to execute Agile Integration Services (AIS), which import the data through a change order. Then the InvokeSDK Java utility is executed using Agile Software Development Kit (SDK) to release the change order.

---

**Note:** If you use the FTP method for the initial load, configure an FTP server on the JD Edwards EnterpriseOne Enterprise server. You should configure the FTP adapter to use the FTP connection set up on the JD Edwards EnterpriseOne server.

---

### 2.2 Item and Bill of Materials Initial Load Integration Flow

This section discusses:

- Item initial load
- BOM initial load
- Initial load orchestration

#### 2.2.1 Item Initial Load Flow

Item data is stored in JD Edwards EnterpriseOne in the Item Master table (F4101) and Item Branch table (F4102). This piece of the process integration enables users to extract item and item branch information from JD Edwards EnterpriseOne and load it into Agile PLM.

Users can select items that are extracted from JD Edwards EnterpriseOne and loaded into Agile PLM using selection criteria that includes category codes.

To extract initial load item records from JD Edwards EnterpriseOne, run the Item Initial Load Extract batch program (R4101D3). This program creates an XML file and, if it completes successfully, it creates a PDF with a successful completion message and the number of items extracted. The XML file is stored in a folder on the JD Edwards EnterpriseOne Enterprise Server. Specify the location of the folder in the processing options of the R4101D3 program.

After the files are written to the JD Edwards EnterpriseOne server, one of these consumer services retrieves and debatches the XML files:

- ItemInitialLoadExtractJDEE1FTPConsumer_ep
  
  Use this consumer service when you configure the FTP adapter to locate files.

- ItemInitialLoadExtractJDEE1FileConsumer_ep
- Use this consumer service to retrieve files from the JDEE1In folder in the Weblogic Server.

After a file is written to the JD Edwards EnterpriseOne server, the appropriate consumer service retrieves and debatches the XML file. XML debatching allows the large XML file created by the extract program to be split into several smaller XML files. The routing service then routes each individual XML file to InitialLoadItemListJDEE1toAgileImpl. This BPEL service transforms the list of items in R4101D3.xml into an XML.xsd format and writes the file with the name agile<BPELInstance>.xml. The BPEL then invokes a Java utility that picks up the agile<BPELInstance >.xml file, zips it, and saves it as agile<BPELInstance >.axml. Then the utility invokes an ant script to call an Agile Integration Service (AIS) Java utility to import the data stored in the agile<BPELInstance >.axml file as a change order. Another Java utility is then invoked to call an ant script to invoke an Agile SDK-built Java utility to release the change order.

**Figure 2–1** illustrates the integration flow for initial load:

![Integration flow for initial load](image)

### 2.2.2 BOM Initial Load Flow

BOM data is stored in JD Edwards EnterpriseOne in the Bill of Material table (F3002). Item information, such as Parent Item Rev Number and Component Stocking Type, is stored in the Item Master table (F4101). This part of the process integration enables users to extract BOM information from JD Edwards EnterpriseOne and load it into Agile PLM.

Users can select BOMs that are extracted from JD Edwards EnterpriseOne and loaded into Agile PLM using selection criteria within the extract UBE.
To extract initial load BOM records from JD Edwards EnterpriseOne, run the BOM Initial Load Extract batch program (R3002D). This program creates an XML file and, if it completes successfully, it creates a PDF with a successful completion message and the number of records extracted. The XML file is stored in a folder on the JD Edwards EnterpriseOne Enterprise server. Specify the location of the folder in the processing options of the R3002D program.

After a file is written to the JD Edwards EnterpriseOne server, the appropriate consumer service retrieves and debatches the XML file. XML debatching allows the large XML file created by the extract program to be split into several smaller XML files. The routing service then routes each individual XML file to InitialLoadBillOfMaterialsListJDEE1toAgileImpl. This BPEL service transforms the list of BOMs in R3002.xml into an XML.xsd format and writes the file with the name agile<BPELInstance>.xml. Then the BPEL invokes a Java utility that picks up the agile< BPELInstance >.xml file, zips it, and saves it as agile< BPELInstance >.axml. Then the utility invokes an ant script to call AIS to import the data stored in the agile<SEQ>.axml file. Another Java utility is then invoked to call an ant script to invoke an Agile SDK-built Java utility to release the change order.

The system uses one of these consumer services to retrieve and debatch the XML files:

- BillOfMaterialsInitialLoadExtractJDEE1FTPConsumer_ep
  Use this consumer service to configure the FTP adapter to locate files.
- BillOfMaterialsInitialLoadExtractJDEE1FileConsumer_ep
  Use this consumer service to retrieve files from the JDEE1In folder in the Weblogic Server.

2.2.3 Initial Load Orchestration

Figure 2–2 illustrates the orchestration of the initial load from JD Edwards EnterpriseOne to Agile PLM:

Figure 2–2 Orchestration of initial load from JD Edwards EnterpriseOne to Agile PLM

Table 2–1 lists the results of each activity involved in an initial load:
Table 2–1 Activities in Initial Load

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Load</td>
<td>Run R4101D3 to export items</td>
<td>R4101D3_mmddyy_hhmss.xml file is placed in the location specified in the UBE processing option.</td>
</tr>
</tbody>
</table>
| 1 | ItemInitialLoadExtractJDEE1FileConsumer  
ItemInitialLoadExtractJDEE1FtpConsumer reads and debatches the file | Batch size is determined from the BatchSize property in ItemInitialLoadExtractJDEE1FileConsumer_ep/ItemInitialLoadExtractJDEE1FtpConsumer_ep. |
| 2 | The system invokes the InitialLoadItemListJDEE1toAgileImpl BPEL service | ItemList ABM is transformed into ItemListAXML, and agile<instanceid>.xml is written with the file adapter. |
| 3 | The system invokes AIS Importer to create a change order | AIS adds affected items to an ECO and imports items in redline mode. |
| 4 | The system retrieves the AIS Result.xml file for error handling | Result<InstanceId>.xml file is retrieved through file adapter and stops processing in case of any error. |
| 5 | The system invokes SDK methods to release the change order | ReleaseECO SDK is invoked to advance the ECO to Released status. |
| 6 | The system invokes SDK methods to release the change order | ReleaseECO SDK is invoked to advance the ECO to Released status. |
| 7 | The system retrieves the SDK Result.xml file for error handling | Result<InstanceId>.xml file is retrieved through file adapter and stops processing in case of any errors. |

BOM Initial Load

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Run R3002D to export BOMs</td>
<td>R3002D_mmddyy_hhmss.xml file is placed in the location specified in the UBE processing options.</td>
</tr>
</tbody>
</table>
Table 2-1 (Cont.) Activities in Initial Load

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>BillOfMaterialsInitialLoadExtractJDEE1FileConsumer or BillOfMaterialsInitialLoadExtractJDEE1FtpConsumer reads and debatches the file</td>
<td>The batch size is determined from the BatchSize Property in BillOfMaterialsInitialLoadExtractJDEE1FileConsumer_\ep/ BillOfMaterialsInitialLoadExtractJDEE1FtpConsumer_\ep</td>
</tr>
<tr>
<td>3</td>
<td>The system invokes the InitialLoadBillOfMaterialsListJDEE1toAgileImplBPEL service</td>
<td>ItemBOM application business message (ABM) is transformed into ItemBOMAXML, and agile&lt;instanceid&gt;.xml is written with the file adapter</td>
</tr>
<tr>
<td>4</td>
<td>The system invokes AIS Importer to create a change order</td>
<td>AIS adds Affected Items and BOM to an ECO and imports BOM in redline mode</td>
</tr>
<tr>
<td>5</td>
<td>The system retrieves the AIS Result.xml file for error handling</td>
<td>Result&lt;InstanceId&gt;.xml file is retrieved through file adapter and stops processing in case of any errors</td>
</tr>
<tr>
<td>6</td>
<td>The system invokes SDK methods to release the change order</td>
<td>ReleaseECO SDK is invoked to advance the ECO to Released status</td>
</tr>
<tr>
<td>7</td>
<td>The system retrieves the SDK Result.xml file for error handling</td>
<td>Result&lt;InstanceId&gt;.xml file is retrieved through file adapter and stops processing in case of any errors</td>
</tr>
</tbody>
</table>

2.3 Solution Assumptions and Constraints

This design assumes that the following statements are true:

1. While installing this integration, items and BOMs from JD Edwards EnterpriseOne does not exist in Agile PLM.

2. When multi site is enabled in Agile PLM, the system sets corresponding sites in Agile PLM, through the AGILE_TARGET_SITE_MAPPING domain value map (DVM), for all items with branch/plants.

3. When multi site is enabled in Agile PLM, the system does not load item initial load batches containing all nonstock items.

   JD Edwards EnterpriseOne users can add a data selection of STKT != ‘N’ in R4101D3 to filter out nonstock items if they choose not to send them.

4. You run the R3002D - BOM Initial Load Extract UBE with appropriate processing options set up to retrieve the Parent Item Revision Level; otherwise, the system
populates the PARENT_ITEM_REVISION_LEVEL Property in AIA configuration properties with the value to be used as a parent item revision level.

5. The system accepts only BOM types that match the "DEFAULT_BOM_TYPE" property in the AIA configuration properties.

JD Edwards EnterpriseOne users can add a data selection in the TBM field to match the value defined in the 'DEFAULT_BOM_TYPE' property.

6. The system accepts only BOMs with a batch quantity of zero (0).

JD Edwards EnterpriseOne users can add a data selection of BQTY = 0 to extract BOMs with a batch quantity equal to zero.

7. The system loads BOM components with the same branch/plant as the parent branch plant into Agile PLM; it filters out other components.

JD Edwards EnterpriseOne users can filter these components by setting the R3002D processing option “Selection for Components” as blank.

8. The system does not accept nonstock BOM components when multi site is enabled in Agile PLM.

JD Edwards EnterpriseOne users can choose to not send nonstock components in an Agile PLM No-Site configuration by adding a data selection for STKT != ‘N’ when running the R3002D UBE.

9. The system considers multiple BOMs for the same parent item as duplicates, and it does not accept them when multi site is not enabled in Agile PLM.

JD Edwards EnterpriseOne users can set appropriate data selections for Branch (MMCU), BOM Type (TBM), Batch Quantity (BQTY), and Batch UOM (UOM) such that only single BOM is selected for a parent Item.

2.4 JD Edwards EnterpriseOne Interfaces

Table 2–2 lists the JD Edwards EnterpriseOne XSD files:

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InitialLoadItemListJDEE1toAgileImpl</td>
<td></td>
</tr>
<tr>
<td>R4101D3.xsd</td>
<td>Contains Item ABM</td>
</tr>
<tr>
<td>ImportAISResult.xsd</td>
<td>Contains Import Data AIS Execution Result ABM</td>
</tr>
<tr>
<td>ReleaseECOSDKResult.xsd</td>
<td>Contains Release ECO SDK Execution Result ABM</td>
</tr>
<tr>
<td>InitialLoadBillOfMaterialsListJDEE1toAgileImpl</td>
<td></td>
</tr>
<tr>
<td>R3002D.xsd</td>
<td>Contains BOM ABM</td>
</tr>
<tr>
<td>ImportAISResult.xsd</td>
<td>Contains Import Data AIS Execution Result ABM</td>
</tr>
<tr>
<td>ReleaseECOSDKResult.xsd</td>
<td>Contains Release ECO SDK Execution Result ABM</td>
</tr>
</tbody>
</table>
2.5 Core AIA Components

Table 2–3 lists the industry components for process integration for initial load:

<table>
<thead>
<tr>
<th>Services</th>
<th>BOM</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABMs</td>
<td>R3002.xsd - JD Edwards EnterpriseOne BOM ABM</td>
<td>R4101D3.xsd - JD Edwards EnterpriseOne Item ABM</td>
</tr>
<tr>
<td></td>
<td>aXML_9226.xsd - Agile PLM BOM ABM</td>
<td>aXML_9226.xsd - Agile Item ABM</td>
</tr>
<tr>
<td></td>
<td>axml_9226.xsd - Agile PLM BOM ABM for Agile 9.3</td>
<td>axml_93.xsd - Agile Item ABM for Agile 9.3</td>
</tr>
<tr>
<td></td>
<td>aXML_931.xsd - Agile PLM BOM ABM for Agile 9.31</td>
<td>aXML_931.xsd - Agile PLM BOM ABM for Agile 9.31</td>
</tr>
<tr>
<td></td>
<td>ImportAISResult.xsd - AIS Result ABM</td>
<td>ReleaseECOSDKResult.xsd - SDK Result ABM</td>
</tr>
</tbody>
</table>

BPEL

| InitialLoadBillOfMaterialsListJDEE1toAgileImpl |

Mediator

| BillOfMaterialsInitialLoadExtractJDEE1Consumer | ItemInitialLoadExtractJDEE1Consumer |

Components Locations

Table 2–4 lists the locations of components:

<table>
<thead>
<tr>
<th>Components</th>
<th>Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Business Objects, ABM, and Common XSD files</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/JDEE1/V1/schemas/</td>
</tr>
<tr>
<td></td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas/</td>
</tr>
<tr>
<td></td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/JDEE1/V1/schemas/AgileInitialLoad/</td>
</tr>
<tr>
<td></td>
<td>http://&lt;servername&gt;:&lt;portname&gt;/AIAComponents/ApplicationObjectLibrary/JDEE1/V1/schemas/AgileInitialLoad/</td>
</tr>
</tbody>
</table>
This chapter provides an overview of the process integration for item attributes and cost updates and discusses:

- Item attributes and cost updates process integration
- Solution assumptions and constraints
- Agile Product Lifecycle Management (PLM) interfaces
- JD Edwards EnterpriseOne interfaces
- Core Application Integration Architecture (AIA) components
- Integration services

3.1 Overview

You complete the update of item attribute and unit cost information from JD Edwards EnterpriseOne to Agile PLM as a batch process. You run a new Universal Batch Engine (UBE) program to extract a list of items (including obsolete items if not filtered in data selection) whose attribute values and unit cost must be updated in Agile PLM. Obsolete items shall be marked as obsolete in the item life cycle field of Agile PLM.

You schedule the execution of the Item data extract UBE R4102D program. This program generates an XML file that contains item data as well as values for all the attributes and unit cost information, which is updated to Agile PLM.

Because this is an update to Agile PLM, the items in the XML input file should already exist in Agile PLM; check the cross-reference record to verify this. If a cross-reference record exists for the item, then Agile PLM updates the corresponding item record with item attribute information from JD Edwards EnterpriseOne. If no cross-reference record exists for the item, then Agile PLM does not update the item record (it is skipped).

The extract includes basic and additional item information, units of measure (UOMs), sales, purchasing classifications, inventory processing information, and so forth, from Item Master (F4101), Item Branch (F4102), and Unit Cost information (F4105).

Item cost information flows one way, from JD Edwards EnterpriseOne to Agile PLM. Agile PLM uses this data strictly for informational purposes to help with product design. You should not change costs in Agile PLM, but if you do, then they are not sent back to JD Edwards EnterpriseOne. R4102D only retrieves items that changed since the last successful run (the last successful run date and time are stored in an
IntegrationTimeStamp table, F0095 for this UBE) or the As of Date specified in the processing option.

The system writes this file to a folder on the JD Edwards EnterpriseOne Enterprise server. You specify the location of this folder in the processing options of the extract program. Details about the extract programs and files are discussed in later sections of this chapter.

After creating the XML files, you can use one of two methods to pass that data to Agile PLM:

- File Transfer Protocol (FTP)
- Weblogic Server

If you use the FTP method, you must configure an FTP adapter for the Update Item Attributes and Unit Cost process to monitor the JDEE1 Enterprise Server folders for newly created extract files. When the FTP adapter locates a new file, the appropriate Mediator process debatches the files into separate instances. Debatching is used to split large XML files into several smaller XML files.

Alternatively, you can move the XML files to the JD Edwards EnterpriseOne JDEE1In folder on the Weblogic Server. If you use this method, the appropriate Mediator process detects the file and debatches it into separate instances.

After the system debatches the XML files, routing services route each XML file to UpdateItemListJDEE1ReqABCSImpl, which then performs the following actions:

- Transforms Item application business message (ABM) to Item enterprise business message (EBM)
- Invokes ItemEBSV2, which does the routing to the provider Application Business Connector Services (ABCS): UpdateItemListAgileProvABCSImpl
- The Agile ABCS provides a call back response to ItemResponseEBSV2. UpdateItemListJDEE1ReqABCSImpl receives this response and updates the IntegrationTimeStamp table with the last successful run date and time.

---

**3.2 Item Attributes and Cost Updates Process Integration**

This section discusses:

- Update item attribute flow
- Update item attribute orchestration

**3.2.1 Update Item Attribute Flow**

To extract the updated item attribute and unit cost information in JD Edwards EnterpriseOne, users should run the Item and Cost Extract batch program (R4102D). R4102D retrieves items that have changed since the last successful run (the last successful run date and time is stored in an IntegrationTimeStamp table - F0095 for this UBE) or the As Of date specified in the processing options.
This program retrieves these items and creates an XML file. If it completes successfully, it also creates a PDF with a successful completion message and the number of items extracted. The system stores the XML file in a folder on the JD Edwards EnterpriseOne Enterprise server. Users should specify the location of the folder in the processing options of the R4102D program.

After the files are written to the JD Edwards EnterpriseOne server, one of these consumer services retrieves and debatches the XML files:

- ItemListExtractJDEE1FTPConsumer_ep
  Use this consumer service to configure the FTP adapter to locate files.

- ItemListExtractJDEE1FileConsumer_ep
  Use this consumer service to move your files to the JD EE1In file on the Weblogic Server.

XML debatching allows the large XML file that the extract program creates to be split into several smaller XML files. The routing service then routes each XML file to UpdateItemListJDEE1ReqABCSImpl. This BPEL service transforms UpdateItemListABM to UpdateItemListEBM. As part of this transformation, if the BPEL service finds any items that were created in JD Edwards EnterpriseOne, it checks them against the cross-reference table and drops them if found.

The BPEL service then makes an asynchronous request-delayed response call to ItemEBSV2 with UpdateItemListEBM. This call is routed to UpdateItemListAgileProvABCSImpl.

UpdateItemListAgileProvABCSImpl updates the item in Agile PLM and sends response UpdateItemListResponseEBM.

UpdateItemListJDEE1ReqABCSImpl receives the response, uses DynamicPartnerlink to determine TargetEndpointLocation, and invokes the IntegrationTimeStampManager web service. IntegrationTimeStampManager web service updates the last successful run date and time in the IntegrationTimeStamp table, F0095 for this batch program (R4102D).

Figure 3–1 illustrates the update item attribute process flow from JD Edwards EnterpriseOne to Agile PLM:
3.2.2 Update Item Attribute Orchestration

Table 3–1 lists the activities involved in the update item attribute orchestration:

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Run R4102D to export items</td>
<td>This activity occurs when you update item-related information in JD Edwards EnterpriseOne</td>
</tr>
<tr>
<td>2</td>
<td>ItemListExtractJDEE1FTPConsumer or ItemListExtractJDEE1FileConsumer</td>
<td>ItemListExtractJDEE1Consumer is triggered.</td>
</tr>
<tr>
<td>3</td>
<td>Activate JD Edwards EnterpriseOne Item Operational Attribute Update Action trigger</td>
<td>JD Edwards EnterpriseOne Item Operational Attribute Update Action trigger invokes the UpdateItemListJDEE1ReqABCSImpl</td>
</tr>
<tr>
<td>4</td>
<td>UpdateItemListJDEE1ReqABCImpl invokes ItemEBSV2</td>
<td>An invoke activity in UpdateItemListJDEE1ReqABCSImpl invokes the UpdateItemList operation on ItemEBSV2 with UpdateItemListEBM as input</td>
</tr>
<tr>
<td>5</td>
<td>ItemEBSV2 routes the UpdateItemListEBM to UpdateItemListAgileProvABCImpl</td>
<td>ItemEBSV2 routes UpdateItemListEBM as input to UpdateItemListAgileProvABCImpl</td>
</tr>
</tbody>
</table>
### 3.3 Solution Assumptions and Constraints

This design assumes that the following statements are true:

1. **If item information needs to be retrieved from multiple branch/plants in JD Edwards EnterpriseOne**, the following constraints should be met to support this process:
   - Configure Agile PLM multi-sites
   - Set up one-to-one mapping between Agile PLM sites and JD Edwards EnterpriseOne branch/plants. Use domain value maps (DVMs) to accomplish this mapping.
   
   Locate the attributes to be updated with JD Edwards EnterpriseOne data on the Sites tab of the item if multi-site is configured.

2. **Agile PLM multi-sites are required when item information needs to be retrieved from multiple branch/plants in JD Edwards EnterpriseOne.** You can update any Title Block or Page Two attribute with the JD Edwards EnterpriseOne data.

---

**Table 3–1 (Cont.) Activities for Update Item Attribute Orchestration**

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>UpdateItemListAgileProvABCImpl invokes the Agile PLM Item Operational Attribute Update web service</td>
<td>ItemEBSV2 routes UpdateItemListEBM as input to UpdateItemListAgileProvABCImpl. UpdateItemListAgileProvABCImpl transforms the UpdateItemListEBM to AgileUpdateItemListABM and invokes the UpdateItem service operation on Agile web service to update item cost-related attribute information from JD Edwards EnterpriseOne to Agile PLM. <strong>Note:</strong> The UpdateItemListAgileProvABCImpl composite uses the oracle/wss_http_token_client_policy client security policy while calling ItemABS Service hosted on the Agile server. The security credentials for this are stored in the csf-key AgileWebServicesKey on Fusion Middleware (FMW). The Agile username and password must be setup correctly so that the Agile service is invoked successfully from the AIA composite in the FMW layer. The AgileUpdateItemListResponseABM is returned to UpdateItemListAgileProvABCImpl</td>
</tr>
<tr>
<td>7</td>
<td>UpdateItemListAgileProvABCImpl sends a response to ItemResponseEBSV2</td>
<td>UpdateItemListAgileProvABCImpl transforms the AgileUpdateItemListResponseABM to UpdateItemListResponseEBM and sends a response to ItemResponseEBSV2</td>
</tr>
<tr>
<td>8</td>
<td>ItemResponseEBSV2 sends the UpdateItemListResponseEBM to UpdateItemListJDEE1ReqABCImpl</td>
<td>ItemResponseEBSV2 sends the UpdateItemListResponseEBM to UpdateItemListJDEE1ReqABCImpl</td>
</tr>
</tbody>
</table>
3. Multiple item cost records can be retrieved for any item when the item cost level is 3. In such a case, the UBE extracts the first record to XML and suppresses the processing of subsequent records.

4. If Currency processing is enabled in JD Edwards EnterpriseOne and if the unit cost is not zero, then the currency code is sent back to Agile PLM.

5. If the cost is maintained at Item Level, then the Default Company Currency code is sent back to Agile PLM.

6. The JD Edwards EnterpriseOne IntegrationTimeStampManager business service updates the last batch extract date/time as the Enterprise Server Date/Time.

Note: If you use the FTP method, you must configure an FTP server on the JD Edwards EnterpriseOne Enterprise server. Configure the FTP adapter to use the FTP connection that is set up on the Enterprise server.

### 3.4 Agile PLM Interfaces

Table 3–2 lists the Agile PLM Web Services Definition Language (WSDL) files:

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateItemListAgileProvABCSImpl</td>
<td></td>
</tr>
<tr>
<td>ItemABS.wsdl</td>
<td>ItemABS.wsdl</td>
</tr>
</tbody>
</table>

Table 3–3 lists the Agile PLM XML Schema Definition (XSD) files:

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateItemListAgileProvABCSImpl</td>
<td></td>
</tr>
<tr>
<td>ItemABM.xsd</td>
<td>Contains the Update Item List Request ABM</td>
</tr>
<tr>
<td>ItemABO.xsd</td>
<td>Contains the Update Item List Request ABM</td>
</tr>
</tbody>
</table>

### 3.5 JD Edwards EnterpriseOne Interfaces

Table 3–4 lists the JD Edwards EnterpriseOne WSDL files:

<table>
<thead>
<tr>
<th>Interfaces</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateItemListJDEE1ReqABCSImpl</td>
<td></td>
</tr>
<tr>
<td>IntegrationTimeStampManager.wsdl</td>
<td>IntegrationTimeStampManager.wsdl</td>
</tr>
</tbody>
</table>
Table 3–5 lists the JD Edwards EnterpriseOne XSD files:

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateItemListJDEE1ReqABCSImpl</td>
<td>Contains the Update Item List Request ABM</td>
</tr>
<tr>
<td>R4102D.xsd</td>
<td></td>
</tr>
</tbody>
</table>

### 3.6 Core AIA Components

Table 3–6 lists the industry components of process integration for update item attributes:

<table>
<thead>
<tr>
<th>Component</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBO</td>
<td>ItemEBO</td>
</tr>
<tr>
<td>EBMs</td>
<td>UpdateItemListEBM</td>
</tr>
<tr>
<td></td>
<td>UpdateItemListResponseEBM</td>
</tr>
<tr>
<td>EBS</td>
<td>ItemEBSV2</td>
</tr>
<tr>
<td></td>
<td>ItemResponseEBSV2</td>
</tr>
</tbody>
</table>

Table 3–7 lists the locations of components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBO and EBM XSD files</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/</td>
</tr>
<tr>
<td>WSDL files</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/</td>
</tr>
</tbody>
</table>

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link, EBO, and EBM detail pages in Oracle Enterprise Repository.

For more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link, see Oracle Fusion Middleware Developer’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, “Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository.”

EBOs can be extended, for instance, to add new data elements. These extensions are protected, and remains intact after a patch or an upgrade.

For more information, see Oracle Fusion Middleware Developer’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1.

### 3.7 Agile PLM and JD Edwards EnterpriseOne Components for Update Item Attributes

Table 3–8 lists the requester and provider for each service:
3.8 Integration Services

The process integration for item attributes and cost updates from JD Edwards EnterpriseOne to Agile PLM uses these industry components:

- ItemEBSV2
- UpdateItemListJDEE1ReqABCSImpl
- UpdateItemListAgileProvABCSImpl

3.8.1 ItemEBSV2

ItemEBSV2 is the EBS that exposes the operations related to the item integration on the Item EBO.

The routing rules are:

- ItemEBSV2 Mediator service
  
  UpdateItemListJDEE1ReqABCSImpl :Route UpdateItemListEBM to UpdateItemListAgileProvABCSImpl

- ItemResponseEBSV2 Mediator service
  
  UpdateItemListAgileProvABCSImpl Route UpdateItemResponseEBM to UpdateItemJDEE1ReqABCSImpl

3.8.2 UpdateItemListJDEE1ReqABCSImpl

UpdateItemListJDEE1ReqABCSImpl transforms the JD Edwards EnterpriseOne message (UpdateItemListABM) into UpdateItemListEBM and calls the routing service to update item list attributes and item cost attributes. It then gets the UpdateItemListResponseEBM response from Agile PLM and invokes the Integration Time Stamp Manager service to update the Integration Time Stamp table with the last runtime and date.
Process flow:
1. Picks up the xml file based on the properties set in the file and ftp consumer.
2. Receives UpdateItemListABM from file and ftp consumer
3. Transforms JD Edwards EnterpriseOne-specific UpdateItemListABM into UpdateItemListEBM
4. Populates the EBM header
5. Calls the ItemEBSV2 service to send message UpdateItemListEBM to UpdateItemListAgileProvABCSImpl
6. Receives UpdateItemListResponseEBM from ItemResponseEBSV2
7. Calls the Integration Time Stamp Manager service to update the Integration Time Stamp table with the last runtime and date
8. Transforms UpdateItemListJDEE1ReqABCSImpl using this transformation: Xform_ItemListABMReqMsg_To_ItemListEBMReqMsg

3.8.3 UpdateItemListAgileProvABCSImpl

You use UpdateItemListAgileProvABCSImpl to facilitate communications between ItemEBSV2 and Agile PLM web service used for updating the cost information for an item in batch mode in Agile PLM.

It performs the following actions:
1. Receives UpdateItemListReqMsg, which contains UpdateItemListEBM
2. Calls a transform operation to convert the UpdateItemListEBM into AgileUpdateItemListABM
3. Sends AgileUpdateItemListABM as input to the web service operation UpdateItems (coarse-grained application programming interfaces (APIs) on the Agile PLM side) to update items in Agile PLM
4. Receives AgileUpdateItemListResponseABM on successful execution of coarse-grained API
5. Calls a transform operation to convert the AgileUpdateItemListABM to UpdateItemListResponseEBM, which is returned as an output of this BPEL process

If the UpdateItems service operation fails on the Agile PLM side, the system generates a new fault and sends it across with the appropriate error message.
This chapter provides an overview of the process integration for item balance updates and discusses:

- Item balance updates process integration
- Solution assumptions and constraints
- Agile Product Lifecycle Management (PLM) interfaces
- JD Edwards EnterpriseOne interfaces
- Core Application Integration Architecture (AIA) components
- Integration services

### 4.1 Overview

You complete the update of item balance information from JD Edwards EnterpriseOne to Agile PLM as a batch process. A new Universal Batch Engine (UBE) program extracts the list of items whose quantity values must be updated in Agile PLM.

You schedule the execution of Item Balance Data Extract UBE R41021D. This program generates an XML file that contains records containing calculated quantity information for items summarized at the branch/plant level based on information from the item location file and user-defined availability constants that are updated into Agile PLM.

Because this is an update to Agile PLM, the items in the XML input file should already exist in Agile PLM, to verify this, check the cross-reference record. If a cross-reference record exists for the item, then Agile PLM updates the corresponding item record with item balance information from JD Edwards EnterpriseOne. If no cross-reference record exists for the item, then Agile PLM does not update the item record (it is skipped).

Item balance information flows one way, from JD Edwards EnterpriseOne to Agile PLM. Agile PLM uses this data strictly for informational purposes to help with product design. You should not change these values in Agile PLM. If you change these values, they will not be sent to JD Edwards EnterpriseOne.

Use R41021D to retrieve the list of Items for which availability information has changed after the last successful run (the last successful run date and time is stored in an IntegrationTimeStampTable, F0095 for this UBE) or the As of Date specified in the processing option for the purpose of updating item availability information from JD Edwards EnterpriseOne to Agile PLM.
The system writes this file to a folder on the JD Edwards EnterpriseOne Enterprise server. The location of this folder is specified in the processing options of the extract program.

After creating the XML files, you can use one of two methods to pass the data to Agile PLM:

- File Transfer Protocol (FTP)
- Weblogic Server

If you use the FTP method, you must configure an FTP adapter to monitor the JDEdwards EnterpriseOne Enterprise Server folders for newly created extract files. When the FTP adapter locates a new file, the appropriate Mediator process debatches the files into separate instances. You use debatching to split large XML files into several smaller XML files.

Alternatively, you can move the XML files to the JD Edwards EnterpriseOne JDEE1In folder on the Weblogic Server. If you use this method, the appropriate Enterprise Business Service (EBS) process detects the file and debatches it into separate instances.

After the XML files are debatched, routing services route each XML file to UpdateItemBalanceListJDEE1ReqABCSImpl.

For more information, see Setting Up Batch Processing Information.

The UpdateItemBalanceListJDEE1ReqABCSImpl performs the following actions:

- Transforms item balance application business message (ABM) to item balance enterprise business message (EBM).
- Invokes ItemBalanceEBS, which routes to the provider Application Business Connector Services (ABCS): UpdateItemBalanceListAgileProvABCSImpl
- The Agile ABCS provides a call back response to ItemBalanceResponseEBS. UpdateItemBalanceListJDEE1ReqABCSImpl receives this response and updates the IntegrationTimeStamp table with the last successful run date and time.

---

**Note:** If you use the FTP method, you must configure an FTP server on the JD Edwards EnterpriseOne server. Configure the FTP adapter to use the FTP connection that is set up on the JD Edwards EnterpriseOne server.

---

### 4.2 Item Balance Updates Process Integration

Figure 4–1 illustrates the update item balance flow from JD Edwards EnterpriseOne to Agile PLM:
Update item balance process flow includes the following steps:

1. The requester ABCS, defined as an asynchronous process, receives a list of ABMs from the JD Edwards EnterpriseOne UBE batch extracts.
   The list contains ABMs that have update_date greater than the last_successful_run_date of the batch program.

2. The requester BPEL process filters the list of ABMs based on the ID to a list of ABMs those IDs are present in the cross-reference tables in the Fusion Middleware (FMW) layer.
   This provides a list of items that were from Agile PLM. The original list of ABMs may also contain ABMs from non-Agile PLM sources as well.

3. The BPEL process then makes a Mediator call out for end-point virtualization.

4. A transformation converts the ABM to an EBM.

5. An asynchronous request-delayed response call is made to the ItemBalanceEBS with the UpdateItemBalanceListEBM.

6. The BPEL instance is invoked when the asynchronous call gets back from the provider and provides the status of the transaction back to the caller.
   Performance is not affected because it is a scheduled call and is invoked by server not by a user.

7. The concurrent program logs the status of this call.

### 4.2.1 Item Balance Update Orchestration

Table 4–1 lists the activities involved in item balance update orchestration:
Table 4–1 Activities Related to Item Balance Update Orchestration

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Run R41021D to export items</td>
<td>This activity occurs when you update item balance-related information in JD Edwards EnterpriseOne</td>
</tr>
<tr>
<td>2</td>
<td>ItemBalanceListExtractJDEE1F TPConsumer or ItemBalanceListExtractJDEE1FileConsumer reads and debatches the file</td>
<td>The system triggers ItemBalanceListExtractJDEE1Consumer</td>
</tr>
<tr>
<td>3</td>
<td>The system triggers an Oracle item balance action</td>
<td>An Oracle item balance action invokes UpdateItemBalanceListJDEE1ReqABCSImpl</td>
</tr>
<tr>
<td>4</td>
<td>UpdateItemBalanceListJDEE1ReqABCSImpl invokes ItemBalanceEBS</td>
<td>An invoke activity in UpdateItemBalanceListJDEE1ReqABCSImpl invokes the UpdateItemBalanceList operation on ItemBalanceEBS with UpdateItemBalanceListEBM as input</td>
</tr>
<tr>
<td>5</td>
<td>ItemBalanceEBS invokes UpdateItemBalanceListAgileProvABCSImpl</td>
<td>ItemBalanceEBS sends UpdateItemBalanceListEBM as input to UpdateItemBalanceListAgileProvABCSImpl</td>
</tr>
<tr>
<td>6</td>
<td>UpdateItemBalanceListAgileProvABCSImpl invokes the Agile PLM item balance web service</td>
<td>UpdateItemBalanceListAgileProvABCSImpl transforms the UpdateItemBalanceListEBM to AgileUpdateItemListABM and invokes the UpdateItem service operation on the Agile web service to update item on-hand quantity information from Oracle to Agile PLM.</td>
</tr>
</tbody>
</table>

Note: The UpdateItemBalanceListAgileProvABCSImpl composite uses the oracle/wss_http_token_client_policy client security policy while calling the ItemABS service hosted on the Agile server. The security credentials for this are stored in the csf-key AgileWebServicesKey on Fusion Middleware (FMW). The Agile username and password must be setup correctly so that the Agile service is invoked successfully from the AIA composite in the FMW layer. The system returns AgileUpdateItemListResponseABM to UpdateItemBalanceListAgileProvABCSImpl.
4.3 Solution Assumptions and Constraints

If item balance information needs to be retrieved from multiple branches or plants in JD Edwards EnterpriseOne, the system must meet these constraints:

1. Configure Agile PLM Multi-Sites.
2. Establish one-to-one mapping between Agile PLM sites and JD Edwards EnterpriseOne branch/plants.
3. Locate the attributes to be updated with the JD Edwards EnterpriseOne data on the Sites tab of the item.

Agile PLM Multi-Sites are not required. You can update any Title Block or Page Two attribute with the JD Edwards EnterpriseOne data.

The JD Edwards EnterpriseOne IntegrationTimeStampManager Business service updates the last batch extract date and time with the enterprise server date and time.

4.4 Agile PLM Interfaces

Table 4–2 lists the Agile PLM Web Services Definition Language (WSDL) files:

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateItemBalanceListAgileProvABCSImpl</td>
<td>ItemABS.wsdl Used to update an item balance information in Agile PLM</td>
</tr>
</tbody>
</table>

Table 4–3 lists the Agile PLM XML Schema Definition (XSD) files:

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateItemBalanceListAgileProvABCSImpl</td>
<td>ItemABM.xsd Contains the update item balance information request and response ABO and ABM</td>
</tr>
<tr>
<td>ItemABO.xsd</td>
<td></td>
</tr>
</tbody>
</table>

---

Table 4–1 (Cont.) Activities Related to Item Balance Update Orchestration

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>UpdateItemBalanceListAgileProvABCSImpl sends a response back to ItemBalanceResponseEBS</td>
<td>UpdateItemBalanceListAgileProvABCSImpl transforms AgileUpdateItemListResponseABM to UpdateItemBalanceListResponseEBM and returns it to ItemBalanceResponseEBS</td>
</tr>
<tr>
<td>8</td>
<td>ItemBalanceResponseEBS sends the UpdateItemBalanceListResponseEBM to UpdateItemBalanceListJDEE1ReqABCSImpl</td>
<td>ItemBalanceResponseEBS sends the UpdateItemBalanceListResponseEBM to UpdateItemBalanceListJDEE1ReqABCSImpl</td>
</tr>
</tbody>
</table>
4.5 JD Edwards EnterpriseOne Interfaces

Table 4–4 lists the JD Edwards EnterpriseOne WSDL files:

<table>
<thead>
<tr>
<th>Interface Description</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateItemBalanceListJDEE1ReqABCSImpl</td>
<td>IntegrationTimeStam pManager.wsdl</td>
</tr>
</tbody>
</table>

Table 4–5 lists the JD Edwards EnterpriseOne XSD files:

<table>
<thead>
<tr>
<th>Interface Description</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateItemBalanceListJDEE1ReqABCSImpl</td>
<td>R41021D.xsd</td>
</tr>
</tbody>
</table>

Table 4–6 lists the components used in the process integration flow for update item balance:

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Component Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBO ItemBalanceEBO</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/</td>
</tr>
<tr>
<td>EBM UpdateItemBalanceListEBM</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/</td>
</tr>
<tr>
<td>EBM UpdateItemBalanceListResponseEBM</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/</td>
</tr>
<tr>
<td>EBS ItemBalanceEBS</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/</td>
</tr>
<tr>
<td>EBS ItemBalanceResponseEBS</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/</td>
</tr>
</tbody>
</table>

Table 4–7 lists the locations of components:

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in Oracle Enterprise Repository.

For more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link, see Oracle Fusion Middleware Developer’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, "Configuring and Using Oracle Enterprise Repository."
EBOs can be extended, for instance, to add new data elements. These extensions are protected, and remains intact after a patch or an upgrade.

For more information, see *Oracle Fusion Middleware Developer’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1*.

### 4.6.1 Agile PLM and JD Edwards EnterpriseOne Components for Update Item Balance

**Table 4–8** lists the Agile PLM and JD Edwards EnterpriseOne components for update item balance:

<table>
<thead>
<tr>
<th>Service</th>
<th>JD Edwards EnterpriseOne (Requester)</th>
<th>Agile PLM (Provider)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABMs</td>
<td>R41021D</td>
<td>AgileUpdateItemListABM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AgileUpdateItemListResponseABM</td>
</tr>
<tr>
<td>ABCS</td>
<td>UpdateItemBalanceListJDEE1ReqABCS</td>
<td>UpdateItemBalanceListAgileProvABCSImpl</td>
</tr>
<tr>
<td>EBS</td>
<td>ItemBalanceEBS</td>
<td>ItemBalanceResponseEBS</td>
</tr>
</tbody>
</table>

**Table 4–9** lists the locations of components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABMs</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas/</td>
</tr>
<tr>
<td>WSDL files</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/JDEE1/V1/wsdls/</td>
</tr>
</tbody>
</table>

### 4.7 Integration Services

The process integration flow for update item balance uses these integration services:

- ItemBalanceEBS
- UpdateItemBalanceListJDEE1ReqABCSImpl
- UpdateItemBalanceListAgileProvABCSImpl

#### 4.7.1 ItemBalanceEBS

ItemBalanceEBS is the Enterprise EBS that exposes the operations related to the item integration on the item EBO. The routing rules are:

- ItemBalanceEBS
  - UpdateItemBalanceListJDEE1ReqABCSImpl
  - Routes UpdateItemBalanceListEBM to UpdateItemBalanceListAgileProvABCSImpl
- ItemBalanceResponseEBS
  - UpdateItemBalanceListAgileProvABCSImpl
Routes UpdateItemBalanceListResponseEBM to UpdateItemBalanceListJDEE1ReqABCSImpl

4.7.2 UpdateItemBalanceListJDEE1ReqABCSImpl

UpdateItemBalanceListJDEE1ReqABCSImpl transforms the JD Edwards EnterpriseOne message (UpdateItemBalanceListABM) into UpdateItemBalanceListEBM and calls the routing service to update item balance list attributes. It then gets the UpdateItemBalanceListResponseEBM response from Agile PLM and invokes the Integration Time Stamp Manager service to update the integration time stamp table with the last runtime and date.

Process flow:
1. Picks up the xml file based on the properties set in the file/ftp consumer
2. Receives UpdateItemBalanceListABM from file/ftp consumer
3. Transforms JD Edwards EnterpriseOne specific UpdateItemBalanceListABM into UpdateItemBalanceListEBM and populates the EBM header
4. Calls the ItemBalanceEBS service to send message UpdateItemBalanceListEBM to UpdateItemBalanceListAgileProvABCSImpl
5. Receives UpdateItemBalanceListResponseEBM from ItemBalanceResponseEBS
6. Calls the Integration Time Stamp Manager service to update the integration time stamp table with the last runtime and date
7. UpdateItemBalanceListJDEE1ReqABCSImpl has the following transformation: Xform_ItemBalanceListABMReqMsg_To_ItemBalanceListEBMReqMsg

4.7.3 UpdateItemBalanceListAgileProvABCSImpl

Use UpdateItemBalanceListAgileProvABCSImpl to facilitate communications between ItemBalanceEBS and the Agile PLM web service used for updating quantity information of the items in batch mode in Agile PLM.

Process flow:
1. Receives UpdateItemBalanceListReqMsg that contains UpdateItemBalanceListEBM
2. Calls transform operation to convert UpdateItemBalanceListEBM into AgileUpdateItemListABM
3. Passes AgileUpdateItemListABM as input to the web service operation UpdateItems (coarse-grained application programming interfaces (APIs) on the Agile PLM side) to update items in Agile PLM
4. Receives AgileUpdateItemListResponseABM on successful execution of coarse-grained API
5. Transforms AgileUpdateItemListResponseABM to UpdateItemBalanceListResponseEBM, which is returned as output of this BPEL process
6. If the UpdateItems service operation fails on the Agile PLM side, generates a new fault and sends it across with an appropriate error message
Process Integration for Product Design Changes

This chapter provides an overview of the integration flow for product design changes and discusses:

- Product design changes process integration
- Solution assumptions and constraints
- Agile Product Lifecycle Management (PLM) interfaces
- JD Edwards EnterpriseOne interfaces
- Core Application Integration Architecture (AIA) components
- Integration services

5.1 Overview

The new part or product release and the product design modification process within Agile PLM results in the publication of an Engineering Change Order (ECO). The ECO that results from a new product contains lines with all of the new components used in a bill of material (BOM) followed by details about the BOM. Similarly, a product design modification results in an ECO containing all of the new or changed components used on the BOM followed by just the changes to the BOM.

Agile PLM can send all BOM information or only net changes to a BOM in an ECO. When Agile PLM releases a new product, it should include all BOM information. After an ECO is created and processed in JD Edwards EnterpriseOne, it should send only net changes in an ECO.

The Agile PLM requester ABCS (ProcessEngineeringChangeOrderAgileReqABCSImpl) sends the ECO enterprise business message (EBM) to EngineeringChangeOrderEBS. The EngineeringChangeOrderEBS does routes the JD Edwards EnterpriseOne Provider Application Business Connector Services (ABCS): CreateEngineeringChangeOrderListJDEE1ProvABCSImpl. The CreateEngineeringChangeOrderListJDEE1ProvABCSImpl performs the following actions:

- Transforms the ECO EBM to ECO application business message (ABM) for JD Edwards EnterpriseOne
- Invokes the JD Edwards EnterpriseOne ECO processor web services

The JD Edwards EnterpriseOne ECO processor provides a response to the requestor by invoking the EngineeringChangeOrderResponseEBS.
5.2 Process Flow for Product Design Changes

This section discusses:
- Creating ECO flow
- Orchestrating ECO flow

5.2.1 Creating ECO Flow

To create an ECO flow, complete these steps:

1. Submit the ECO for approval (workflow step).
2. After the ECO is approved, the system generates an Agile Content Service (ACS) workflow event to trigger the ECO process flow.
3. The queue framework captures the ACS payload (aXML), which is generated for the event, and adds it to the integration queue.
4. The queue framework identifies the highest priority queue message, processes it to create an ECO ABM, and triggers the requestor ABCS.
5. The requestor ABCS transforms the ECO ABM to ECO EBM and triggers an operation on the Enterprise Business Service (EBS) that routes the EBM to JD Edwards EnterpriseOne with ECO EBM as the input.
6. The EBS invokes the provider ABCS with EBM as input; the provider transforms EBM to JD Edwards EnterpriseOne ABM before invoking the EngineeringChangeOrderManager PBSSV (web service).
7. The ECO business flow completes these steps:
   - Creates and/or updates items in JD Edwards EnterpriseOne and links them to branches or plants.
   - Creates an ECO in JD Edwards EnterpriseOne.
   - Associates the list of revised items with new revisions and effectivity dates, and schedules the ECO for implementation.
8. The system updates the status of the queue message in the integration queue for monitoring.
9. In JD Edwards EnterpriseOne, execute the UBE R30510 for creating BOMs.

Figure 5–1 illustrates the Create ECO flow:

Note: The Create ECO flow is one way from Agile PLM to JD Edwards EnterpriseOne. The system does not send any ECOs created in JD Edwards EnterpriseOne to Agile PLM.
5.2.1.1 Monitoring ECO Processes

To monitor each ECO process:

1. Log in to the Oracle Enterprise Manager Console with user name and password.
2. Navigate to SOA > soa-infra > default > CreateEngineeringChangeOrderListJDEE1ProvABCSImpl.
3. Click the Instances tab to view the successful and unsuccessful runs.
4. Click the instance id to display the Flow trace window.
5. Select the CreateEngineeringChangeOrderListJDEE1ProvABCSImpl flow.
7. Scroll down and click InvokeEngineeringChangeOrderManager.
   The Activity Audit Trail page is displayed.
8. Scroll down to find the ECO number under the CreateEngineeringChangeOrderManagerAppRespMsg tag > ConfirmEngineeringChangeOrders.
9. If multiple ECOs are created in this flow, all such ECOs are displayed in a sequential order, Figure 5–2:
Figure 5–2  ECO created in a sequence

Item Product and Item Catalog hold the part number that comes in from Agile PLM. JD Edwards EnterpriseOne generates the Item ID. The user is given this number for reference.

### 5.2.2 Orchestrating ECO Flow

Table 5–1 lists the flow for orchestrating an ECO:

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Agile ACS transmits Agile PLM ECO data in payload in the form of predefined XML format known as aXML. The system queues this file for further processing.</td>
<td>Agile ACS acts as a trigger for ECO use case.</td>
</tr>
</tbody>
</table>
### Table 5–1 (Cont.) Flow for Orchestrating an ECO

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>The QueueController framework reads the highest priority queue message and transforms the payload (aXML) to AgileCreateEngineeringChangeOrderListABM.</td>
<td>QueueController processes the payload.</td>
</tr>
<tr>
<td>3</td>
<td>QueueController invokes the ProcessEngineeringChangeOrderAgileReqABCSImpl with AgileCreateEngineeringChangeOrderListABM as input.</td>
<td>NA</td>
</tr>
<tr>
<td>4</td>
<td>AgileCreateEngineeringChangeOrderListABM is transformed into CreateEngineeringChangeOrderListEBM.</td>
<td>ProcessEngineeringChangeOrderAgileReqABCSImpl makes call backs to Agile web services if needed.</td>
</tr>
<tr>
<td>5</td>
<td>ProcessEngineeringChangeOrderAgileReqABCSImpl invokes the CreateEngineeringChangeOrderList operation on EngineeringChangeOrderEBS with CreateEngineeringChangeOrderListEBM as input.</td>
<td>NA</td>
</tr>
<tr>
<td>6</td>
<td>EngineeringChangeOrderEBS routes CreateEngineeringChangeOrderListEBM to CreateEngineeringChangeOrderListJDEE1ProvABCSImpl.</td>
<td>NA</td>
</tr>
<tr>
<td>7</td>
<td>CreateEngineeringChangeOrderListJDEE1ProvABCSImpl transforms CreateEngineeringChangeOrderListEBM into EngineeringChangeOrderManagerABM and invokes the JD Edwards EnterpriseOne web service with this input payload.</td>
<td>Creates or updates items in JD Edwards EnterpriseOne and links them to the branch/plants, creates an ECO, associates revised items to it, and creates a BOM.</td>
</tr>
<tr>
<td>8</td>
<td>CreateEngineeringChangeOrderListJDEE1ProvABCSImpl invokes CreateEngineeringChangeOrderListResponse operation on EngineeringChangeOrderResponseEBS with CreateEngineeringChangeOrderListResponseEBM as input.</td>
<td>NA</td>
</tr>
<tr>
<td>10</td>
<td>ProcessEngineeringChangeOrderAgileReqABCSImpl sends the status back to the QueueController to update the queue.</td>
<td>QueueController updates this status against the queue message in the database.</td>
</tr>
</tbody>
</table>
5.3 Solution Assumptions and Constraints

This design assumes that the following statements are true:

1. Do not use JD Edwards EnterpriseOne to create any new item or BOM.

2. In Agile PLM, we use the term change order generically to describe an ECO, a Manufacturing Change Order (MCO), and a Site Change Order (SCO).

3. This integration does not support MCOs.

4. The system does not import any AML data passed to JD Edwards EnterpriseOne in this integration into JD Edwards EnterpriseOne.

5. You should use ECO to transfer the new part and update part information from Agile PLM to JD Edwards EnterpriseOne.

   An Agile PLM user must ensure that new and updated items are available on the Affected Items tab before releasing the ECO to JD Edwards EnterpriseOne.

6. Use ECOs to redline BOMs for associating new revisions, effectivity date changes, and inclusion of new items on BOMs.

   When creating new items, an Agile PLM user must ensure that the new items are available on the Affected Items tab before releasing the ECO to JD Edwards EnterpriseOne.

7. Agile PLM user can redline a BOM and add a new site in one change order.

8. If multiple sites are associated in Agile PLM for a single ECO, then multiple ECOs are created for each branch plant (site) in JD Edwards EnterpriseOne.

9. You must approve ECOs in Agile PLM before initiating the integration flow.

10. ECOs are pre-approved in Agile PLM; therefore, an approval layer in JD Edwards EnterpriseOne is not required before implementing these ECOs through the scheduled Batch Process - R30510 for creating BOMs.
However, if the JD Edwards EnterpriseOne user decides to add an approval layer, the system does not support any rejection in this approval process.

11. If BOM creation through R30510 fails for any reason, the message is not sent back to Agile PLM automatically.

12. If the JD Edwards EnterpriseOne user decides to notify Agile PLM about an ECO rejection or failure to create a BOM, then the user must manually update the appropriate status code on the ECO.

The subsequent run of the update ECO flow from JD Edwards EnterpriseOne to Agile PLM will send the status message to Agile PLM.

13. These processes are synchronous.

14. Because, Agile PLM maintains effectivity dates at the parent level, JD Edwards EnterpriseOne forwards the same date to all related component items.

15. The BOMs loaded from JD Edwards EnterpriseOne as part of an initial load might have a mixture of stock and nonstock component items.

You can modify the stock items on the BOM and send them successfully to JD Edwards EnterpriseOne through ECO. However, you cannot change the nonstock components.

16. The maximum description length of ECOs and items in JD Edwards EnterpriseOne is 30.

If an Agile PLM user sends a description that exceeds this length, the message is truncated and the first 30 characters are inserted or updated into JD Edwards EnterpriseOne.

17. The Stocking type DVM (ITEM_TYPE.dvm) must be in sync with the stocking types in JD Edwards EnterpriseOne.

### 5.4 Agile PLM Interfaces

Table 5–2 lists the Agile PLM Web Services Definition Language (WSDL) files:

<table>
<thead>
<tr>
<th>Interfaces</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProcessEngineeringChangeOrderAgileReqABCSImpl</td>
<td>Sends the released ECO to be created as an EngineeringChangeOrder in JD Edwards EnterpriseOne</td>
</tr>
<tr>
<td>ChangeABS.wsdl</td>
<td></td>
</tr>
</tbody>
</table>

Table 5–3 lists the Agile PLM XML Schema Definition (XSD) files:

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateEngineeringChangeOrderAgileReqABCS</td>
<td></td>
</tr>
<tr>
<td>EngineeringChangeOrderABM.xsd</td>
<td>Contains CreateEngineeringChangeOrderListABM and CreateEngineeringChangeOrderListResponseABM, and corresponding ABOs</td>
</tr>
<tr>
<td>EngineeringChangeOrderABO.xsd</td>
<td></td>
</tr>
</tbody>
</table>
5.5 JD Edwards EnterpriseOne Interfaces

Table 5–4 lists the JD Edwards EnterpriseOne WSDL files:

<table>
<thead>
<tr>
<th>Interface Description</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creates an ECO in JD Edwards EnterpriseOne</td>
<td>CreateEngineeringChangeOrderListJDEE1ProvABCSImpl</td>
</tr>
<tr>
<td>EngineeringChangeOrderManager.wSDL</td>
<td>EngineeringChangeOrderManager.wSDL</td>
</tr>
</tbody>
</table>

5.6 Core AIA Components

Table 5–5 lists the components used in the process integration for Create ECO flow:

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Business Object (EBO)</td>
<td>EngineeringChangeOrderEBO</td>
</tr>
<tr>
<td>EBOs</td>
<td>CreateEngineeringChangeOrderListEBM</td>
</tr>
<tr>
<td>EBM</td>
<td>CreateEngineeringChangeOrderListResponseEBM</td>
</tr>
<tr>
<td>EBSs</td>
<td>EngineeringChangeOrderEBS</td>
</tr>
<tr>
<td>EBS</td>
<td>EngineeringChangeOrderResponseEBS</td>
</tr>
</tbody>
</table>

Table 5–6 lists the locations of components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBO and EBM XSD files</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/</td>
</tr>
<tr>
<td>WSDL files</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/</td>
</tr>
</tbody>
</table>

For detailed documentation of individual EBOs and EBM, click the AIA Reference Doc link on EBO and EBM detail pages in Oracle Enterprise Repository.

For more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link, see Oracle Fusion Middleware Developer’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository."

You can extend EBOs, for instance, to add new data elements. These extensions are protected and remains intact after a patch or an upgrade.

For more information, see Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, "Understanding Extensibility."

5.6.1 Agile PLM and JD Edwards EnterpriseOne Components for Creating ECO

Table 5–7 lists the Agile PLM and JD Edwards EnterpriseOne components for creating ECO flow:
Table 5–7  Agile PLM and JD Edwards EnterpriseOne components for creating ECO flow:

<table>
<thead>
<tr>
<th>Services</th>
<th>Agile PLM (Requester)</th>
<th>JD Edwards EnterpriseOne (Provider)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABMs</td>
<td>AgileCreateEngineeringChangeOrderListABM</td>
<td>EngineeringChangeOrderResponseEBS</td>
</tr>
<tr>
<td></td>
<td>AgileUpdateEngineeringChangeOrderListABM</td>
<td></td>
</tr>
<tr>
<td>ABCS</td>
<td>ProcessEngineeringChangeOrder</td>
<td>CreateEngineeringChangeOrderListJDEE1ProvABCSImpl</td>
</tr>
<tr>
<td></td>
<td>AgileReqABCSImpl</td>
<td></td>
</tr>
<tr>
<td>EBS</td>
<td>EngineeringChangeOrderEBS</td>
<td>EngineeringChangeOrderResponseEBS</td>
</tr>
<tr>
<td>Business Process Execution Language (BPEL)</td>
<td>CreateQueueService</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>QueueProcessorServiceImpl</td>
<td></td>
</tr>
</tbody>
</table>

Table 5–8 lists the locations of the components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Business Objects, ABM, and Common XSD files</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas</td>
</tr>
<tr>
<td></td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/JDEE1/V1/schemas</td>
</tr>
<tr>
<td>WSDL files</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas</td>
</tr>
<tr>
<td></td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/JDEE1/V1/schemas</td>
</tr>
</tbody>
</table>

5.7 Integration Services

These services are delivered with the process integration for product design changes from Agile PLM to JD Edwards EnterpriseOne:

- EngineeringChangeOrderEBS
- ProcessEngineeringChangeOrderAgileReqABCSImpl
- CreateEngineeringChangeOrderListJDEE1ProvABCSImpl

5.7.1 EngineeringChangeOrderEBS

EngineeringChangeOrderEBS is the EBS that exposes the operations related to the EngineeringChangeOrder Integration on the EngineeringChangeOrder EBO. The routing rules are:

- EngineeringChangeOrderEBS
- EngineeringChangeOrderResponseEBS
5.7.2 ProcessEngineeringChangeOrderAgileReqABCSImpl

ProcessEngineeringChangeOrderAgileReqABCSImpl transforms the Agile PLM message (AgileCreateEngineeringChangeOrderListABM) into CreateEngineeringChangeOrderListEBM and invokes the EBS service to Create ECOs in JD Edwards EnterpriseOne. JD Edwards EnterpriseOne processes these ECOs and sends a CreateEngineeringChangeOrderListResponseEBM response to Agile PLM. ProcessEngineeringChangeOrderAgileReqABCSImpl receives the CreateEngineeringChangeOrderListResponseEBM and transforms it to Agile ABM message (AgileUpdateEngineeringChangeOrderListABM) before invoking ACS.

The process flow:
- Receives AgileCreateEngineeringChangeOrderListABM from ProcessEngineeringChangeOrderAgileReqABCSImpl, which gets the same from ACS
- Transforms this message into the CreateEngineeringChangeOrderListEBM
- Populates the EBM header by determining the target system IDs and adding them to the EBM header to control routing
- Validates required fields
- Transforms ABM to EBM
- Calls target CreateEngineeringChangeOrderList service with operation CreateEngineeringChangeOrderList
- Receives CreateEngineeringChangeOrderListResponseEBM
- Transforms response EBM to AgileUpdateEngineeringChangeOrderListABM
- Invokes ACS for propagating the response to queue and Agile PLM

ProcessEngineeringChangeOrderAgileReqABCSImpl has the following transformation:
- AgileCreateEngineeringChangeOrderListABM_to_CreateEngineeringChangeOrderListEBM
- CreateEngineeringChangeOrderListResponseEBM_to_AgileUpdateEngineeringChangeOrderListABM

5.7.3 CreateEngineeringChangeOrderListJDEE1ProvABCSImpl

CreateEngineeringChangeOrderListJDEE1ProvABCSImpl receives the CreateEngineeringChangeOrderListEBM message from ECOEBS and transforms the EBM into the JD Edwards EnterpriseOne-specific CreateECOABM. It then invokes the JD Edwards EnterpriseOne EngineeringChangeOrderManager service for creating and updating items and creating ECOs. JDEE1EngineeringChangeOrderManager service returns the list of items and ECOs created for a success or throws the appropriate fault for a failure.

It then transforms the JD Edwards EnterpriseOne-specific response message CreateEngineeringChangeOrderResponseABM to CreateEngineeringChangeOrderListResponseEBM and invokes the routing service
EngineeringChangeOrderResponseEBS for sending the response message back to Agile PLM.

The process flow is:

- Receives CreateEngineeringChangeOrderListEBM from EngineeringChangeOrderEBS
- Transforms CreateEngineeringChangeOrderListEBM into JD Edwards EnterpriseOne-specific message CreateEngineeringChangeOrderABM
  Populate EBM header by determining Target System IDs and adds them to the EBM header to control routing
  Validates required fields
- Assigns implemented status if the payload only contains items
  If the payload contains BOM information, then the status does not change. BOM’s will be created in "Approved" state.
- Uses DynamicPartnerlink to determine TargetEndpointLocation
- Invokes EngineeringChangeOrderManager web service
- Transforms the JD Edwards EnterpriseOne-specific message CreateEngineeringChangeOrderResponseABM into the CreateEngineeringChangeOrderListResponseEBM
- Invokes the EngineeringChangeOrderResponseEBS service to send the response message (CreateEngineeringChangeOrderListResponseEBM) to ProcessEngineeringChangeOrderAgileReqABCSImpl

ProcessEngineeringChangeOrderListJDEE1ProvABCSImpl has the following transformation:

- Xform_EngineeringChangeOrderListEBM_TO_EngineeringChangeOrderManagerReqMsgABM
- Xform_EngineeringChangeOrderManagerRespMsg_TO_EngineeringChangeOrderEBSResponseRespMsg
This chapter provides an overview of the process integration for Engineering Change Order (ECO) implementation and discusses:

- ECO implementation process integration
- Solution assumptions and constraints
- Agile Product Lifecycle Management (PLM) interfaces
- JD Edwards EnterpriseOne interfaces
- Core Application Integration Architecture (AIA) components
- Integration services

### 6.1 Overview

After synchronizing an ECO to JD Edwards EnterpriseOne, any changes you make to the ECO in JD Edwards EnterpriseOne must also be made in Agile PLM. If the status of an ECO changes and moves to an implemented state in JD Edwards EnterpriseOne, then you must notify Agile PLM. This case has restrictions in a multi-organization environment because the interface changes from centralized (Agile PLM) to decentralized (JD Edwards EnterpriseOne).

ECO is implemented completely in JD Edwards EnterpriseOne. You can complete this step manually or set the scheduler to update the ECO in Agile PLM. Because an ECO may be created in multiple branch/plants in JD Edwards EnterpriseOne and can have a different status in each branch/plant, carry out this operation only when the ECO is completely implemented in all the branch/plants for which it was created in JD Edwards EnterpriseOne.

To transfer change order status information from JD Edwards EnterpriseOne to Agile PLM, you run an extract program (R3013D) in JD Edwards EnterpriseOne. This program generates an XML file that contains the data to be uploaded into Agile PLM.

The R3013D program retrieves ECOs that have changed since the last successful run (the last successful run date and time are stored in an IntegrationTimeStamp table - F0095 for this Universal Batch Engine [UBE]) or the As of Date specified in the processing option.

The system writes this file to a folder on the JD Edwards EnterpriseOne Enterprise server. The location of this folder is specified in the processing options of the extract program. The details of the extract programs and files are discussed in later sections of this chapter.
After creating the XML files, you can use one of two methods to pass that data to Agile PLM:

- File Transfer Protocol (FTP)
- Weblogic Server

If you use the FTP method, you must configure an FTP adapter to monitor the JD Edwards EnterpriseOne Enterprise Server folders for newly created extract files. When the FTP adapter locates a new file, the appropriate Mediator process debatches the files into separate instances. You use debatching to split large XML files into several smaller XML files.

Alternatively, you can move the XML files to the JD Edwards EnterpriseOne JDEE1In folder on the Weblogic Server. If you use this method, the appropriate Mediator process detects the file and debatches it into separate instances.

After the XML files are debatched, routing services route each XML file to UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl.

The UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl performs the following actions:

- Transforms an ECO application business message (ABM) to an ECO enterprise business message (EBM)
- Invokes EngineeringChangeOrderEBS, which routes EBM to provider Application Business Connector Services (ABCS):
  UpdateEngineeringChangeOrderListAgileProvABCSImpl

The Agile ABCS provides a call back response to EngineeringChangeOrderResponseEBS. UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl receives this response, which updates the IntegrationTimeStamp table with the last successful run date and time.

---

**Note:** If you use the FTP method, you should configure an FTP server on the JD Edwards EnterpriseOne Enterprise server. Configure the FTP adapter to use the FTP connection that is set up on the JD Edwards EnterpriseOne server.

---

### 6.2 ECO Implementation Process Integration

This section discusses:

- Updating ECO flow
- Changing status operation
- Updating ECO orchestration

#### 6.2.1 Updating ECO Flow

The Update ECO process flow includes these steps:

1. The requester ABCS, defined as an asynchronous process, receives a list of ABMs from the JD Edwards EnterpriseOne UBE batch extracts.

   The list of ABMs is those that have update_date greater than the last_successful_run_date of the batch program.
2. The requester BPEL process filters the list of ABMs based on the ID to a list of ABMs those IDs are present in the XREF tables in the Fusion Middleware (FMW) layer.

   This filtering provides a list of change orders that were actually from Agile PLM alone. The original list of ABMs may be from non-Agile PLM sources as well.

3. The BPEL process then makes a Mediator call out for endpoint virtualization.

4. The ABM is transformed to an EBM.

5. An asynchronous request-delayed response call is made to the EngineeringChangeOrderEBS with the EngineeringChangeOrderListEBM.

6. The BPEL instance is invoked when the asynchronous call gets back from the provider and provides the status of the transaction back to the caller.

   Performance is not affected because the call is scheduled and invoked by a server not by a user.

7. The concurrent program logs the status of this call.

   Figure 6–1 illustrates the update ECO flow from JD Edwards EnterpriseOne to Agile PLM:

---

**Figure 6–1 Update ECO flow from JD Edwards EnterpriseOne to Agile PLM**

---

### 6.2.2 Changing Status Operation

When there is a change of status in JD Edwards EnterpriseOne, the status of the workflow in Agile PLM is updated through the update ECO flow.

The update ECO provider service in Agile PLM does the following:

1. Whenever a change occurs to the status change of any ECO in JD Edwards EnterpriseOne, JD Edwards EnterpriseOne sends the ECO status to Agile PLM.

2. UpdateEngineeringChangeOrderListAgileProvABCImpl service checks the status of the ECO sent by JD Edwards EnterpriseOne and updates the status field in the ECO in Agile PLM.
3. If the ECO that comes in from JD Edwards EnterpriseOne has the Implemented status, the system pushes the change in Agile PLM for the ECO to the Implemented status in addition to updating the status field in ECO in Agile PLM.

For this, the entry in the CHANGE_STATUS table is used.

If no entry corresponds to the event change implementation and the change type is the same as the change type of the ECO in Agile PLM, such as ECO or Site change Order (SCO), only then is the change moved to the next status; otherwise, the change is not moved forward to the next status at all.

4. The CHANGE_STATUS table is located in the AIA Schema created on the Service Oriented Architecture (SOA) server.

The default user ID is plmpip. The password is the same password that the customer has setup for the FP AIA database. This value is the same as the property fp.db.aia.password, which can be found in the AIAInstallProperties.xml file. The DB details, such as URL, port, server ID, and so on can be found in the AIAInstallProperties.xml file at <AIA_INSTANCE>/config/AIAInstallProperties.xml.

### CHANGE_STATUS Sample Data

<table>
<thead>
<tr>
<th>EVENT</th>
<th>OUTCOME</th>
<th>SUBCLASS</th>
<th>WORKFLOW</th>
<th>NEXT_STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Implemented</td>
<td>SUCCESS</td>
<td>ECO</td>
<td>Default Change Orders</td>
<td>Implemented</td>
</tr>
</tbody>
</table>

- The EVENT column key is Change Implemented, and the OUTCOME column key is SUCCESS.
- In the SUBCLASS column, set up the change order subclass as ECO or SCO.
- In the WORKFLOW column, set up the workflows.
  - For example, if you are using Default Change Order workflow for ECO, enter Default Change Order (the values can be picked up from Agile Java client).
- In the NEXT_STATUS column, enter the status of the workflow you want it to move to when the conditions are met.

### Sample Use Case

1. Release an ECO, C0001, from Agile PLM to JD Edwards EnterpriseOne.
2. In JD Edwards EnterpriseOne, create this ECO in three branch/plants, BP10, BP20, and BP30, using the new part and product release process.
3. Change the status of this ECO in 10 to Implemented in JD Edwards EnterpriseOne.
4. JD Edwards EnterpriseOne sends this implemented ECO status to Agile PLM.
5. When it is received on the Agile PLM side, the ECO data is updated and workflow status of ECO is changed.

The change of workflow status is based on the rule: When the status of ECO is implemented in any of the branch plants in JD Edwards EnterpriseOne, move the ECO in Agile PLM to the Implemented status.

The status field in Agile PLM is also updated with the Implemented status.
6. If an entry in the CHANGE_STATUS table corresponds to the event Change Implemented and SubClass ECO, and to workflow mentioned on ECO C0001 in Agile PLM, read the next status and send it to the next status mentioned.

7. If the data mentioned has no entry in the table, the system does not attempt to send ECO C0001 to the next status.

6.2.3 Updating ECO Orchestration

Table 6–2 lists the activities involved in ECO Update orchestration:

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Run R3013D to export ECOs</td>
<td>This activity occurs when ECO-related information is updated in JD Edwards EnterpriseOne</td>
</tr>
<tr>
<td>2.</td>
<td>EngineeringChangeOrderListExtractJDEE1FIPCons or EngineeringChangeOrderListExtractJDEE1FileCons</td>
<td>EngineeringChangeOrderListExtractJDEE1Consumer is triggered</td>
</tr>
<tr>
<td>3.</td>
<td>Invoke the UpdateEngineeringChangeOrderListJDEE1ReqABCImpl process</td>
<td>UpdateEngineeringChangeOrderListEBM is created inside UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl</td>
</tr>
<tr>
<td>4.</td>
<td>UpdateEngineeringChangeOrderListJDEE1ReqABCImpl invokes the EngineeringChangeOrderEBS with UpdateEngineeringChangeOrderList operation</td>
<td>An invoke activity in UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl invokes the UpdateEngineeringChangeOrderList operation on EngineeringChangeOrderEBS with UpdateEngineeringChangeOrderListEBM as the input</td>
</tr>
</tbody>
</table>
6.3 Solution Assumptions and Constraints

This design assumes that the following statements are true:

1. The ECO update flow is only from JD Edwards EnterpriseOne to Agile PLM.
2. This integration does not support updates to ECOs in Agile PLM.
3. This integration does not support sending ECO parts list and related items information from JD Edwards EnterpriseOne.
4. The JD Edwards EnterpriseOne IntegrationTimeStampManager business service updates the last batch extract date and time to the JD Edwards EnterpriseOne server date and time.
5. If you change an ECO status in JD Edwards EnterpriseOne, the system updates Agile PLM.
6. When Agile PLM is configured for multiple sites (and when multiple sites are attached to an ECO), a single ECO in Agile PLM creates multiple pre-approved ECOs in JD Edwards EnterpriseOne.

In this integration, all the pre-approved ECOs go into an implemented state when the scheduled batch, R30510 (BOM Creation batch), is run. In case of any failures, such as if an ECO is not implemented, the user should ensure that the failed ECO is implemented before the update ECO flow is triggered; Agile PLM considers the ECO to be implemented if any one of the corresponding ECOs at the JD Edwards EnterpriseOne end goes to an implemented state. Alternatively, if users want to send the failure status to Agile PLM, they must manually configure the status on P48020 to reflect the failure. To do this, create a new status user-defined code item for the error and configure the ECO_STATUS_CODE DVM, and then you can send the changes during the subsequent run of the update ECO flow. The failure status appears only in the Page 2 Status Flex field (you must configure this), while the workflow status for the ECO may still appear as Implemented.

7. The system does not send any effectivity date changes in JD Edwards EnterpriseOne back to Agile PLM.
6.4 Agile PLM Interfaces

Table 6–3 lists the Agile PLM Web Services Definition Language (WSDL) files:

<table>
<thead>
<tr>
<th>Interface Description</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used to update change order information in Agile PLM.</td>
<td>UpdateEngineeringChangeOrderListAgileProvABCSImpl</td>
</tr>
<tr>
<td>UpdateEngineeringChangeOrderListAgileProvABCSImpl</td>
<td>ChangeABS.wsdl</td>
</tr>
<tr>
<td></td>
<td>ChangeMerge.wsdl</td>
</tr>
</tbody>
</table>

Table 6–4 lists the Agile PLM XML Schema Definition (XSD) files:

<table>
<thead>
<tr>
<th>Interface Description</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains UpdateEngineeringChangeOrderListABM and UpdateEngineeringChangeOrderListResponseABM</td>
<td>EngineeringChangeOrderA BM.xsd</td>
</tr>
<tr>
<td></td>
<td>EngineeringChangeOrderA BO.xsd</td>
</tr>
</tbody>
</table>

6.5 JD Edwards EnterpriseOne Interfaces

Table 6–5 lists the JD Edwards EnterpriseOne WSDL files:

<table>
<thead>
<tr>
<th>Interface Description</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>IntegrationTimeStampManager.wsdl</td>
<td>IntegrationTimeStampManager.wsdl</td>
</tr>
<tr>
<td></td>
<td>Update EngineeringChangeOrderListJDEE1ReqABCSImpl</td>
</tr>
</tbody>
</table>

Table 6–6 lists the JD Edwards EnterpriseOne XSD files:

<table>
<thead>
<tr>
<th>Interface Description</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains the Update ECO Request ABM</td>
<td>UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl</td>
</tr>
<tr>
<td></td>
<td>R3013D.xsd</td>
</tr>
</tbody>
</table>

6.6 Core AIA Components

Table 6–7 lists the industry components for process integration for updating ECO flow:

<table>
<thead>
<tr>
<th>Component</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBO</td>
<td>EngineeringChangeOrderEBO</td>
</tr>
<tr>
<td>EBM</td>
<td>UpdateEngineeringChangeOrderListEBM</td>
</tr>
<tr>
<td></td>
<td>UpdateEngineeringChangeOrderListResponseEBM</td>
</tr>
<tr>
<td>EBS</td>
<td>EngineeringChangeOrderEBS</td>
</tr>
</tbody>
</table>
Table 6–8 lists the locations of components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBO and EBM XSD files</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/</td>
</tr>
<tr>
<td>WSDL files</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/</td>
</tr>
</tbody>
</table>

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in Oracle Enterprise Repository.

For more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link, see Oracle Fusion Middleware Developer’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are protected, and remains intact after a patch or an upgrade.

For more information, see Oracle Fusion Middleware Developer’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1.

### 6.7 Agile PLM and JD Edwards EnterpriseOne Components for Update ECO

Table 6–9 lists the Agile PLM and JD Edwards EnterpriseOne components for Update ECO:

<table>
<thead>
<tr>
<th>Service</th>
<th>JD Edwards EnterpriseOne (Requester)</th>
<th>Agile PLM (Provider)</th>
</tr>
</thead>
</table>
| ABM     | R3013D                               | AgileUpdateEngineerin

Table 6–10 lists the locations of the core components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSDL files</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/Agile/V1/wsdls</td>
</tr>
<tr>
<td></td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/JDEE1/V1/wsdls</td>
</tr>
</tbody>
</table>
6.8 Integration Services

The process integration for ECO implementation from JD Edwards EnterpriseOne to Agile PLM uses these industry components:

- EngineeringChangeOrderEBS
- UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl
- UpdateEngineeringChangeOrderListAgileProvABCSImpl

6.8.1 EngineeringChangeOrderEBS

EngineeringChangeOrderEBS is an EBS, which exposes the operations related to the EngineeringChangeOrder Integration on the EngineeringChangeOrder EBO. The routing rules are:

- EngineeringChangeOrderEBS
  - UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl: Route UpdateEngineeringChangeOrderListEBM to UpdateEngineeringChangeOrderListAgileProvABCSImpl
- EngineeringChangeOrderResponseEBS
  - UpdateEngineeringChangeOrderListAgileProvABCSImpl: Route UpdateEngineeringChangeOrderListResponseEBM to UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl

6.8.2 UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl

UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl transforms the JD Edwards EnterpriseOne message (UpdateEngineeringChangeOrderABM) into UpdateEngineeringChangeOrderListEBM, calls the routing service to update ECO, and gets the UpdateEngineeringChangeOrderListResponseEBM response from Agile PLM. It then invokes the integration time stamp manager service to update the integration time stamp table with the last runtime and date.

Process flow:
1. Picks up the xml file based on the properties set up in file/ftp consumer
2. Receives UpdateEngineeringChangeOrderABM from file/ftp consumer
3. Transforms JD Edwards EnterpriseOne-specific UpdateEngineeringChangeOrderABM into UpdateEngineeringChangeOrderListEBM
4. Populates the EBM header

### Table 6–10 (Cont.) Location of components

<table>
<thead>
<tr>
<th>Component</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Business Objects, ABM, and Common XSD files</td>
<td>$AIA_HOME/AIAMetaData/AIAComponents/Application ObjectLibrary/Agile/V1/schemas/ $AIA_HOME/AIAMetaData/AIAComponents/Application ObjectLibrary/JDEE1/V1/schemas/</td>
</tr>
</tbody>
</table>
5. Calls the EngineeringChangeOrderEBS service to send message
   UpdateEngineeringChangeOrderListEBM to
   UpdateEngineeringChangeOrderAgileProvABCSImpl

6. Receives UpdateEngineeringChangeOrderListResponseEBM from
   EngineeringChangeOrderResponseEBS

7. Calls the Integration Time Stamp Manager service to update the integration time
   stamp table with the last runtime and date

8. Transforms UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl as follows
   Xform_EngineeringChangeOrderListABMReqMsg_To_
   EngineeringChangeOrderListListEBMReqMsg

6.8.3 UpdateEngineeringChangeOrderListAgileProvABCSImpl

The UpdateEngineeringChangeOrderListAgileProvABCSImpl updates an ECO in
Agile PLM. In this integration,
UpdateEngineeringChangeOrderListAgileProvABCSImpl updates the status-related
fields. It is implemented as an asynchronous process.

Process flow:

1. EngineeringChangeOrderEBS with UpdateEngineeringChangeOrderListReqMsg,
   which contains UpdateEngineeringChangeOrderListEBM as input, invokes
   UpdateEngineeringChangeOrderListAgileProvABCSImpl

2. The system calls the transform operation to convert the
   UpdateEngineeringChangeOrderListEBM into
   AgileUpdateEngineeringChangeOrderListABM

3. The system passes AgileUpdateEngineeringChangeOrderListABM as input to the
   web service operation, which carries out the following functions for this
   integration:
      Update Status-related attributes on ECO (for Agile PLM to JD Edwards
      EnterpriseOne flow)

4. After Coarse Grained API UpdateChange successfully executes,
   AgileUpdateEngineeringChangeOrderListResponseABM is received

5. If the UpdateChange service operation fails on the Agile PLM side, a new fault is
   generated and sent across with an appropriate error message
Part II

Implementing the Delivered Integrations

Part 2 contains the following chapters:

Chapter 7, "Configuring Design to Release: Agile PLM - JDE E1"
This chapter discusses:

- Prerequisites
- Setting up participating applications
- Setting up batch processing information
- SQL Script for Loading Cross-Reference Records
- Identifying cross-references
- Populating cross-references
- Describing Domain Value Maps (DVMs)
- Setting configuration properties
- Handling error
- Viewing EBO Implementation Maps (EIMs)

7.1 Prerequisites

This section discusses the prerequisites for the following integration flows:

- Initial load of items and Bill of Materials (BOM)
- Product design Changes
- ECO implementation
- Item attributes and cost updates
- Item balance updates

7.1.1 Prerequisites for Initial Load of Items and BOM

Before running an initial load, complete the following actions:

- Set up versions of each extract program in JD Edwards EnterpriseOne.
- Set up batch processing information and invoke utilities for an initial load.
- Create a new workflow for initial load change orders and set privileges.

7.1.2 Prerequisites for Product Design Changes

Before running the Create ECO flow, complete the following actions:
7.1.3 Prerequisites for ECO Implementation
Before running the Update ECO flow, complete the following steps:
- Set up versions of each extract program in JD Edwards EnterpriseOne.
- Set up batch processing information.

7.1.4 Prerequisites for Item Attributes and Cost Updates
Before running the update item flow, you must complete these steps:
- Set up versions of each extract program in JD Edwards EnterpriseOne.
- Set up the batch processing information.

7.1.5 Prerequisites for Item Balance Updates
Before running the update item balance flow, complete the following actions:
- Set up versions of each extract program in JD Edwards EnterpriseOne.
- Set up batch processing information.

7.2 Setting Up Participating Applications
Before integration, you must set up Agile Product Lifecycle Management (PLM), JD Edwards EnterpriseOne, Oracle Web Services Manager Security Information (OWSM), and Application Integration Architecture (AIA).

7.2.1 Setting Up Agile Content Services
The following set ups are required using the Agile PLM Java client:
1. Create new destinations
2. Create new events for Engineering Change Order (ECO) and Site Change Order (SCO)
3. Define filters
4. Create new subscribers for ECO and SCO
5. Set ECO and SCO privileges
6. Set up Agile item quantity attributes
7. Create a new ECO workflow for initial load change orders
8. Set ECO workflow privileges
9. Set up Change_Status table for workflow
10. Setup Agile notifications

7.2.1.1 Create New Destinations
To create a new JMS destination:
If Agile PLM is deployed on Oracle Application Server (OAS)
1. Copy the wthint3client.jar file (this file is located at FMW’s $WLS_HOME/Middleware/wlserver_10.3/server/lib) put under the OAS j2ee\home\applib directory in the Agile PLM environment.
2. Bounce the Agile Oracle Application Server.
3. Restart the complete SOA server using the commands from the <SOA_HOME>/opmn/bin directory.
   For Linux (x86), Solaris SPARC (64-bit), IBM AIX Based Systems (64-bit), and HP-UX 11i (64 bit)
   ■ ./opmnctl.exe stopall
   ■ ./opmnctl.exe startall
   For Microsoft Windows (32-bit)
   ■ opmnctl.exe stopall
   ■ opmnctl.exe startall
If Agile PLM is deployed on the Weblogic server (WLS), no need to copy the wthint3client.jar file, it should work fine as it is.

For more information on how to restart Agile PLM deployed on the Weblogic server (WLS), refer to the Agile Product Lifecycle Management Installing Agile PLM on WebLogic Server Guide.
1. On the Admin tab, navigate to System Settings > Agile Content Service > Destinations.
2. Select Protocol JMS.
3. Enter or set the essential values as illustrated in Table 7–1:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Tabs</td>
<td>Add Sites, Title Block, Page Two, Page Three, BOM, Manufacturers</td>
</tr>
<tr>
<td>BOM Options</td>
<td>Tabs and Items</td>
</tr>
<tr>
<td>BOM Levels</td>
<td>Select the All Levels check box</td>
</tr>
<tr>
<td>AML Options</td>
<td>Tabs and Manufacturer Parts</td>
</tr>
<tr>
<td>Attachment Options</td>
<td>Tab only</td>
</tr>
</tbody>
</table>

4. Click Test to validate.

7.2.1.2 Create New Events for ECO and SCO
To create new events for ECO and SCO:
1. On the Admin tab, navigate to System Settings > Agile Content Service > Events.
2. Set the essential values as illustrated in Table 7–2:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>For ECO</td>
<td>For SCO</td>
</tr>
</tbody>
</table>
7.2.1.3 Define Filters

To define filters:

1. On the Admin tab, navigate to System Settings > Agile Content Service > Filters.
2. Modify the default item filter to set the parameters as illustrated in Table 7–3:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Tabs</td>
<td>Add Sites, Title Block, Page Two, Page Three, BOM, Manufacturers</td>
</tr>
<tr>
<td>BOM Options</td>
<td>Tabs and Items</td>
</tr>
<tr>
<td>BOM Levels</td>
<td>Select the All Levels check box</td>
</tr>
<tr>
<td>AML Options</td>
<td>Tabs and Manufacturer Parts</td>
</tr>
<tr>
<td>Attachment Options</td>
<td>Tab only</td>
</tr>
</tbody>
</table>

3. Modify the filters to set the given fields as illustrated in Table 7–4:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Default Change Order Filter</td>
</tr>
<tr>
<td>Affected Items Options</td>
<td>Tab and Items</td>
</tr>
<tr>
<td>Redline Changes only</td>
<td>No</td>
</tr>
</tbody>
</table>

7.2.1.4 Create New Subscribers for ECO and SCO

To create new subscribers for ECO and SCO:

1. On the Admin tab, navigate to System Settings > Agile Content Service > Subscribers
2. Create new subscribers, one each for ECO and SCO, and set the parameters as illustrated in Table 7–5:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Define your own</td>
</tr>
</tbody>
</table>
3. Enter or set the subscriber details for each ECO and SCO by adding a new row.

4. Use the information listed in Table 7–6 for creating a new row:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Define your own</td>
</tr>
<tr>
<td>Subclass</td>
<td>ATO</td>
</tr>
<tr>
<td>Workflow</td>
<td>Default ATOs</td>
</tr>
<tr>
<td>Criteria</td>
<td>All Site Change Orders</td>
</tr>
<tr>
<td></td>
<td>All Change Orders</td>
</tr>
<tr>
<td>AutoNumber</td>
<td>ATO Number</td>
</tr>
<tr>
<td>Event</td>
<td>Select the Name of the Event that you created in step 2 for ECO</td>
</tr>
<tr>
<td></td>
<td>Select the Name of the Event that you created in step 2 for SCO</td>
</tr>
</tbody>
</table>

5. Enable all the newly created subscribers

7.2.1.5 Set ECO and SCO Privileges
To set ECO and SCO privileges:

1. On the Admin tab, navigate to User Settings > Privileges > Modify
2. Create new Modify Privileges for ECO and SCO
3. Set privilege to Modify
4. Select the criteria that correspond to each ECO and SCO
5. Select all the attributes, including the invisible and disabled attributes in the Applied to field and save them
6. On the Where Used tab, add roles to all the privileges you created
   The default role is Admin user.

   **Note:** The user should have privileges to modify the Released items and the Released changes.

7.2.1.6 Set up Agile Item Quantity Attributes
To set up Agile item quantity attributes:
1. On the Admin tab, navigate to Data Settings > Classes

2. For both Parts and Document classes, enable the flex fields on Page2, Page3, or Site tab according to the MultiSite_Enabled property value in AIAConfigProperties.xml for Agile Product Lifecycle Management (PLM)

3. The field names, listed in Step 2, in Agile PLM reflect the following fields from JD Edwards EnterpriseOne:
   - Unit Cost
   - Available Quantity
   - On Hand Quantity
   - Reserved Quantity
   The flex fields should be the same as those entered as values for the following properties:
   - Item.UnitCostAttribute
   - Item.AvailableQuantityAttribute
   - Item.OnHandQuantityAttribute
   - Item.ReservedQuantityAttribute

4. Select Visible for the fields you created.

   **Note:** Ensure that these attributes have Read and Modify privileges.

5. Click OK.

### 7.2.1.7 Create a New Workflow for Initial Load Change Orders

To create a new workflow for initial load change orders:

1. On the Admin tab, navigate to Workflow Settings > Workflows

2. Create new workflow for the initial load change orders and set the values as illustrated in Table 7–7:

   **Table 7–7** Values for creating new workflow

<table>
<thead>
<tr>
<th>Field</th>
<th>Value/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Define your own</td>
</tr>
<tr>
<td>Workflow Criteria matching Type</td>
<td>All</td>
</tr>
<tr>
<td>Status Criteria Matching Type</td>
<td>Same</td>
</tr>
<tr>
<td>Object Type</td>
<td>Changes</td>
</tr>
<tr>
<td>Matching Criteria</td>
<td>All Change Orders</td>
</tr>
</tbody>
</table>

3. Go to the Status tab and add two new workflow statuses and set the values as illustrated in Table 7–8 and Table 7–9:
4. Add workflow criteria for each of the statuses created earlier and set the select criteria as All Change Orders.

5. Enable the newly created workflow.

### 7.2.1.8 Set ECO Workflow Privileges

To set ECO workflow privileges:

1. On the Admin tab, navigate to User Settings > Privileges > Change Status

2. Create new change status privileges for initial load change orders workflow and set the values as illustrated in Table 7–10:

### Table 7–8 Values for workflow status 1

<table>
<thead>
<tr>
<th>Field</th>
<th>Value/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Define your own</td>
</tr>
<tr>
<td>Status Type</td>
<td>Pending</td>
</tr>
<tr>
<td>Status Stamp Color</td>
<td>Define your own</td>
</tr>
</tbody>
</table>

### Table 7–9 Values for workflow status 2

<table>
<thead>
<tr>
<th>Field</th>
<th>Value/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Define your own</td>
</tr>
<tr>
<td>Status Type</td>
<td>Released</td>
</tr>
<tr>
<td>Status Stamp Color</td>
<td>Define your own</td>
</tr>
</tbody>
</table>

3. On the Where Used tab, add roles to all the privileges you created.
   
   The default role is Admin user.

### 7.2.1.9 Setup Change_Status Table for Workflow

Because the change status name depends on the workflow being used, the CHANGE_STATUS configuration table is provided as part of the integration setup. The CHANGE_STATUS table is located in the AIA schema created on the Service-Oriented Architecture (SOA) server. The default user ID is plmpip. The password is the same password that the customer has setup for the FP AIA database. This value is the same as the property fp.db.aia.password, which can be found in the

### Table 7–10 Values for creating new workflow

<table>
<thead>
<tr>
<th>Field</th>
<th>Value/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Define your own</td>
</tr>
<tr>
<td>Description</td>
<td>Define your own</td>
</tr>
<tr>
<td>Enabled</td>
<td>Yes</td>
</tr>
<tr>
<td>Privilege</td>
<td>Change Status</td>
</tr>
<tr>
<td>Criteria</td>
<td>All Change Orders</td>
</tr>
<tr>
<td>Workflow</td>
<td>New workflow created in Step 9</td>
</tr>
<tr>
<td>Status - From</td>
<td>Status defined in Step 9 with Status type of Pending</td>
</tr>
<tr>
<td>Status - To</td>
<td>Status defined in Step 9 with Status type of Released</td>
</tr>
</tbody>
</table>
AIAInstallProperties.xml file. The DB details, such as URL, port, server ID, and so on can be found in the AIAInstallProperties.xml file at `<AIA_INSTANCE>/config/AIAInstallProperties.xml`.

This table allows the administrators to specify the next status for each possible combination of change object type and the workflow being used for each event that qualifies for the change status operation. The administrators can add as many rows as required. The number of sub-classes of change objects and the number of workflows is unlimited in Agile PLM.

Table 7–11 illustrates how a partially configured table looks:

<table>
<thead>
<tr>
<th>EVENT</th>
<th>OUTCOME</th>
<th>SUBCLASS</th>
<th>WORKFLOW</th>
<th>NEXT_STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Part Request</td>
<td>Success</td>
<td>ECR</td>
<td>Default Change Requests</td>
<td>Completed</td>
</tr>
<tr>
<td>New Part Request</td>
<td>Failure</td>
<td>ECR</td>
<td>Default Change Requests</td>
<td>Pending</td>
</tr>
<tr>
<td>Pre-release Audit</td>
<td>Failure</td>
<td>ECO</td>
<td>Default Change Orders</td>
<td>Submitted</td>
</tr>
<tr>
<td>Pre-release Audit</td>
<td>Failure</td>
<td>ECO</td>
<td>Fast-track Change Orders</td>
<td>Pending</td>
</tr>
<tr>
<td>Change Implementation</td>
<td>Success</td>
<td>ECO</td>
<td>Default Change Orders</td>
<td>Implemented</td>
</tr>
<tr>
<td>Change Implementation</td>
<td>Success</td>
<td>ECO</td>
<td>Fast-track Change Orders</td>
<td>Implemented</td>
</tr>
</tbody>
</table>

The EVENT column key is **Change Implemented**.

The OUTCOME column key is **SUCCESS**.

In the SUBCLASS column, set up the change order subclass as ECO, MCO, etc.

In the WORKFLOW column, set up the workflows. For example, if you are using Default Change Order workflow for ECO, then enter Default Change Order in this column. These values can be picked up from Agile PLM Java client.

In the NEXT_STATUS column, enter the status of the workflow you want it to move to when the conditions are met. For example, when a Change is implemented (identified by event - this is a key. do not change it) and the OUTCOME is SUCCESS (identified by outcome, gets checked while changing status here), and the WORKFLOW chosen is Default Change Orders, the set up requesting the Change to be moved to the Implemented status (NEXT_STATUS).

In a new installation, this table is empty. The integration administrator needs to add rows in this table if the change status process needs to be supported for any of the given processes.

- The value of the EVENT column must be limited to the values corresponding to the processes supported: New Part Request, Pre-release Audit, and Change Implementation. The change implementation process can also be associated with a status update back to a flexfield on the change order.
- The value of the OUTCOME column must be limited to Success and Failure.
- The value of the SUBCLASS column must be limited to the set of subclasses of the Change Requests Class, Change Orders Class, Manufacturing Orders Class, and Site Change Orders Class.

- The value of the WORKFLOW column must be limited to the set of workflows for the selected subclass. For example, if you are using the default change order workflow for ECO, then enter the default change order in this column. These values can be picked up from the Agile PLM Java client.

- In the NEXT_STATUS column, enter the status of the workflow to move to when the conditions are met. For example, when a change is implemented (identified by event; this is a key, so do not change it) and the OUTCOME is SUCCESS (identified by outcome; this is checked while changing status here) and the WORKFLOW chosen is Default Change Orders, the setup requesting the change to be moved to the implemented status (NEXT_STATUS).

### 7.2.1.10 Set Up Agile Notifications

Notifications can be setup in Agile PLM to send email when any error occurs during ACS processing in the change order release process integration flow.

To set up the notifications:

1. Log in to Agile Java Client.
2. In the Admin tab, navigate to System Settings > Agile Content Service > Destinations.
3. Open the destination that was configured for the change order release flow to send the data to the JMS queue.
4. In the Notification User field, add the users and user groups who must be notified on failure of ACS transfer.
5. Save the changes.

### 7.2.2 Setting Up JD Edwards EnterpriseOne

Before you can use the Design to Release Agile PLM - JD Edwards EnterpriseOne integration; you must set up several applications in JD Edwards EnterpriseOne that the integration uses. The integration processes and flows supported in this integration require that JD Edwards EnterpriseOne is set up correctly before any initial load or updates occur in Agile PLM.

This section discusses:

- Setting up batch programs
- Setting up integration user-defined codes (UDCs)
- Setting up JD Edwards EnterpriseOne web services
- Setting up business services (BSSVs)
- Configuration parameters for creating Engineering Change order (ECO) flow

#### 7.2.2.1 Setting Up Batch Programs

Setting up batch programs includes:

- Setting up the item initial load extract to XML program (R4101D3)
- Setting up the Bill of Material (BOM) initial load extract to XML program (R3002D)
- Setting up ECO batch extract programs
Setting Up Participating Applications

- Setting up the ECO extract to XML program (R3013D)
- Setting up item list batch extract programs
- Setting up the item list extract to XML program (R4102D)
- Setting up item balance batch extract programs
- Setting up the item balance extract to XML program (R41021D)

For more information, see JD Edwards EnterpriseOne Tools Development Tools: Batch Versions Guide.

**Setting Up Item Initial Load Extract to XML Program (R4101D3)**

Use this program to extract the initial load item data from JD Edwards EnterpriseOne to Agile PLM. Before running this program, you must set the processing options to ensure that data is extracted correctly.

On the Processing Options tab, set up the path where the xml file is written.

Use this processing option to specify the file location that the system uses for storing the resulting XML file. The name of the resulting XML file is in the format of R4101D3_MMDDYY_hhmms.xml. The file name is appended to the value entered in this processing option to determine the fully qualified path and name. All directories specified in the path should exist. Also, ensure that the value ends with ‘\’ or ‘/’ as required by the operating system. If you leave this processing option blank, the system writes the resulting XML file to the directory where this report is running.

**Setting Up BOM Initial Load Extract to XML Program (R3002D)**

Use this program to extract initial load BOM data from JD Edwards EnterpriseOne to Agile PLM. Before running this program, you must set the processing options to ensure that data is extracted correctly.

This program has two default versions:

- XJDE0001 - BOM Initial Load Extract to XML
  We recommend this version for Agile PLM with Multi-Site OFF.

  Figure 7–1 illustrates an example of the recommended data selection for the Universal Batch Engine (UBE) program:

  ![Figure 7–1 Example of data selection for the batch program](image)

- XJDE0002 - BOM Initial Load - Exclude Non-Stock Components
  We recommend this version for Agile PLM 39 with Multi-Site ON.
Figure 7–2 illustrates the recommended data selection:

**Figure 7–2  Example of data selection**

Set the following processing options on the Processing Options tab:

- **Path where the XML file is written**
  
  Use this processing option to specify the file location that the system uses for storing the resulting XML file.
  
  The name of the resulting XML file is in the format of R3002D/MMDDYY_hhmmss.xml.
  
  The system appends the file name to the value entered in this processing option to determine the fully qualified path and name. All directories specified in the path should already exist. If you leave this processing option blank, the system writes the resulting XML file to the directory where this report is running.
  
- **As of Date**
  
  This date is used for effectivity checking. Enter a specific date to display documents (orders, BOMs, and routings, as applicable) that are effective on or after that date. The current system date is the default, but you can enter any future or past date.
  
- **Selection for Components**
  
  Use this processing option to specify if the system selects all the components associated with a selected BOM. If you leave this processing option blank, the system only selects components with the same branch/plant as the parent. The system writes skipped components details to the report output. The system selects all associated components for the BOMs selected if set to the value 1.
  
- **Extract Parent Item Revision Level**
  
  Use this processing option to indicate whether the system extracts the BOM parent item revision level from the item master (F4101). If you leave this processing option blank, the system does not extract the parent item revision level. The system extracts the parent item revision level if the value is set to 1.

**Setting Up ECO Batch Extract Programs**

The ECO batch extract program extracts ECO changes from JD Edwards EnterpriseOne to Agile PLM. Before the ECOs are processed using batch programs, you must configure versions for each of them.
The Update ECO program includes the Update ECO Extract to XML program (R3013D).

For more information, see *JD Edwards EnterpriseOne Tools Development Tools: Batch Versions Guide*.

**Setting Up ECO Extract to XML Program (R3013D)**

Before running this program, set the processing options to ensure that data is extracted correctly.

This program has one default version: XJDE0001 - ECO Extract to XML

ECO Extract R3013D has no default data selection.

Set the following processing options on the Processing Options tab:

- **Path where the XML File is written**
  
  Use this processing option to specify the file location that the system uses for storing the resulting XML file. The name of the resulting XML file is in the format of R3013D_MMDYY_hhmmss.xml. All directories specified in the path should already exist. If you leave this processing option blank, the system writes the resulting XML file to the default JD Edwards EnterpriseOne system directory.

- **As of Date**
  
  This date is used for effectivity checking. Enter a specific date to extract the updated ECOs from JD Edwards EnterpriseOne that are effective on or after that date. If you leave this processing option blank, the system uses the integration time stamp date.

- **Integration Code**
  
  Use this processing option to specify the integration code, which is a mandatory field in UDC (00/IE). If you leave this processing option blank, the system will not process any data or write the XML file.

**Setting Up Item List Batch Extract Programs**

The item list batch extract program extracts item attribute and cost changes from JD Edwards EnterpriseOne to Agile PLM.

Before the Item List can be processed using batch programs, you should configure versions for each program.

Item list programs include the Item List Extract to XML program (R4102D).

For more information, see *JD Edwards EnterpriseOne Tools Development Tools: Batch Versions Guide*.

**Setting Up Item List Extract to XML Program (R4102D)**

Use this program to extract item list data from JD Edwards EnterpriseOne to Agile PLM. Before running this program, set the processing options to ensure that data is extracted correctly.

This program has one default version: XJDE0001 - Item List Extract to XML.

This program has no default data selection.

Set the following processing options on the Processing Options tab:

- **Path where the XML File is written**
  
  Use this processing option to specify the file location that the system uses for storing the resulting XML file. The name of the resulting XML file is in the format
of R4102D_MMDDYY_hhmmss.xml. All directories specified in the path should already exist. If you leave this processing option blank, the system will write the resulting XML file to the default JD Edwards EnterpriseOne system directory.

- As of Date
  This date is used for effectivity checking. Enter a specific date to extract the updated ECOs from JD Edwards EnterpriseOne that are effective on or after that date. If you leave this processing option blank, the system will use the integration time stamp date.

- Integration Code
  Use this processing option to specify the integration code, which is a mandatory field in UDC (00/IE). If you leave this processing option blank, the system will not process any data and it will not write an XML file.

- Ledger Type
  Use this processing option to specify the costing method. If you leave this processing option blank, the system will use the cost method from branch/plant constants. If you do not set the cost method in branch/plant constants, then the system will use the default value of data dictionary item CSMT.

**Setting Up Item Balance Batch Extract Programs**

The item balance list batch extract program extracts item availability changes from JD Edwards EnterpriseOne to Agile PLM.

Before the item balance can be processed using batch programs, you should configure versions for each program.

Item balance programs include the Item Balance Extract to XML program (R41021D).

For more information, see *JD Edwards EnterpriseOne Tools Development Tools: Batch Versions Guide*.

**Setting Up Item Balance Extract to XML Program (R41021D)**

Use this program to extract item balance data from JD Edwards EnterpriseOne to Agile PLM. Before running this program, set the processing options to ensure that data is extracted correctly.

This program has a default version:

- XJDE0001 - Item Balance Extract to XML

  Item Balance Extract R41021D has no default data selection.

Set the following processing options on the Processing Options tab:

- Path where the XML File is written
  Use this processing option to specify the file location that the system uses for storing the resulting XML file. The name of the resulting XML file is in the format of R41021D_MMDDYY_hhmmss.xml. All directories specified in the path should already exist. If you leave this processing option blank, the system will write the resulting XML file to the default JD Edwards EnterpriseOne system directory.

- As of Date
  This date is used for effectivity checking. Enter a specific date to extract the updated ECOs from JD Edwards EnterpriseOne that are effective on or after that date. If you leave this processing option blank, the system will use the integration time stamp date.
Integration Code

Use this processing option to specify the integration code, which is a mandatory field in UDC (00/IE). If you leave this processing option blank, the system will not process any data and it will not write the XML file.

7.2.2.2 Setting Up UDCs

You set up UDCs to store the information about the types of integration codes used.

UDC: 00/IE - Integration Code

To set up a UDC, complete the following steps:

1. Navigate to the Work With User Defined Codes page using the Fast Path. See Figure 7–3

Figure 7–3 Work with User Defined Code Page

2. Enter Product Code as 00 and User Defined Codes as IE. See Figure 7–4
3. Click Add.

Enter the code and description as AGILEJDE and Agile to JDE Integration, respectively. See Figure 7–5

**Figure 7–5  Entering Code for UDC**

4. Click OK to save the UDC values. See Figure 7–6
7.2.2.3 Setting Up JD Edwards EnterpriseOne Web Services

You use these JD Edwards EnterpriseOne web services, also called business services, in the Design to Release integration:

- EngineeringChangeOrderManager (JP300000)
- IntegrationTimeStampManager (JP300010)
- EngineeringChangeOrdersProcessor (J3000010)
- EngineeringChangeOrderPartsListProcessor (J3000020)
- ProcessIntegrationTimeStamp (J3000030)
- InventoryItemsProcessor (J4100020)
- nventoryItemBranchProcessor (J4100030)

For more information about JD Edwards EnterpriseOne business services, see JD Edwards EnterpriseOne Business Services Reference Guide.

---

**Note:** JD Edwards EnterpriseOne business services must be built with the option for migration from 10g to 11g turned on.

---

For more information about the migration option, see JD Edwards EnterpriseOne Release 8.98.3 Building Business Services Packages with Migration (My Oracle Support Document ID 1233332.1).

**EngineeringChangeOrderManager (JP300000)**

The EngineeringChangeOrderManager Published Business Service (PBSSV) manages the processing of the following:

Table 7–12 lists and describes the web service operations:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EngineeringChangeOrdersProcessor (J3000010)</td>
<td>Use this operation to call other processors internally to add records into Item, Item Branch, and ECO files.</td>
</tr>
</tbody>
</table>
Setting Up Participating Applications

Configuring Design to Release: Agile PLM - JDE E1

For more information about the web services, see *JD Edwards EnterpriseOne Business Services Reference Guide*.

**IntegrationTimeStampManager (JP300010)**

The IntegrationTimeStampManager PBSSV manages the processing of the following web service operation:

Table 7–13 lists and describes the web service operations:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProcessIntegrationTimeStamp (J3000030)</td>
<td>Use this operation to add and modify the Integration Time Stamp table.</td>
</tr>
</tbody>
</table>

For more information, see *JD Edwards EnterpriseOne Business Services Reference Guide*, "IntegrationTimeStampManager."

**EngineeringChangeOrdersProcessor (J3000010)**

The EngineeringChangeOrderManager PBSSV calls the EngineeringChangeOrdersProcessor BSSV from the provider when a user creates an ECO. For the Design to Release integration, the external system is Agile PLM. The user creates an ECO in Agile PLM and releases the same to JD Edwards EnterpriseOne. This processor calls the create ECO (B3004100) business function to add a header record in F4801. This action generates an ECO number, which the system uses to create a BOM in JD Edwards EnterpriseOne.

The EngineeringChangeOrdersProcessor PBSSV manages the processing of the following web service operations:

Table 7–14 lists and describes the web service operations:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EngineeringChangeOrderPartsListProcessor (J3000020)</td>
<td>Use this operation to add ECOs.</td>
</tr>
<tr>
<td>InventoryItemsProcessor (J4100020)</td>
<td>Use this operation to add or modify item records.</td>
</tr>
<tr>
<td>InventoryItemBranchProcessor (J4100030)</td>
<td>Use this operation to add or modify item branch records.</td>
</tr>
</tbody>
</table>

For more information, see *JD Edwards EnterpriseOne Business Services Reference Guide*, "EngineeringChangeOrdersProcessor."

**EngineeringChangeOrderPartsListProcessor (J3000020)**

Use this operation to add ECOs in the ECO file.

**InventoryItemsProcessor (J4100020)**

Use this operation to add or modify item records.

**InventoryItemBranchProcessor (J4100030)**

Use this operation to add or modify item branch records.

For more information, see *JD Edwards EnterpriseOne Business Services Reference Guide*, "EngineeringChangeOrderPartsListProcessor."
The EngineeringChangeOrdersProcessor calls the EngineeringChangeOrderPartsListProcessor web service. This processor calls the create ECO (B3004100) business function to add a parts list and related items in JD Edwards EnterpriseOne.

For more information, see *JD Edwards EnterpriseOne Business Services Reference Guide*, "EngineeringChangeOrderPartsListProcessor."

**ProcessIntegrationTimeStamp (J3000030)**

The IntegrationTimeStampManager PBSSV calls the ProcessIntegrationTimeStamp web service. This processor adds and modifies the Integration Time Stamp table (F0095) with the last successful runtime.

For more information, see *JD Edwards EnterpriseOne Business Services Reference Guide*, "ProcessIntegrationTimeStamp."

**InventoryItemsProcessor (J4100020)**

The EngineeringChangeOrdersProcessor calls the InventoryItemsProcessor web service. This processor calls the F4101 ItemMasterAddition (B4101062) business function to add items in JD Edwards EnterpriseOne.

For more information, see *JD Edwards EnterpriseOne Business Services Reference Guide*, "InventoryItemsProcessor."

**InventoryItemBranchProcessor (J4100030)**

The EngineeringChangeOrdersProcessor calls the InventoryItemsProcessor web service. This processor calls the F4102 ItemBranchAddition (B4101072) business function to add item branch records in JD Edwards EnterpriseOne.

For more information, see *JD Edwards EnterpriseOne Business Services Reference Guide*, "InventoryItemsProcessor."

Table 7–15 lists and describes the operations:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EngineeringChangeOrdersProcessor</td>
<td>Use this operation to call other processors internally to add records into item, item branch, and ECO files.</td>
</tr>
<tr>
<td>EngineeringChangeOrderPartsListProcessor</td>
<td>Use this operation to add ECOs in the ECO file.</td>
</tr>
<tr>
<td>InventoryItemsProcessor</td>
<td>Use this operation to add or modify item records.</td>
</tr>
<tr>
<td>InventoryItemBranchProcessor</td>
<td>Use this operation to add or modify item branch records.</td>
</tr>
</tbody>
</table>

**7.2.2.4 Setting Up BSSVs**

Complete the following steps to set up BSSVs:

1. Enter P951000 in the Fast Path field.
2. Click Add.

   The Work with Business Service Property page appears, see Figure 7–7.
3. Enter the values in the Key and Group fields as listed in Table 7–16

Table 7–16  Values for Key and Group fields

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>J4100003_ITEM_MBF_VERSION</td>
<td>ZJDE0001</td>
</tr>
<tr>
<td>J4100003_ITEM_STOCKING_TYPE_CODE</td>
<td>S</td>
</tr>
<tr>
<td>J3000010_ECO_MBF_VERSION</td>
<td>ZJDE0001</td>
</tr>
</tbody>
</table>

4. Enter values in the Value and Description fields according to your requirements.

7.2.2.5 Configuration Parameters for Creating ECO Flow

The configuration parameters include:

- Enter the UDC value of the implemented state in JD Edwards EnterpriseOne into the ECO_Implemented status configuration property.
  
  This status is updated in the ECO, which is created for an ECO that does not hold any BOM information. Such ECOs carry only the items, which are created and updated from Agile PLM.

- Set the Default_ECO_Branch_Plant in the AIA configuration property to specify the branch plant under which the ECO should be created for facilitating the modification of nonstock items from Agile PLM.

- Ideally, nonstock items are not created in Agile PLM. However, because nonstock items flow in from JD Edwards EnterpriseOne to Agile PLM as part of initial loads, any changes made to these items are brought back as ECOs from Agile PLM, and they hold no branch plant information. Because the integration demands that ECOs be created with branch plant information, the system uses the default branch plant in this scenario.

7.2.3 Setting Up OWSM

You must set up the OWSM client agent to store the security credentials that are used by the JD Edwards EnterpriseOne web services.

This section discusses how to create credential maps and credential keys.
7.2.3.1 Creating Credential Maps and Credential Keys

WLS stores the user name and password for web service (BSSV) calls here.

1. In FMW Control, navigate to WebLogic Domain >soa_domains and select Security > Credentials from the available options. See Figure 7–8.

Figure 7–8 Selecting security credentials

2. If the oracle.wsm.security map does not exist, click Create Map.

3. Enter the map name as oracle.wsm.security. You must not change this name. See Figure 7–9.

Figure 7–9 Create map

4. Highlight the new map.

5. Click Create Key.

6. Enter the key as JDEE1_BSSV.

7. Enter the E1 user name and password
   You can also use the special E1 user name that includes environment and role as DN=JDE, ENV=DV900, ROLE=*ALL. See Figure 7–10.
7.2.4 Setting Up AIA

In the AIA config file, set a module-level property, DEFAULT_BOM_TYPE, which enables users to specify the BOM type that should be passed to JD Edwards EnterpriseOne. This information does not flow in from Agile PLM.

If you do not set this property for the forward flow from Agile PLM to JD Edwards EnterpriseOne, then JD Edwards EnterpriseOne will select the default BOM type from the Data Dictionary default value for the item.

Note: This property is maintained at the module level because the BOM initial load uses the same property.

7.3 Setting Up Batch Processing Information

To use the Initial Load and Update flows from JD Edwards EnterpriseOne, you must first set up batch information. This section discusses how to:

- Set up FTP adapter information
- Set up batch consumer properties in the Enterprise Manager (EM) console
- Set up resequencer for batch processing
- Adjust timeout settings

7.3.1 Setting Up FTP Adapter Information

Setting up FTP adapter information includes configuring the JNDI name for the FTP adapter.

To set up FTP Adapter information, complete these steps:

1. Access the Weblogic console.
   Navigate to Deployments and look for FtpAdapter.
2. Click the FtpAdapter link.
3. Navigate to Configuration tab > Outbound Connection Pools and click New.
4. Select the javax.resource.cci.ConnectionFactory group.
5. Click Next.

6. Enter the JNDI name as eis/Ftp/JDEE1FtpAdapter.

7. Click Finish.

8. Make sure the JNDI name was created on the Outbound Connection Pools tab.

9. Click the JNDI name that you created.

The Outbound Connection Properties page appears.

10. Edit the ftpAbsolutePathBegin, such as /slot/ems4309/appmgr, and press Enter.

11. Click Next to edit host, password, port, serverType, and username.

12. After you set the properties, click Save.

13. To update the deployment plan, navigate to the deployments page and select the FtpAdapter check box.

14. Click Update.

15. Click Next.

16. Click Finish.

17. Restart the soa_server to effect the FtpAdapter changes.

7.3.2 Setting Up Batch Consumer Properties

Complete the following steps to set up batch consumer properties:

1. For the initial load processes and other update processes to work correctly, set up properties from the Oracle Enterprise Manager console for these batch consumer services:

   - ItemInitialLoadExtractJDEE1Consumer
   - BillOfMaterialsInitialLoadExtractJDEE1Consumer

   To set these properties, access the Oracle Enterprise Manager console, locate the services, and select the Properties tab. The Properties tab becomes available when you click <Consumer>_ep.

   The PhysicalDirectory property for the ItemInitialLoadExtractJDEE1FtpConsumer_ep should match the path specified in the R4101D3 processing option relative to the FTP server default directory. If the files are written to D:\JDEdwards\E900\DDP\PrintQueue \ and the FTP server default directory is D:\JDEdwards, then the PhysicalDirectory property should be \E900\DDP\PrintQueue. The PhysicalDirectory property for the BillOfMaterialsInitialLoadExtractJDEE1FtpConsumer_ep should match the path specified in the processing options for R3002D relative to the FTP server default directory.

   The PhysicalDirectory property for the ItemInitialLoadExtractJDEE1FileConsumer_ep and BillOfMaterialsInitialLoadExtractJDEE1FileConsumer_ep should match the location of <ORACLE_HOME>/JDEE1In/. These services are delivered with an example path that should be updated to reflect the implementation. After the extract programs execute in JD Edwards EnterpriseOne, move the files to the JDEE1 In folder arranged in the order for these file consumer services to find and process the files.
All consumer services discussed in this section have the following two properties, which you can configure as listed in Table 7–17:

### Table 7–17 Consumer services properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchSize</td>
<td>This property determines how many records to include in each batch. The default value for this property is 50. If you use a value larger than 50, the consumer services will not function properly because Agile Integration Services (AIS) would consume more time and cause a middleware timeout. We recommend that you do not change this value.</td>
</tr>
</tbody>
</table>
| PollingFrequency        | This property specifies the time interval, in seconds, when the file consumers check the PhysicalDirectory property for new files to process. The default value for this property is 30.  
  Note: We recommend that you do not change the pollingFrequency or batchSize properties for the FTP adapter. |

2. For ECO, item list, item balance processes, and other update processes to work correctly, set up properties from the Oracle Enterprise Manager console for these batch consumer:

- EngineeringChangeOrderListExtractJDEE1Consumer
- ItemListExtractJDEE1Consumer
- ItemBalanceListExtractJDEE1Consumer

To set up these properties, access the Oracle Enterprise Manager console, locate the services, and select the Properties tab, which becomes available when you click `<Consumer>_ep`. The PhysicalDirectory property for the EngineeringChangeOrderListExtractJDEE1FTPConsumer_ep should match the path specified in the R3013D processing option. The physicalDirectory property for the ItemListExtractJDEE1FTPConsumer_ep should match the path specified in the processing options for R4102D. The physicalDirectory property for the ItemBalanceListExtractJDEE1FTPConsumer_ep should match the path specified in the processing options for R41021D. These paths should be relative to the FTP setup path on the EnterpriseOne Enterprise Server.

The PhysicalDirectory property for the EngineeringChangeOrderListExtractJDEE1FileConsumer_ep, ItemListExtractJDEE1FileConsumer_ep, and ItemBalanceListExtractJDEE1FileConsumer_ep should match the location of `<ORACLE_HOME>/JDEE1In/`. These services are delivered with an example path, which is /slot/ems2593/oracle/Middleware/Oracle_SOA1/JDEE1In. Update this path to reflect your implementation. After the extract programs execute in JD Edwards EnterpriseOne, move the files to the JDEE1 In folder so that these file consumer services can find and process the files.

All of these consumer services have the following two properties, which you can configure as listed in Table 7–18:
7.3.3 Setting Up Resequencer for Batch Processing

1. Log in to the Oracle Enterprise Manager console.

2. Select Soa Infrastructure and navigate to Administration > System MBean Browser. See Figure 7–11

3. Navigate to oracle.as.soainfra.config > Server:soa_server1 > MediatorConfig > mediator

4. On the right side, set the value of the attribute ResequencerWorkerThreadCount to 6. See Figure 7–12

For more information, see Enterprise Service Bus Quick Start Guide, “Creating, Configuring, and Managing an Oracle Enterprise Service Bus.”

### Table 7–18 Consumer service properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchSize</td>
<td>This property determines how many records to include in each batch. The default value for this property is 40.</td>
</tr>
</tbody>
</table>
| PollingFrequency | This property specifies the time interval, in seconds, when the file consumers check the PhysicalDirectory property for new files to process. The default value for this property is 10.  
  **Note:** We recommend that you do not change the pollingFrequency or batchSize properties for the FTP adapter. |
7.3.4 Adjusting Timeout Settings

You must set up the transaction timeout values using the Weblogic console and SyncMaxWaitTime using the Oracle Enterprise Manager console.

7.3.4.1 Setting SyncMaxWaitTime Value

1. Log in to the Oracle Enterprise Manager console.
2. Select Soa Infrastructure > navigate to Administration > System MBean Browser.
3. Navigate to oracle.as.soainfra.config > Server:soa_server1 > BPELConfig > bpel
4. On the right side, set the value of the SyncMaxWaitTime attribute to 360.
   
   By default, the value is 45 and you should increase it to at least 360 to accommodate batch processing.

7.3.4.2 Setting Transaction Timeout Values

1. Log in to the Weblogic console.
2. Navigate to Services > JTA.
3. On the configuration page set the Timeout Seconds to 420.
4. Save the setting and restart the server.

7.4 SQL Script for Loading Cross-Reference Records

There is no utility in 11g to load cross-reference (XRef) records similar to 10g. However, this can be accomplished with an SQL script.
The following is an example of a SQL script that loads cross-reference records for XRef table, ITEM_ITEMID with columns AGILE_01 and JDEE1_01. You can modify this SQL script to suit your specific requirements.

DECLARE
XrefTableName_vc2 VARCHAR2 (2000);
XrefColumnName1_vc2 VARCHAR2 (2000);
XrefColumnName2_vc2 VARCHAR2 (2000);
XrefColumnName3_vc2 VARCHAR2 (2000);
Value_vc2 VARCHAR2 (2000);
RowNumber_vc2 VARCHAR2 (48);
IsDeleted_vc2 VARCHAR2 (1);
LastModified_ts TIMESTAMP (6);
BEGIN
/* The following values need to be set just once per load*/
XrefTableName_vc2 := 'oramds:/apps/AIAMetaData/xref/ITEM_ITEMID.xref';
IsDeleted_vc2 := 'N';
XrefColumnName1_vc2 := 'COMMON';
XrefColumnName2_vc2 := 'AGILE_01';
XrefColumnName3_vc2 := 'JDEE1_01';
/* You'll just need to change the Value_vc2 variable for your specific edge app values */
/* Row Number is unique to a group of three inserts and ties them together. Its a system generated guid in all cases */
/**************************** Begin set of 3 inserts ****************************/
RowNumber_vc2 := SYS_GUID;
/**************************** Begin set of 3 inserts ****************************/
Value_vc2 := SYS_GUID;
/* Common row should have a guid for its value */
Value_vc2 := SYS_GUID;
LastModified_ts := SYSTIMESTAMP;
Insert into XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_DELETED, LAST_MODIFIED) values
(XrefTableName_vc2, XrefColumnName1_vc2, RowNumber_vc2, Value_vc2,
IsDeleted_vc2, LastModified_ts);
/* Value specific to each application, for Agile value generation please check XREF_ Instructions.doc */
Value_vc2 := 'AgileValue1';
LastModified_ts := SYSTIMESTAMP;
Insert into XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_DELETED, LAST_MODIFIED) values
(XrefTableName_vc2, XrefColumnName2_vc2, RowNumber_vc2, Value_vc2, IsDeleted_vc2, LastModified_ts);
/* Value specific to each application, for JDEE1 value generation please check XREF_ Instructions.doc */

7.5 Identifying Cross-References

Cross-references map and connect records within the application network, and they enable these applications to communicate in the same language. The integration server stores the relationship in a persistent way so that others can refer to it.

The three virtual tables in the AIA XRef schema’s XREF_DATA table that maintain this cross-reference information are as follows:

- CHANGE_CHANGEID: Includes change order information.
- ITEM_ITEMID: Includes item information with branch/plant.
- JDEE1_ITEMID: Includes item information without branch/plant.

Table 7–19 lists the cross-references:

<table>
<thead>
<tr>
<th>Name</th>
<th>Purpose</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEM_ITEMID</td>
<td>Query/Create</td>
<td>JDEE1_01, COMMON, AGILE_01</td>
</tr>
<tr>
<td>CHANGE_CHANGEID</td>
<td>Query/Create</td>
<td>JDEE1_01, COMMON, AGILE_01</td>
</tr>
<tr>
<td>JDEE1_ITEMID</td>
<td>Query/Create</td>
<td>JDEE1_01, COMMON, AGILE_01</td>
</tr>
</tbody>
</table>

7.6 Populating Cross-References

If item and BOM initial loads are not used to synchronize item and BOM data between Agile and JDE, then you must manually populate the ITEM_ITEMID and JDEE1_ITEMID cross-reference (XRef) tables after installing and configuring the PIP.

You must manually add the cross-reference data into the ITEM_ITEMID and JDEE1_ITEMID XRef tables by using SQL insert statements. These tables hold item information for common value in edge applications.

For each JD Edwards EnterpriseOne item processed through initial loads, the following rows need to be created in the XRef data table:

- One row for Agile: AGILE_01
- One row for JD Edwards EnterpriseOne: JDEE1_01
- One row for Common: COMMON

The physical XREF table (XREF_DATA) can be located in the database configured for FMW/SOA and contains the following columns:
Table 7–20  XRef Table Format

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XREF_TABLE_NAME</td>
<td>This column stores the XREF data types. For example:</td>
</tr>
<tr>
<td></td>
<td>■ oramds:/apps/AIAMetaData/xref/ITEM_ITEMID.xref</td>
</tr>
<tr>
<td></td>
<td>■ oramds:/apps/AIAMetaData/xref/JDEE1_ITEMID.xref</td>
</tr>
<tr>
<td>XREF_COLUMN_NAME</td>
<td>This column stores the values indicating the edge applications and COMMON value:</td>
</tr>
<tr>
<td></td>
<td>■ AGILE_01</td>
</tr>
<tr>
<td></td>
<td>■ JDEE1_01</td>
</tr>
<tr>
<td></td>
<td>■ COMMON</td>
</tr>
<tr>
<td>ROW_NUMBER</td>
<td>This column stores the GUID.</td>
</tr>
<tr>
<td>VALUE</td>
<td>This column stores the actual data that is cross-referenced.</td>
</tr>
<tr>
<td>IS_DELETED</td>
<td>This column indicates whether the column is deleted or not. For example, 'N'.</td>
</tr>
<tr>
<td>LAST_MODIFIED</td>
<td>This column stores last modified time.</td>
</tr>
</tbody>
</table>

Table 7–21  ITEM_ITEMID XRef Table Value Format

<table>
<thead>
<tr>
<th>XRef_Column_Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMON</td>
<td>GUID</td>
</tr>
<tr>
<td>AGILE_01</td>
<td>Agile ItemID::Site</td>
</tr>
<tr>
<td></td>
<td>For example, 60031::M30</td>
</tr>
<tr>
<td>EBIZ_01</td>
<td>JDE ItemID::Branch/ Plant</td>
</tr>
<tr>
<td></td>
<td>For example, 60031::M30</td>
</tr>
</tbody>
</table>

Sample SQL Insert Statements for ITEM_ITEMID Table

For JDEE1_01

INSERT INTO XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_DELETED, LAST_MODIFIED) VALUES ('oramds:/apps/AIAMetaData/xref/ITEM_ITEMID.xref', 'JDEE1_01', 'ROWNUM_ITEM_1', 'BP', 'N', SYSTIMESTAMP)

Note: The value for branch/plant in the value field varies depending on the JDE item type and Agile multi-site setup. For example, branch/plant that has disabled multi-site and non-stock item will have a value, "NOT_FOUND".

For AGILE_01

INSERT INTO XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_DELETED, LAST_MODIFIED) VALUES ('oramds:/apps/AIAMetaData/xref/ITEM_ITEMID.xref', 'AGILE_01', 'ROWNUM_ITEM_1', '88-JKO29::BP', 'N', SYSTIMESTAMP)
For COMMON

```
INSERT INTO XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_DELETED, LAST_MODIFIED) VALUES
('oramds:/apps/AIAMetaData/xref/ITEM_ITEMID.xref', 'COMMON', 'ROWNUM_ITEM_1', '1001001', 'N', SYSTIMESTAMP)
```

**Note:** The value for branch/plant in the value field varies depending on the JDE item type and Agile multi-site setup. For example, branch/plant that has disabled multi-site and non-stock item will have a value, "NOT_FOUND".

For the COMMON row, the value field will store a unique ID, for example, GUID. For each set, which includes JDEE1_01, AGILE_01, and COMMON, the row numbers must be identical to link these records to each other.

**Table 7–22 Sample Data for XRef ITEM_ITEMID Table**

<table>
<thead>
<tr>
<th>XRef_ Table</th>
<th>XRef_Column</th>
<th>Row_Number</th>
<th>Value</th>
<th>IS_Deleted</th>
<th>Last_Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>oramds:/apps/AIAMetaData/xref/ITEMITEMID.xref</td>
<td>JDEE1_01</td>
<td>B396F6 CE586 040F00 A5B09 7CBF</td>
<td>60031:: M30 08-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oramds:/apps/AIAMetaData/xref/ITEMITEMID.xref</td>
<td>AGILE_01</td>
<td>B396F6 CE586 040F00 A5B09 7CBF</td>
<td>60031:: M30 08-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oramds:/apps/AIAMetaData/xref/ITEMITEMID.xref</td>
<td>COMMON</td>
<td>B396F6 CE586 040F00 A5B09 7CBF</td>
<td>B396F6 6CE58 6211E6 E040F 00A5B 097CB F 08-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 7–23 JDEE1_ITEMID XRef Table Value Format**

<table>
<thead>
<tr>
<th>XRef_Column_Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMON</td>
<td>GUID</td>
</tr>
<tr>
<td>AGILE_01</td>
<td>Agile ItemID:Site</td>
</tr>
<tr>
<td></td>
<td>For example, 60031</td>
</tr>
<tr>
<td>JDEE1_01</td>
<td>JDE ItemID</td>
</tr>
<tr>
<td></td>
<td>For example, 60031</td>
</tr>
</tbody>
</table>
Sample SQL Insert statements for JDEE1_ITEMID Table

For JDEE1_01

```sql
INSERT INTO XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_DELETED, LAST_MODIFIED) VALUES
('oramds:/apps/AIAMetaData/xref/JDEE1_ITEMID.xref', 'JDEE1_01', 'ROWNUM_ITEM_1', '204', 'N', SYSTIMESTAMP)
```

For AGILE_01

```sql
INSERT INTO XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_DELETED, LAST_MODIFIED) VALUES
('oramds:/apps/AIAMetaData/xref/JDEE1_ITEMID.xref', 'AGILE_01', 'ROWNUM_ITEM_1', '88-JKO29', 'N', SYSTIMESTAMP)
```

For COMMON

```sql
INSERT INTO XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_DELETED, LAST_MODIFIED) VALUES
('oramds:/apps/AIAMetaData/xref/JDEE1_ITEMID.xref', 'COMMON', 'ROWNUM_ITEM_1', '1001001', 'N', SYSTIMESTAMP)
```

**Note:** For the COMMON row, the value field will store a unique ID, for example, GUID.

For more information about how to populate cross-references, see Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite, "Working with Cross References".

### Table 7–24 Sample Data for XRef JDEE1_ITEMID Table

<table>
<thead>
<tr>
<th>XRef Table</th>
<th>XRef Column</th>
<th>Row Number</th>
<th>Value</th>
<th>IS Deleted</th>
<th>Last Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>oramds:/apps/AIAMetaData/xref/JDEE1_ITEMID.xref</td>
<td>JDEE1_01</td>
<td>9C2518</td>
<td>60031</td>
<td>N</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>503F82</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11E0B</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FC94F</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4D513</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>04781</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oramds:/apps/AIAMetaData/xref/JDEE1_ITEMID.xref</td>
<td>AGILE_01</td>
<td>9C2518</td>
<td>60031</td>
<td>N</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>503F82</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11E0B</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FC94F</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4D513</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>04781</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oramds:/apps/AIAMetaData/xref/JDEE1_ITEMID.xref</td>
<td>COMMON</td>
<td>9C2518</td>
<td>2d3535</td>
<td>N</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>503F82</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11E0B</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FC94F</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4D513</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>04781</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>31</td>
<td>09-DEC-11 08.14.34.028279000 AM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.7 Describing DVMs

DVMs are a standard feature of the Oracle SOA Suite. They enable you to equate lookup codes and other static values across applications, for example, FOOT and FT or US and USA.

DVMs are static in nature, though administrators can add additional maps as needed. Transactional business processes never update DVMs; they only read from them. DVMs are stored in XML files and cached in memory at runtime.

DVM types are seeded for the Oracle Design to Release: Agile Product Lifecycle Management - JD Edwards EnterpriseOne flows and administrators can extend the list of mapped values by adding more maps. The DVM data should be synchronized with what the participating applications use. This synchronization should occur before any initial loads are run or any incremental transactional flows are initiated.

During installation, the DVMs used for the integration are imported with default data mappings. The values mapped by these DVMs must be changed as needed. Many DVMs are seeded and do not need to be changed. Because most of the Agile PLM attributes being mapped are list values, the Agile PLM data is not seeded and should be changed accordingly.

Table 7–25 lists DVMs used for this integration and their delivered values:

<table>
<thead>
<tr>
<th>DVM</th>
<th>Delivered Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECO_CLASSIFICATION_CODE.xml</td>
<td>ECO_OBJ/ClassificationCode</td>
</tr>
</tbody>
</table>

**Note:** You can modify the values according to your requirements. You can add rows of value mappings, but you cannot change the DVM name, column name, or the number of columns.
### Table 7–25 (Cont.) Domain Value Mapping

<table>
<thead>
<tr>
<th>DVM</th>
<th>Delivered Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECO_ECOSPECIFICATIONGROUPCATEGORYCODE7.xml</td>
<td>ECO_EngineeringChangeOrderSpecificationGroup/Specification</td>
</tr>
<tr>
<td>ECO_ECOSPECIFICATIONGROUPCATEGORYCODE8.xml</td>
<td>ECO_EngineeringChangeOrderSpecificationGroup/Specification</td>
</tr>
<tr>
<td>ECO_PRIORITY_CODE.xml</td>
<td>ECO_PriorityCode</td>
</tr>
<tr>
<td>ECO_REASON_CODE.xml</td>
<td>ECO_ReasonCode</td>
</tr>
<tr>
<td>ECO_STATUS_CODE.xml</td>
<td>ECO_Status/Code</td>
</tr>
<tr>
<td>ECO_TYPECODE.xml</td>
<td>ECO_TypeCode</td>
</tr>
<tr>
<td>ITEM_DUAL_UOM_TRACKINGIndicator.xml</td>
<td>ITEM_DualUOMTrackingIndicator</td>
</tr>
<tr>
<td>ITEM_INDICATOR.xml</td>
<td>ITEM_ItemPurchasingCharacteristics/TaxableIndicator or ITEM_ItemManufacturingCharacteristics/StructureAllowedIndicator</td>
</tr>
<tr>
<td>ITEM_INVENTORY_PLANNINGCODE.xml</td>
<td>ITEM_ItemPlanningCharacteristics/InventoryPlanningCode</td>
</tr>
<tr>
<td>ITEM_INVOICINGENABLEDINDICATOR.xml</td>
<td>ITEM_ItemOrderManagementCharacteristics/InvoicingEnabledIndicator</td>
</tr>
<tr>
<td>ITEM_ITEMSPECIFICATIONGROUPCYCLECOUNTCATEGORY.xml</td>
<td>ITEM_ItemSpecificationGroup/SpecificationGroup/Specification</td>
</tr>
<tr>
<td>ITEM_ITEMSPECIFICATIONGROUPPURCHASINGREPORTCODE1.xml</td>
<td>ITEM_ItemSpecificationGroup/SpecificationGroup/Specification</td>
</tr>
<tr>
<td>ITEM_ITEMSPECIFICATIONGROUPPURCHASINGREPORTCODE15.xml</td>
<td>ITEM_ItemSpecificationGroup/SpecificationGroup/Specification</td>
</tr>
<tr>
<td>DVM</td>
<td>Delivered Value</td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
<tr>
<td>ITEM_</td>
<td>ITEM_</td>
</tr>
</tbody>
</table>
Table 7–25 (Cont.) Domain Value Mapping

| DVM | Delivered Value |
|-----|-----------------
| ITEM_ITEMSPECIFICATIONGROUP_SHIPPINGCOMMODITYCLASS.xm | ITEM_Obj/ItemSpecificationGroup/SpecificationGroup/Specifi | |
| l | cation |
| ITEM_ITEMSPECIFICATIONGROUP_SHIPPINGCONDITIONSCODE.xml | ITEM_Obj/ItemSpecificationGroup/SpecificationGroup/Specifi | |
| | cation |
| ITEM_MAKEORBUY_CODE.xml | ITEM_Obj/ItemPlanningCharacteristics/MakeOrBuyCode |
| ITEM_PURCHASINGALLOWEDINDICATOR.xml | ITEM_Obj/ItemPurchasingCharacteristics/PurchasingAllowedIn | |
| | dicator |
| ITEM_REPLENISHMENT_SOURCE_CODE.xml | ITEM_Obj/ItemPlanningCharacteristics/ReplenishmentSource | |
| | Code |
| ITEM_SERIALIZATION_EVENT_CODE.xml | ITEM_Obj/InventoryCharacteristics/SerializationEventCode |
| ITEM_STATUS_CODE | ITEM_Obj/Status/Code |
| ITEM_STOCKINGALLOWEDINDICATOR.xml | ITEM_Obj/InventoryCharacteristics/StockingAllowedIndica | |
| | tor |
| ITEM_TYPE.xml | ITEM_Obj/TypeCode |
| ITEM_UOM_CODE.xml | ITEM_Obj/BaseUOMCode ITEM_Obj/ShippingUOMCode ITEM_Obj/Sec | |
| | ondaryUOMCode ITEM_Obj/ItemPhysicalCharacteristics/VolumeMeasure |
| | ITEM_Obj/ItemPurchasingCharacteristics/IssueUOMCode | |
| | ITEM_Obj/ItemPhysicalCharacteristics/WeightMeasure |
| AGILE_SITE_TARGET_MAPPING | DEFAULT_MASTER_ORG in JD Edwards EnterpriseOne is used when the Multisite Enabled property is set to False and no branch/plant is associated with the item in Agile. |
| AGILE_TARGET_SITE_MAPPING | The JD Edwards EnterpriseOne branch/plant to Agile PLM sites is mapped. A one-to-one mapping exists between the EnterpriseOne branch/plant to Agile PLM site. |

7.8 Viewing EIMs

For more information about using XSL Mapping Analyzer (XMAN), see Oracle Fusion Middleware Infrastructure Components and Utilities User’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, Using the XSL Mapping Analyzer.

For more information about how services are mapped, see the My Oracle Support document: EBO Implementation Maps (EIMs) 1095494.1.
7.9 Setting Configuration Properties

This integration uses various configuration parameters that control the behavior of the flow. You use the Standard AIA XML configuration file and the AIAConfigurationProperties.xml file for capturing configuration parameters. AIA configuration file supports system-level configuration parameters, service-level parameters, and module configuration parameters. System-level parameters apply to all integrations running on the SOA suite. You can configure service-level parameters at the individual service level, such as ABCS.

Note: This section lists the configuration properties from the Agile PLM Module and the JD Edwards EnterpriseOne Module separately only for the purpose of identification. The actual AIAConfigurationProperties.xml file on the AIA Server is a merge of both.

For more information about updating SOA MDS with AIA metadata, see Oracle Fusion Middleware Developer’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, “How to Set Up AIA Workstation,” Updating SOA MDS with AIA MetaData.

Configuration Parameters

This integration uses the following configuration parameters:

- **PIP level configuration parameters:** PIP Configuration parameters are implemented using AIA module configuration entry.

  The module configuration entry has a name and can contain any number of configuration parameters. A naming convention of PIPS.PIPName is used for naming modules. The parameters inside the module are named using with cascaded naming convention where individual words are separated with dots.

  For example, agile.replicate.item

- **Service level configuration parameters:** While most configuration requirements are satisfied by the PIP Level configuration parameters, sometimes the behavior of a flow needs to be controlled at the service level.

  These parameters can be captured using AIA service configuration parameters. Service configuration entry is identified by the service name such as CreateItemAgileReqABCSImpl. The parameter names themselves are named using cascaded naming convention as explained earlier.

For more information about requirements for working with AIAConfigurationProperties.xml, see Oracle Fusion Middleware Developer’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, “Building AIA Integration Flows,” How to Set Up AIA Workstation.

For more information about updating SOA MDS with AIA metadata, see Oracle Fusion Middleware Developer’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, “How to Set Up AIA Workstation,” Updating MDS.

Table 7–26 lists the Agile PLM configuration properties, value/setting default values, and descriptions:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>moduleName</td>
<td>Agile</td>
<td></td>
</tr>
</tbody>
</table>
Table 7–27 lists the setting for the CreateQueueService service property:

**MULTISITE_ENABLED** TRUE  
When set to True, the sites specified on Sites tab of items are used to determine the Orgs in JD Edwards EnterpriseOne to which they are mapped.  
When set to False, Page2 Multilist01 attribute is used to determine the Orgs in JD Edwards EnterpriseOne to which the Item is extended.

**Item.UnitCostAttribute** Site.Numeric01  
Determines the attribute to which the unit cost from JD Edwards EnterpriseOne is updated in Agile PLM.

**Item.AvailableQuantityAttribute** Site.Numeric02  
Determines the attribute to which the available quantity from JD Edwards EnterpriseOne is updated in Agile PLM.

**Item.OnHandQuantityAttribute** Site.Numeric03  
Determines the attribute to which the on-hand quantity from JD Edwards EnterpriseOne is updated in Agile PLM.

**Item.ReservedQuantityAttribute** Site.Numeric04  
Determines the attribute to which the reserved quantity from JD Edwards EnterpriseOne is updated in Agile PLM.

**REPLICATE_BOM_ENABLED** FALSE  
Used for sample replicate BOM customization.

**COMMON_BOM_ENABLED** FALSE  
Used for sample common BOM customization.

---

**Note:** Multisite Enabled property is governed by Distributed Processing aspects covered in Release of Change Order in Agile PLM. When it is set to TRUE (default), the Item.UnitCostAttribute, Item.AvailableQuantityAttribute, Item.OnHandQuantityAttribute, and Item.ReservedQuantityAttribute are set to Site tab flex attributes.

You can use Numeric, Text, or Money flex fields of Site tab for these settings, and it is denoted by the first element, Site. For example, Site.Numeric01

When set to FALSE; all these attributes are set to Page2 or Page3 Flex Fields. Therefore, you must change the settings to PageTwo.Numeric01 or PageThree.Numeric01, accordingly. Derive the names of the attributes from the ItemABM Schema, which you can find in Agile PLM Interfaces.

Table 7–27 lists the setting for the CreateQueueService service property:
### Table 7–27 Settings CreateQueueService service property

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRACE.LOG.ENABLED</td>
<td>False</td>
<td>Use tracelog for the flow.</td>
</tr>
</tbody>
</table>

### Table 7–28 Settings for QueueProcessorServiceImpl service property

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRACE.LOG.ENABLED</td>
<td>False</td>
<td>Use tracelog for the flow.</td>
</tr>
</tbody>
</table>

### Table 7–29 Settings for UpdateEngineeringChangeOrderListAgileProvABCSImpl service properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Default Value/Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCSExtension. PreProcessABM</td>
<td>false</td>
<td>User exit for the preprocess Application Business Message (ABM) should be called or not</td>
</tr>
<tr>
<td>ABCSExtension. PreProcessEBM</td>
<td>false</td>
<td>User exit for the preprocess Enterprise Business Message (EBM) should be called or not</td>
</tr>
<tr>
<td>ABCSExtension. PostProcessEBM</td>
<td>false</td>
<td>User exit for the post-process EBM should be called or not</td>
</tr>
<tr>
<td>ABCSExtension. PostProcessABM</td>
<td>false</td>
<td>User exit for the post-process ABM should be called or not</td>
</tr>
<tr>
<td>TRACE.LOG.ENABLED</td>
<td>true</td>
<td>Use tracelog for the flow.</td>
</tr>
<tr>
<td>Default.SystemID</td>
<td>AGILE_01</td>
<td>System ID of Agile PLM application instance</td>
</tr>
<tr>
<td>Routing.EngineeringChangeOrderResponseEBS.UpdateEngineeringChangeOrderListResponse.RouteToCAVS</td>
<td>http://${hostname}:${port}/AIAValidationSystemServiceAsyncResponsesimulator</td>
<td>CAVS SOAP URL. When the RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation relates to the ECO response EBS</td>
</tr>
<tr>
<td>Routing.EngineeringChangeOrderResponseEBS.UpdateEngineeringChangeOrderListResponse.CAVS. EndpointURI</td>
<td>PRODUCTION</td>
<td>Environment code such as 'PRODUCTION'/'CAVS'. Identifies the installation environment</td>
</tr>
</tbody>
</table>
### Table 7–29 (Cont.) Settings for UpdateEngineeringChangeOrder property

<table>
<thead>
<tr>
<th>Property</th>
<th>Default Value/Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing.ChangeABSService.RouteToCAVS</td>
<td>false</td>
<td>If set to true, route to CAVS; otherwise, route to the Agile PLM application. This invocation relates to the change ABS service.</td>
</tr>
<tr>
<td>Routing.ChangeABSService.AGILE_01.EndpointURI</td>
<td>http://${agile.hostname}:${agile.port}/AgilePLM/integration/services/ChangeABS</td>
<td>ChangeABS URL: When the RouteToCAVS property is set to false, use the URL mentioned to connect to the Agile PLM ChangeABS Service for the ChangeABSService partnerlink.</td>
</tr>
<tr>
<td>Routing.ChangeABSService.CAVS.EndpointURI</td>
<td>http://${agile.hostname}:${agile.port}/AIVSValidationSystemServlet/syncresponsesimulator</td>
<td>CAVS SOAP URL: When the RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation relates to the change ABS service.</td>
</tr>
<tr>
<td>Routing.MergeABSService.RouteToCAVS</td>
<td>false</td>
<td>If set to true, route to CAVS; otherwise, route to the Agile PLM application. This invocation relates to the change merge ABS service.</td>
</tr>
<tr>
<td>Routing.MergeABSService.CAVS.EndpointURI</td>
<td>http://${agile.hostname}:${agile.port}/AIVSValidationSystemServlet/syncresponsesimulator</td>
<td>CAVS Simple Object Access Protocol (SOAP) URL: When the RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation relates to the change merge ABS service.</td>
</tr>
<tr>
<td>Routing.MergeABSService.AGILE_01.EndpointURI</td>
<td>http://${agile.hostname}:${agile.port}/AgilePLM/integration/services/MergeABSService_Port</td>
<td>MergeABS URL: When the RouteToCAVS property is set to false, use the URL mentioned to connect to the Agile PLM MergeABS Service for the MergeABSService partnerlink.</td>
</tr>
<tr>
<td>Routing.ChangeStatusService.RouteToCAVS</td>
<td>false</td>
<td>If set to true, it is routed to CAVS; otherwise, routed to the Agile PLM application. This invocation relates to the change status ABS service.</td>
</tr>
<tr>
<td>Routing.ChangeStatusService.CAVS.EndpointURI</td>
<td>http://${agile.hostname}:${agile.port}/AIVSValidationSystemServlet/syncresponsesimulator</td>
<td>CAVS URL: When the RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation relates to the change status ABS service.</td>
</tr>
<tr>
<td>Routing.ChangeStatusService.AGILE_01.EndpointURI</td>
<td>http://${agile.hostname}:${agile.port}/AgilePLM/integration/services/ChangeStatusService</td>
<td>ChangeABS URL: When the RouteToCAVS property is set to False, use the URL mentioned to connect to the Agile PLM ChangeABS Service for the ChangeStatusService partnerlink.</td>
</tr>
<tr>
<td>Routing.ChangeStatusDBAdapter.RouteToCAVS</td>
<td>false</td>
<td>If set to true, it is routed to CAVS; otherwise, routed to the Agile PLM application. This invocation relates to the change status EDB Adapter.</td>
</tr>
<tr>
<td>Routing.ChangeStatusDBAdapter.CAVS.EndpointURI</td>
<td>http://${agile.hostname}:${agile.port}/AIVSValidationSystemServlet/syncresponsesimulator</td>
<td>CAVS URL: When the RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation relates to the change status EDB Adapter.</td>
</tr>
</tbody>
</table>
Table 7–30 lists the settings for the UpdateItemBalanceListAgileProvABCSImpl service properties:

Table 7–30 Settings for UpdateItemBalance property

<table>
<thead>
<tr>
<th>Property</th>
<th>Default Value/Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default.SystemID</td>
<td>AGILE_01</td>
<td>System ID of Agile PLM application instance</td>
</tr>
<tr>
<td>Routing.ItemABSService.RouteToCAVS</td>
<td>FALSE</td>
<td>If set to True, route to CAVS; otherwise, route to Agile PLM application. This invocation is for Item ABS</td>
</tr>
<tr>
<td>Routing.ItemABSService.CAVS.EndpointURI</td>
<td>http://${http.hostname}:${http.port}/AIAValidationSystemService/syncresponsesimulator</td>
<td>CAVS URL; When RouteToCAVS property is set to True, use the URL mentioned to connect to CAVS. This invocation is for item ABS</td>
</tr>
<tr>
<td>Routing.ItemBalanceResponseEBS.UpdateItemBalanceListResponse.RouteToCAVS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABCSExtension.PresProcessesABM</td>
<td>FALSE</td>
<td>Decides whether user exit for the preprocess ABM should be called or not</td>
</tr>
<tr>
<td>ABCSExtension.PresProcessesEBM</td>
<td>FALSE</td>
<td>Decides whether user exit for the preprocess EBM should be called or not</td>
</tr>
</tbody>
</table>
### Table 7–30 (Cont.) Settings for UpdateItemListAgile Property

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCSExtension.PostProcessEBM</td>
<td>FALSE</td>
<td>Decides whether user exit for the post-process EBM should be called or not</td>
</tr>
<tr>
<td>ABCSExtension.PostProcessABM</td>
<td>FALSE</td>
<td>Decides whether user exit for the post-process ABM should be called or not</td>
</tr>
<tr>
<td>TRACE.LOG.ENABLED</td>
<td>FALSE</td>
<td>Use tracelog for the flow</td>
</tr>
<tr>
<td>ROUTE_TO_CAVS</td>
<td>FALSE</td>
<td>Route to CAVS if set as True</td>
</tr>
<tr>
<td>DEFAULT_TARGET_ENDPOINT_URI</td>
<td>http://&lt;host&gt;:&lt;port&gt;/event/CoreAgile/ItemBalanceResponseEBS</td>
<td>ItemBalanceResponseEBS runtime target endpoint URI</td>
</tr>
</tbody>
</table>

### Table 7–31 Settings for UpdateItemListAgileProvABCSImpl service properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default.SystemID</td>
<td>AGILE_01</td>
<td>System ID of Agile PLM application instance</td>
</tr>
<tr>
<td>Routing.ItemResponseEBS.V2.UpdateItemListResponse.MessageProcessingInstruction.EnvironmentCode</td>
<td>Environment code such as 'PRODUCTION' / 'CAVS' and so on. Identifies the installation environment</td>
<td></td>
</tr>
<tr>
<td>Routing.ItemABSService.RouteToCAVS</td>
<td>FALSE</td>
<td>If set to True, route to CAVS; otherwise, route to Agile PLM application. This invocation is for Agile Item ABS service.</td>
</tr>
<tr>
<td>Routing.ItemABSService.AGILE_01.EndpointURI</td>
<td>http://${agile.host}:${agile.port}/${agiag.path}/integration/services/ItemABS</td>
<td>ItemABS URL. When RouteToCAVS property is set to False, use the URL mentioned to connect to the Agile ItemABS Service.</td>
</tr>
<tr>
<td>Routing.ItemABSService.CAVS.EndpointURI</td>
<td>http://${http.hostname}:${http.port}/AIAValidationSystemService/syncreponsesimulator</td>
<td>Agile Item ABS service SOAP URL. When RouteToCAVS property is set to True, use the URL mentioned to connect to CAVS. This invocation is for Agile Item ABS service.</td>
</tr>
<tr>
<td>Routing.ItemResponseEBS.V2.UpdateItemListResponse.RouteToCAVS</td>
<td>FALSE</td>
<td>If set to True, route to CAVS; otherwise, route to response item EBS. This invocation is for response item EBS</td>
</tr>
<tr>
<td>Routing.ItemResponseEBS.V2.UpdateItemListResponse.CAVS.EndpointURI</td>
<td>http://${http.hostname}:${http.port}/AIAValidationSystemService/asyncresponsesimulator</td>
<td>CAVS URL; When RouteToCAVS property is set to True, use the URL mentioned to connect to CAVS. This invocation is for Item response EBS</td>
</tr>
</tbody>
</table>
Table 7–32 lists the settings for the ProcessEngineeringChangeOrderAgileReqABCSImpl service properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCSExtension.PreProcessABM</td>
<td>FALSE</td>
<td>Decides whether user exit for the preprocess ABM should be called or not</td>
</tr>
<tr>
<td>ABCSExtension.PreProcessEBM</td>
<td>FALSE</td>
<td>Decides whether user exit for the preprocess EBM should be called or not</td>
</tr>
<tr>
<td>ABCSExtension.PostProcessABM</td>
<td>FALSE</td>
<td>Decides whether user exit for the post-process ABM should be called or not</td>
</tr>
<tr>
<td>ABCSExtension.PostProcessEBM</td>
<td>FALSE</td>
<td>Decides whether user exit for the post-process EBM should be called or not</td>
</tr>
<tr>
<td>TRACE.LOG.ENABLED</td>
<td>FALSE</td>
<td>Use trace log for the flow</td>
</tr>
<tr>
<td>ROUTE_TO_CAVS</td>
<td>FALSE</td>
<td>Route to CAVS if set as True</td>
</tr>
<tr>
<td>DEFAULT_TARGET_ENDPOINT_URI</td>
<td><a href="http://localhost:8888/event/CoreAgile/ItemResponseEBSV2">http://localhost:8888/event/CoreAgile/ItemResponseEBSV2</a></td>
<td>ItemResponseEBSV2 runtime target endpoint URI</td>
</tr>
</tbody>
</table>

Table 7–31 (Cont.) Settings for UpdateItemListAgile Property

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
</table>

Configuring Design to Release: Agile PLM - JDE E1
Table 7–33 lists the settings for the JD Edwards EnterpriseOne properties:

Table 7–33  Settings for JDEE1 Property

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGILE_NAMESPACE</td>
<td>AGILE_AXML_SCHEMA_NAMESPACE</td>
<td>This property is used in initial loads. The token AGILE_AXML_SCHEMA_NAMESPACE is replaced automatically with the Agile namespace corresponding to the versions of Agile PLM during PIP Installation.</td>
</tr>
<tr>
<td>DEFAULT_BOM_TYPE</td>
<td>M</td>
<td>Default BOM type</td>
</tr>
<tr>
<td>INITIALLOAD.DEF_AULTSOURCE</td>
<td>JDEE1_01</td>
<td>System ID of JD Edwards EnterpriseOne instance</td>
</tr>
<tr>
<td>INITIALLOAD.DEF_AULTTARGET</td>
<td>AGILE_01</td>
<td>System ID of Agile PLM application instance</td>
</tr>
</tbody>
</table>
Table 7–34 lists the settings for the CreateEngineeringChangeOrderListJDEE1ProvABCSImpl service properties:

**Table 7–34 Settings for CreateEngineeringChangeOrder property**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default.SystemID</td>
<td>JDEE1_01</td>
<td>System ID of JD Edwards EnterpriseOne instance</td>
</tr>
<tr>
<td>Routing.Engineering</td>
<td>False</td>
<td>If set to True, route to CAVS otherwise, route to JD Edwards EnterpriseOne. This invocation is for the EngineeringChangeOrderManager ABS service</td>
</tr>
<tr>
<td>ChangeOrderManager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>r.RouteToCAVS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routing.Engineering</td>
<td>False</td>
<td>If set to True, route to CAVS otherwise, route to ECO Response EBS. This invocation is for ECO response EBS</td>
</tr>
<tr>
<td>ChangeOrderResponse</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seEBS.RouteToCAVS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 7–35  (Cont.) Settings for CreateEngineeringChangeOrder property

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing.EngineeringChangeOrderManager.CAVS.EndpointURI</td>
<td>http://${http.hostname}:${http.port}//AIValidationSystemServlet/syncresponcesimulator</td>
<td>CAVS SOAP URL. When RouteToCAVS property is set to True, use the URL mentioned to connect to CAVS. This invocation is for the EngineeringChangeOrderManager ABS service.</td>
</tr>
<tr>
<td>Routing.EngineeringChangeOrderResponseEBS.CAVS.EndpointURI</td>
<td>http://${http.hostname}:${http.port}//AIValidationSystemServlet/asyncresponcesimulator</td>
<td>CAVS SOAP URL. When RouteToCAVS property is set to True, use the URL mentioned to connect to CAVS. This invocation is for ECO response EBS.</td>
</tr>
<tr>
<td>Routing.EngineeringChangeOrderManager.JDEE1_01.EndpointURI</td>
<td>${participatingapplications.jdee1.ws.url}/EngineeringChangeOrderManager</td>
<td>JD Edwards EnterpriseOne EngineeringChangeOrderManager ABS service URL. When RouteToCAVS property is set to False, use the URL mentioned to connect to JD Edwards EnterpriseOne. This invocation is for the EngineeringChangeOrderManager ABS service.</td>
</tr>
<tr>
<td>ABCSExtension.PreProcessABM</td>
<td>False</td>
<td>Decides whether user exit for the preprocess ABM should be called or not.</td>
</tr>
<tr>
<td>ABCSExtension.PostProcessABM</td>
<td>False</td>
<td>Decides whether user exit for the preprocess EBM should be called or not.</td>
</tr>
<tr>
<td>ABCSExtension.PreProcessEBM</td>
<td>False</td>
<td>Decides whether user exit for the post-process EBM should be called or not.</td>
</tr>
<tr>
<td>ABCSExtension.PostProcessEBM</td>
<td>False</td>
<td>Decides whether user exit for the post-process ABM should be called or not.</td>
</tr>
<tr>
<td>Trace.Log.Enabled</td>
<td>False</td>
<td>Use tracelog for the flow.</td>
</tr>
<tr>
<td>Routing.EngineeringChangeOrderResponseEBS.MessageProcessingInstruction.EnvironmentCode</td>
<td>PRODUCTION</td>
<td>Environment code such as 'PRODUCTION'/'CAVS' and so on. Identifies the installation environment.</td>
</tr>
<tr>
<td>ECO_Implemented_Status</td>
<td>E5</td>
<td></td>
</tr>
<tr>
<td>Default_ECO_Branch_Plant</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7–35 lists the settings for the UpdateEngineeringChangeOrderListJDEE1ReqABCSImp service properties:
<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default.SystemID</td>
<td>JDEE1_01</td>
<td>System ID of JD Edwards EnterpriseOne instance</td>
</tr>
<tr>
<td>ABCExtension.PreXformABMtoEBM</td>
<td>False</td>
<td>Decides whether user exit for the preprocess ABM should be called or not</td>
</tr>
<tr>
<td>ABCExtension.PostXformABMtoEBM</td>
<td>False</td>
<td>Decides whether user exit for the preprocess EBM should be called or not</td>
</tr>
<tr>
<td>ABCExtension.PreInvokeEBS</td>
<td>False</td>
<td>Decides whether user exit for the post-process EBM should be called or not</td>
</tr>
<tr>
<td>ABCExtension.PostInvokeEBS</td>
<td>False</td>
<td>Decides whether user exit for the post-process ABM should be called or not</td>
</tr>
<tr>
<td>Routing.TimeStampManager.RouteToCAVS</td>
<td>False</td>
<td>If set to true route to CAVS else route to JD Edwards EnterpriseOne. This invocation is for IntegrationTimeStampManager ABS service</td>
</tr>
<tr>
<td>Routing.TimeStampManager.CAVS.EndpointURI</td>
<td>http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponse simulator</td>
<td>CAVS SOAP URL When RouteToCAVS property is set to True, use the URL mentioned to connect to CAVS. This invocation is for IntegrationTimeStampManager ABS service</td>
</tr>
<tr>
<td>Routing.EngineeringChangeOrderEBS.UpdateEngineeringChangeOrderList.RouteToCAVS</td>
<td>False</td>
<td>If set to True, route to CAVS; otherwise, route to ECO EBS. This invocation is for ECO EBS</td>
</tr>
<tr>
<td>Routing.EngineeringChangeOrderEBS.UpdateEngineeringChangeOrderList.CAVS.EndpointURI</td>
<td>http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrespon sesimulator</td>
<td>CAVS SOAP URL When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation is for ECO EBS</td>
</tr>
<tr>
<td>Routing.EngineeringChangeOrderEBS.UpdateEngineeringChangeOrderList.MessageProcessingInstruction.EnvironmentCode</td>
<td>PRODUCTION</td>
<td>Environment code such as 'PRODUCTION' / 'CAVS' and so on. Identifies the installation environment</td>
</tr>
<tr>
<td>Routing.TimeStampManager.JDEE1_01.EndpointURI</td>
<td>$[participatingapplications.jdee1.ws.url]/IntegrationTimeStampManager</td>
<td>JD Edwards EnterpriseOne IntegrationTimeStampManager ABS service URL. When RouteToCAVS property is set to False, use the URL mentioned to connect to JD Edwards EnterpriseOne. This invocation is for IntegrationTimeStampManager ABS service</td>
</tr>
<tr>
<td>TRACE.LOG.ENABLED</td>
<td>False</td>
<td>Use tracelog for the flow</td>
</tr>
</tbody>
</table>
Table 7–36 lists the settings for the UpdateItemBalanceListJDEE1ReqABCSImpl service properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default.SystemID</td>
<td>JDEE1_01</td>
<td>System ID of JD Edwards EnterpriseOne instance</td>
</tr>
<tr>
<td>ABCSExtension.PreXformABMtoEBM</td>
<td>False</td>
<td>Decides whether user exit for the preprocess ABM should be called or not</td>
</tr>
<tr>
<td>ABCSExtension.PostXformABMtoEBM</td>
<td>False</td>
<td>Decides whether user exit for the preprocess EBM should be called or not</td>
</tr>
<tr>
<td>ABCSExtension.PreInvokeEBS</td>
<td>False</td>
<td>Decides whether user exit for the post-process EBM should be called or not</td>
</tr>
<tr>
<td>ABCSExtension.PostInvokeEBS</td>
<td>False</td>
<td>Decides whether user exit for the post-process ABM should be called or not</td>
</tr>
<tr>
<td>Routing.TimeStampManager.RouteToCAVS</td>
<td>False</td>
<td>If set to True, route to CAVS; otherwise, route to JD Edwards EnterpriseOne. This invocation is for IntegrationTimeStampManager ABS service</td>
</tr>
<tr>
<td>Routing.TimeStampManager.CAVS.EndpointURI</td>
<td>http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator</td>
<td>CAVS SOAP URL: when RouteToCAVS property is set to True, use the URL mentioned to connect to CAVS. This invocation is for IntegrationTimeStampManager ABS service</td>
</tr>
<tr>
<td>Routing.ItemBalanceEBS.UpdateItemBalanceList.RouteToCAVS</td>
<td>FALSE</td>
<td>If set to True, route to CAVS; otherwise, route to ItemBalance EBS. This invocation is for ItemBalance EBS</td>
</tr>
<tr>
<td>Routing.ItemBalanceEBS.UpdateItemBalanceList.CAVS.EndpointURI</td>
<td>http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncresponsesimulator</td>
<td>CAVS SOAP URL. When RouteToCAVS property is set to True, use the URL mentioned to connect to CAVS. This invocation is for ItemBalance EBS.</td>
</tr>
<tr>
<td>Routing.ItemBalanceEBS.UpdateItemBalanceList.MessageProcessingInstruction.EnviornmentCode</td>
<td>PRODUCTION</td>
<td>Environment code such as ‘PRODUCTION’/’CAVS’ and so on. Identifies the installation environment</td>
</tr>
<tr>
<td>Routing.TimeStampManager.JDEE1_01.EndpointURI</td>
<td><code>${participatingapplications.jdee1.ws.url}/IntegrationTimeStampManager</code></td>
<td>JD Edwards EnterpriseOne IntegrationTimeStampManager ABS service URL. When RouteToCAVS property is set to False, use the URL mentioned to connect to JD Edwards EnterpriseOne. This invocation is for IntegrationTimeStampManager ABS service</td>
</tr>
<tr>
<td>TRACE.LOG.ENABLED</td>
<td>false</td>
<td>Use trace log for the flow</td>
</tr>
</tbody>
</table>
Table 7–37 lists the settings for the UpdateItemListJDEE1ReqABCSImpl service properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default.SystemID</td>
<td>JDEE1_01</td>
<td>System ID of JD Edwards EnterpriseOne instance</td>
</tr>
<tr>
<td>ABCSExtension.PreXformABMtoEBM</td>
<td>False</td>
<td>Decides whether user exit for the preprocess ABM should be called or not</td>
</tr>
<tr>
<td>ABCSExtension.PostXformABMtoEBM</td>
<td>false</td>
<td>Decides whether user exit for the preprocess EBM should be called or not</td>
</tr>
<tr>
<td>ABCSExtension.PreInvokeEBS</td>
<td>False</td>
<td>Decides whether user exit for the post-process EBM should be called or not</td>
</tr>
<tr>
<td>ABCSExtension.PostInvokeEBS</td>
<td>False</td>
<td>Decides whether user exit for the post-process ABM should be called or not</td>
</tr>
<tr>
<td>Routing.TimeStampManager.RouteToCAVS</td>
<td>False</td>
<td>If set to True, route to CAVS; otherwise, route to JD Edwards EnterpriseOne. This invocation is for IntegrationTimeStampManager ABS service</td>
</tr>
<tr>
<td>Routing.TimeStampManager.CAVS.EndpointURI</td>
<td>http://$[http.hostname]:$[http.port]/AIAValidationSystemService/syncresponsesimulator</td>
<td>CAVS SOAP URL. When RouteToCAVS property is set to True, use the URL mentioned to connect to CAVS. This invocation is for IntegrationTimeStampManager ABS service</td>
</tr>
<tr>
<td>Routing.ItemEBSV2.UpdateItemList.RouteToCAVS</td>
<td>FALSE</td>
<td>If set to True, route to CAVS; otherwise, route to ItemBalance EBS. This invocation is for ItemBalance EBS</td>
</tr>
<tr>
<td>Routing.ItemEBSV2.UpdateItemList.CAVS.EndpointURI</td>
<td>http://$[http.hostname]:$[http.port]/AIAValidationSystemService/asyncresponsesimulator</td>
<td>CAVS SOAP URL. When RouteToCAVS property is set to True, use the URL mentioned to connect to CAVS. This invocation is for ItemBalance EBS</td>
</tr>
<tr>
<td>Routing.ItemEBSV2.UpdateItemList.MessageProcessingInstruction.EnvironmentCode</td>
<td>PRODUCTION</td>
<td>Environment code such as 'PRODUCTION' / 'CAVS' and so on. Identifies the installation environment</td>
</tr>
<tr>
<td>Routing.TimeStampManager.JDEE1_01.EndpointURI</td>
<td>$[participatingapplications.jdee1.ws.url]/IntegrationTimeStampManager</td>
<td>JD Edwards EnterpriseOne IntegrationTimeStampManager ABS service URL. When RouteToCAVS property is set to False, use the URL mentioned to connect to JD Edwards EnterpriseOne. This invocation is for IntegrationTimeStampManager ABS service</td>
</tr>
<tr>
<td>TRACE.LOG.ENABLED</td>
<td>false</td>
<td>Use tracelog for the flow</td>
</tr>
</tbody>
</table>
Table 7–38 lists the settings for the InitialLoadBillOfMaterialsListJDEE1toAgileImpl service properties:

**Table 7–38  Settings for InitialLoadBillOfMaterialsList property**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default.SystemID</td>
<td>JDEE1_01</td>
<td>System ID of JD Edwards EnterpriseOne instance</td>
</tr>
<tr>
<td>TRACE.LOG.ENABLED</td>
<td>False</td>
<td>Use tracelog for the flow</td>
</tr>
<tr>
<td>ABCExtension.PreXformABMtoAXML</td>
<td>False</td>
<td>Decides whether user exit for the preprocess ABM should be called or not</td>
</tr>
<tr>
<td>ABCExtension.PreInvokeAIS</td>
<td>False</td>
<td>Decides whether user exit for the preprocess Agile aXML should be called or not</td>
</tr>
<tr>
<td>PARENT_ITEM_REVISION_LEVEL</td>
<td>B01</td>
<td>The item revision that is used in Agile PLM for the parent items updated</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> If the BOM Extract UBE is not set up to retrieve the BOM Parent Item Revision level, then this needs to be populated with the non-blank value; otherwise, the BOM initial load will not function correctly.</td>
</tr>
<tr>
<td>RELEASE_CHANGE_ORDER_REASON</td>
<td>JD Edwards EnterpriseOne to Agile PLM BOM Initial Load</td>
<td>The Change Order Reason to add to the Change Order created in Agile PLM</td>
</tr>
<tr>
<td>RELEASE_CHANGE_ORDER_DESCRIPTION</td>
<td>JDEE1 to Agile PLM BOM Initial Load</td>
<td>The Change Order Description to add to the Change Order created in Agile PLM</td>
</tr>
</tbody>
</table>

Table 7–39 lists the settings for the InitialLoadItemListJDEE1toAgileImpl service properties:

**Table 7–39  Settings for InitialLoadItemList property**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default.SystemID</td>
<td>JDEE1_01</td>
<td>System ID of JD Edwards EnterpriseOne instance</td>
</tr>
<tr>
<td>TRACE.LOG.ENABLED</td>
<td>False</td>
<td>Use tracelog for the flow</td>
</tr>
<tr>
<td>ABCExtension.PreXformABMtoAXML</td>
<td>False</td>
<td>Decides whether user exit for the preprocess ABM should be called or not</td>
</tr>
<tr>
<td>ABCExtension.PreInvokeAIS</td>
<td>False</td>
<td>Decides whether user exit for the preprocess Agile aXML should be called or not</td>
</tr>
<tr>
<td>PRIMARY_ITEM_ID</td>
<td>2ndItemNumber</td>
<td>Property that determines which JD Edwards EnterpriseOne Item Number to use in Agile PLM</td>
</tr>
</tbody>
</table>
Multisite.Enabled property is governed by distributed processing aspects. When it is set to TRUE (default), the Item.UnitCostAttribute, Item.AvailableQuantityAttribute, Item.OnHandQuantityAttribute, and Item.ReservedQuantityAttribute are set to the Site tab flex attributes. You can use the Numeric, Text, or Money flex fields of the Site tab for these settings, and it is denoted by the first element, Site. For example, Site.Numeric01. When set to FALSE, these attributes are set to Page2 or Page3 flex fields, requiring you to change the setting to PageTwo.Numeric01 or PageThree.Numeric01, accordingly. Derive the names of the attributes from the ItemABM schema, which you can find in the Agile PLM Interfaces section.

For more information, see Release of Change Order in Agile PLM and Agile PLM Interfaces sections of this document.

### 7.10 Handling Errors

The Design to Release integration uses the Mediator resequencer to manage errors and failures during web service processing. No PIP-specific errors are issued by the Design to Release services within the AIA system.

For more information about the errors caused by Agile PLM or JD Edwards EnterpriseOne, see the product-specific documentation.

For more information about AIA error handling, see Oracle Fusion Middleware Infrastructure Components and Utilities User’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, “Setting Up and Using Error Handling and Logging.”

### 7.10.1 Resequencing and Error Handling

The initial load flow in the Design to Release Agile PLM - JD Edwards EnterpriseOne integration uses the Mediator resequencer feature to manage errors and failures. The resequencer introduces a data store that stores failed messages until the system successfully processes them. If the message fails, it remains in the resequencer store and blocks any other messages that belong to the same group.

These services have resequencing enabled:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value/Setting (default)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEM_REVISION_LEVEL</td>
<td>A01</td>
<td>The item revision that is used in Agile PLM for all items imported.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This field needs to be populated with the non-blank value; otherwise, the item initial load will not be processed.</td>
</tr>
<tr>
<td>RELEASE_CHANGE_ORDER_REASON</td>
<td>JD Edwards EnterpriseOne to Agile PLM Item Initial Load</td>
<td>The Change Order Reason to add to the Change Order created in Agile PLM</td>
</tr>
<tr>
<td>RELEASE_CHANGE_ORDER_DESCRIPTION</td>
<td>JD Edwards EnterpriseOne to Agile PLM Item Initial Load</td>
<td>The Change Order Description to add to the Change Order created in Agile PLM</td>
</tr>
</tbody>
</table>
Handling Errors

- ItemInitialLoadExtractJDEE1FileConsumer_RS
- ItemInitialLoadExtractJDEE1FTPCongsumer_RS
- BillOfMaterialsInitialLoadExtractJDEE1FileConsumer_RS
- BillOfMaterialsInitialLoadExtractJDEE1FTPCongsumer_RS

ItemInitialLoadExtractJDEE1FileConsumer_RS/ItemInitialLoadExtractJDEE1FTPCongsumer_RS and BillOfMaterialsInitialLoadExtractJDEE1FileConsumer_RS/BillOfMaterialsInitialLoadExtractJDEE1FTPCongsumer_RS are initiated before the initial load item and BillOfMaterials JDEE1toAgile PLM services in their respective flows, and they protect the item and BOMs flows from JD Edwards EnterpriseOne to Agile PLM. If an error occurs in either of these flows, the message remains in the resequencer store that is associated with Mediator, and the system locks the group.

For item processing, the group is defined by the first short item number in the batch. For BOM processing, the group is defined by a first parent short item number in the batch. For example, if an item message for short item 61021 fails and the system then attempts to process item message for item 61021, then the second message will not be processed. It remains in the resequencer store along with the first message until the system successfully processes the first message.

The main purpose of the resequencer during initial load processing is to regulate the initial load processing so that all messages are split into a number of threads defined in the Mediator resequencing configuration. By default, the system creates four groups, which are called 1, 2, 3, and 4.

After unlocking the group, the message that contained the failed record is reprocessed. Therefore, the error should be identified and corrected before unlocking the group. After the failed message is successfully processed, the system processes any messages that follow it in the same group.

You can unlock a group by connecting to the WLS database with the appropriate user name and password and running the following script:

```sql
UPDATE mediator_group_status SET status = '0'
WHERE status!=0 AND group_id='844334' AND component_dn = 'default/ItemInitialLoadExtractJDEE1Consumer!1.0/ItemInitialLoadExtractJDEE1FileConsumer_RS'
COMMIT;
```

**Note:** Obtain the user name and password from your SOA administrator.

**Note:** The values associated with group_id and name should be changed to the appropriate values, where group_id is the name of the group to unlock, and name is the name of the mediator routing service with the resequencer. Then, the message that failed is reprocessed. If it passes, any other messages in that group is processed one at a time until the system encounters another failure or all messages are processed successfully.
7.10.1.1 Logic Used to Determine Notification Roles for an Error

The Error Handling Framework uses runtime values and the data you enter on this page to execute the following hierarchical logic to determine the appropriate notification roles for an error:

- If all four runtime values (SYSTEM_CODE, ERROR_CODE, SERVICE_NAME, and PROCESS_NAME) are available and they map to an error notification entry in this table, use the specified notification roles.
- If the ERROR_CODE, SERVICE_NAME, and PROCESS_NAME are available and map to an error notification entry in this table, use the specified notification roles.
- If the SERVICE_NAME and PROCESS_NAME are available and map to an error notification entry in this table, use the specified notification roles.
- If the SERVICE_NAME is available and maps to an error notification entry in this table, use the specified notification roles.
- If none of these values is available, the system fetches default values from the AIAConfigurationProperties.xml file.

7.10.2 Error Handling in PIP Queue Manager

You can review processes that fail during the change order flow using the Queue Manager. To see why a process failed, click the Errored link for detailed information. Figure 7–13 illustrates how errors appear on the Change Order Queue page:

Figure 7–13 Change Order Queue

Figure 7–14 illustrates a sample error message:
Handling Errors

Figure 7–14  Example of sample error message

The sample error message has two parts:

- Error Text: Text of error source, which can be from any participating ABCS that may have faulted.
- Details: Error Details consist of:
  - Service Name: The name of the service where an instance failed to process
  - Instance ID: The identification number of the instance that failed

Multiple faults generated by the service are captured and appeared in this error message. In addition, any failures in the flows are captured in the AIA error logs. You can see these from the Oracle Enterprise Manager console logs section.

If a service is down and the error is not related to the payload, then users can resubmit the change in the Queue Manager.

Table 7–40 lists the errors generated by the PIP services and the message text of each:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Message Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIA_ERR_AIAAGILEJDEE1_0003</td>
<td>No error message has been specified for the input key</td>
</tr>
<tr>
<td>AIA_ERR_AIAAGILEJDEE1_0004</td>
<td>None of the ECOs selected have been integrated</td>
</tr>
</tbody>
</table>

Table 7–41 lists the errors generated by initial load and the message text for each:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Message Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke AIS Design</td>
<td></td>
</tr>
<tr>
<td>AIA_ERR_AIAAGILEJDEE1_0013</td>
<td>File does not exist:</td>
</tr>
<tr>
<td>Release ECO Design</td>
<td></td>
</tr>
<tr>
<td>AIA_ERR_AIAAGILEJDEE1_0010</td>
<td>Failed</td>
</tr>
<tr>
<td>AIA_ERR_AIAAGILEJDEE1_0011</td>
<td>Release ECO SDK Execution Failed:</td>
</tr>
<tr>
<td>AIA_ERR_AIAAGILEJDEE1_0012</td>
<td>Could not find Change Order workflow:</td>
</tr>
<tr>
<td>AIA_ERR_AIAAGILEJDEE1_0015</td>
<td>The Change Order Not found -</td>
</tr>
<tr>
<td>Initial Load BOM JD Edwards EnterpriseOne to Agile PLM design</td>
<td></td>
</tr>
</tbody>
</table>
Handling Errors

Table 7–41  (Cont.) Errors generated by initial load

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Message Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIA_ERR_AIAAGILEJDEE1_0001</td>
<td>For additional information navigate to &lt;SOA_HOME&gt;/agile/ais/fileResult and review the following file:</td>
</tr>
<tr>
<td>AIA_ERR_AIAAGILEJDEE1_0006</td>
<td>All Bill Of Materials Batch Quantity should be equal to Zero.</td>
</tr>
<tr>
<td>AIA_ERR_AIAAGILEJDEE1_0007</td>
<td>All Bill Of Materials Type should be equal to 'DEFAULT_BOM_TYPE' AIA Configuration Property value.</td>
</tr>
<tr>
<td>AIA_ERR_AIAAGILEJDEE1_0008</td>
<td>Multiple Bill Of Materials for the same Parent Item is not allowed when 'MULTISITE_ENABLED' property is set to 'FALSE'.</td>
</tr>
<tr>
<td>AIA_ERR_AIAAGILEJDEE1_0009</td>
<td>Bill Of Materials Components with 'Non-Stock' Stocking Type is not allowed when 'MULTISITE_ENABLED' property is set to 'TRUE'.</td>
</tr>
</tbody>
</table>

For more information about the errors generated by Agile PLM or JD Edwards Enterprise Business Service (EBS), see the product-specific documentation.

For more information about AIA error handling, see Oracle Fusion Middleware Infrastructure Components and Utilities User’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, “Setting Up and Using Error Handling and Logging.”

7.10.2.1 Mechanism for Error Handling and Reporting

1. Log in to the Oracle Enterprise Manager console.
2. Click Farm_soa_domain > SOA > soa_infra (<server instance>).
3. Select the Instances tab. See Figure 7–15
4. Click the faulted instance (each faulted instance has the symbol in the State field).
5. Click the faulted instance again in the next window. See Figure 7–16

Figure 7–16  Fault Instance

6. The next window displays the details of the flow trace; click the component that faulted. See Figure 7–17
7. The next window displays details of the flow; select either the Audit Trail or the Flow tab from this window.

Figure 7–18 illustrates the example of the Audit Trail window:
Figure 7–18  Audit Trial

Figure 7–19 illustrates the example of Flow window:
7.11 Viewing EIMs

For more information about using XSL Mapping Analyzer (XMAN), see Oracle Fusion Middleware Infrastructure Components and Utilities User’s Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1, Using the XSL Mapping Analyzer.

For more information about how services are mapped, see the My Oracle Support document: EBO Implementation Maps (EIMs) 881022.1.
The Queue Management feature in the pre-built integration requires:

- An event to produce filtered payload to a file destination to a JMS destination
- That the payload be defined using a standard XSD
- That the files or JMS messages produced by events be sequenced in the order in which the objects are released
- A queue to manage the order of messages
- A queue monitoring the user interface to enable reordering and resubmitting of unprocessed messages
- A queue that manages the payloads based on the business process for which the message is produced by the event
- A queue controlling mechanism to:
  - Trigger the business flow based on the business process of the message
  - Process the messages sequentially depending on the order specified in the message (the highest order message is picked first for processing).

  A message is not picked for processing unless the processing of the previous message is complete.

  You can reorder the order of the messages not picked for processing.

For more information about the features and functionalities of Queue Manager, and how to use it, see Agile PLM Integration Pack for Oracle E-Business Suite, Design to Release - User Guide. You can find this document at http://www.oracle.com/technology/documentation/agile.html.

A.1 Queue Management Solution

The Queue Management Solution has the following components:

- Queue DB (database): Persists the data related to a queue messages
Queue Controller: Polls for new event payloads and adds them to the queue DB. The highest priority message for each business process is picked and processed sequentially to trigger its business flow.

Queue Monitoring: The user interface that monitors the queue message status supports reordering of the priorities of the queue messages. It also provides the ability to resubmit unprocessed messages.

A.1.1 Queue Controller

A polling strategy on the queue DB addresses the queue management business requirements. The queue controller provides an ECO system to ensure that this polling strategy works in tandem to ensure that:

- All event transmitted files and JMS messages are added to the queue.
- At any time, only one pending message is in the control table.
- After the processing of a message in the control table is complete, it inserts the highest priority queue message from the queue table to the control table.
- In case the integration flow errors out, the queue manager waits until the message is resubmitted or removed.

A.1.2 Queue Schema

To support the queue controller solution flow, the queue manager uses a polling strategy similar to PollingControlTableStrategy. Two tables manage the sequential processing and reordering of messages.

The first table, QUEUE_TABLE, has all the queue messages that are being provided by the event trigger. The QUEUE_CONTROL_TABLE table stores the relevant information of the message from the QUEUE_TABLE, which has not yet been processed.

The queue manager must ensure that the control table has only one message that is not yet processed. When the processing of a message is complete, a pending message from the queue table is inserted into this table to facilitate the sequential processing of the message. Because all pending messages are stored in the queue table, you can reorder them.

A.1.3 Queue Monitor

When a change order is released by ACS, the queue controller picks it up. The queue monitor displays a list of all the change orders that are waiting to be processed. It also enables you to reorder their processing sequence.

For more information about the Queue Monitor, see Agile PLM Integration Pack for Oracle E-Business Suite Design to Release - User Guide.

A.2 User Interface

The following user interface components are available when working with queues:

- Accessing the Process Queue Monitor
- Fields and Attributes
- Filters
A.2.1 Accessing the Process Queue Monitor

The Process Queue Monitor (User Interface) is deployed at your Integration Server and can be accessed through web browser.

The Integration Administrator is provided with its URL, with login ID and password. After you login, you can see a page similar to the one below:

When a Change Order is released, it is picked up by the Queue Controller, which assigns it an Automated Transfer Object (ATO) Number before passing it on for processing. The Queue Monitor displays this ATO number as Reference Number.

Figure A–1 Process Queue Monitor

A.2.2 Fields and Attributes

The Fields and Attributes component includes the Change Order Queue Monitor description.

A.2.2.1 Change Order Queue Monitor

The Changer Order Queue is a tabular display of the released orders lined up by Queue Manager for processing. Each row in this table is a Change Order. The first row denotes the 'first-in-sequence' Change Order, when it is in Pending state of processing.

Figure A–2 shows the process denoters and their functions:
A.2.2.2 Queue Operators
This section provides a list of queue operators and their operations.

Figure A–3 shows the Change Priorities page

A.2.3 Filters
At any given time, a queue may have hundreds of COs under processing, depending on the size of the organization. Although, the Queue Monitor displays all of them, it gets difficult to find the specific ones that you may require to see quickly.

Queue filters facilitate display of the change orders on the basis of their processing state and further criterion. The tables below list the search criterion of the predefined Saved Search criteria. The bold text indicates the default value or operator.
Figure A–4  Filter All Change Orders

This filter displays the complete list of all change orders in the queue.

Figure A–5 shows the search parameters to be selected to view all change orders.

Figure A–5  All Change Orders

This table shows how to set up a filter to view all change orders:

A.2.3.2 Filter 2: Errored Change Orders Only
This filter displays the change orders with errors.

Figure A–6 shows the search parameters to find change orders with errors.

Figure A–6  Change Orders with Errors

A.2.3.3 Filter 3: Pending Changes Only
This filter displays the pending change orders.

Figure A–7 shows the search parameters to view pending change orders.
This table shows how to set up a filter to view pending change orders:

**A.2.3.4 Filter 4: Completed Change Orders Only**  
This filter displays the completed change orders.  
*Figure A–8* shows the search parameters to view completed change orders.

This table shows how to set up a filter to view completed change orders:

**A.2.3.5 Filter 5: Changes Errored Within Last Week**  
This filter displays the changes errored within last week.  
*Figure A–9* shows changes which errored within last week.

This table shows how to set up a filter to view change orders that errored within last week:

**A.2.3.6 Filter 6: Unprocessed Change Orders**  
This table shows how to set up a filter to view unprocessed change orders:
**A.2.3.7 Filter 7: All Delete Flags**

This table shows how to set up a filter to view all delete flags:

![Figure A–11 All Delete Flags](image)

**A.2.3.8 Advance Query Abilities**

The Advance Query Abilities feature allows you to search for queries using advanced search parameters. You can use the advanced query abilities by clicking the **Advance Query Abilities** button. This button can be used when advanced searching is required and if the pre-defined saved searches do not meet your needs. When this button is pressed, the **Add Fields** button is added next to the **Reset** button. Additional fields can be added to your current search criteria.

**Example:** If the All Change Orders default search criteria was being used, you could also select Change Number from the list below:

![Figure A–12 Advance Query Abilities](image)

Then the Change Number is added to your search criteria. The operator and value can then be entered and a search can be performed. The X button can then be used to delete that additional field. Or the reset button can use to reset your criteria back to the Saved Search criteria.
In addition to the header search criteria, each column in the table (Reference, Change Number, Release Time, Processed Time, Process Status) has a QBE Line that can also be used as a filter.

A.3 Queue Manager Services

The Queue Manager deploys these services:

- CreateQueueService
- CreateQueueControlService
- QueueProcessorService
- QueueProcessorServiceImpl
- CreateQueueService

CreateQueueService is implemented as a Mediator Routing service. An adapter service (File/JMS Adapter) polls the destinations for any event payloads. The payload is in the form of aXML files. This service receives message as a binary element (aXML file). For each payload received, the service inserts a new row in the QUEUE table. An adapter service (DB adapter) is used for the same. The Toplink solution generates the required schema from the table for this DB adapter.

- The service uses transformation services to populate any NOT NULL columns in the table.
- OBJECT_REFERENCE is inserted with the file name of the aXML file using the Mediator header transformation extension functions.
- PROCESS_STATUS is pending for the newly inserted row.
- PROCESS_PRIORITY is captured from the file name. (ACS can be configured to append an default order for the file name)
- CreateQueueControlService

CreateQueueControlService is implemented as a Mediator Routing service. A DB adapter polls on the QUEUE_CONTROL_TABLE table. If no rows are at the pending status, then the CreateQueueControlService invokes a DB adapter service that executes a custom SQL. This SQL identifies the highest priority pending queue message from QUEUE_TABLE table and inserts the same in QUEUE_CONTROL_TABLE table.

This polling strategy ensures that at any time only one pending message is in the QUEUE_CONTROL_TABLE table. After the pending message is processed and its status is completed, the QUEUE_TABLE table inserts a new pending message in the QUEUE_CONTROL_TABLE table. When the status for a message is completed in the QUEUE_CONTROL_TABLE, the system deletes that row from the table.
QueueProcessorService

QueueProcessorService is implemented as a Mediator service that acts as an interface and provides a façade for the QueueProcessorServiceImpl service. A DB adapter polls on the QUEUE_CONTROL table for any pending messages. A pending message in the table is routed to the QueueProcessorServiceImpl service that processes the message. Based on the result from the implementation service, the message status is updated in the control table.

QueueProcessorServiceImpl

The primary task of this service is to invoke the RequestorABCS. The response from RequestorABCS is processed, and the queue is updated with the processing status.

Input: The QueueMessage generated by the Toplink solution in the QueueProcessorService is used as the input for this service.

Output: QueueStatusMessage containing the status and result of processed queue message

This table lists and describes the steps for this process:

A.3.1 Transformations

The aXML payload is transformed to the ABM, which is input for the RequestorABCS. Because the ABM schema is defined on the lines of the aXML schema, this transformation is simpler to do in the JDeveloper XSL Mapper.
A.3.2 Implementation Details

The QueueProcessorServiceImpl is implemented as an asynchronous BPEL process. Calls to the RequestorABCS, DB adapters update queue status and invoke the RequestorABCS. These calls involve some logic (parsing the aXML payload) that cannot be achieved using Mediator.

---

**Note:** The QueueID is used for correlation set between QueueProcessorServiceImpl and RequestorABCS.

---

A.3.3 Error Management

All errors in the integration flow are handled in the RequestorABCS. Any errors leading to failure of the queue processing is handled in this process. Because of such an error, the queue status and result with failure status is updated in the Queue DB.
Mapping Page 2 and Page 3 Attributes

You must insert a specific code to map Page 2 and Page 3 attributes.

B.1 Inserting Code to Map Page 2 and Page 3 Attributes

To map the Page 2 and Page 3 attributes, insert the following code into the AgileCreateEngineeringChangeOrderListABM_to_CreateEngineeringChangeOrderListEBM_Impl.xsl file:

```xml
<xsl:if test="./changeABO:AffectedItem/changeABO:TitleBlock/itemABO:Size">
  <xsl:variable name="VarUOMCODE">
    <xsl:value-of select="./changeABO:AffectedItem/changeABO:TitleBlock/itemABO:Size" />
  </xsl:variable> <corecomEBO:BaseUOMCode>
  <xsl:call-template name="lookupDVM">
    <xsl:with-param name="varDVMName" select="'ITEM_UOM_CODE'" />
    <xsl:with-param name="varSourceValueColumnName" select="'AGILE_01'" />
    <xsl:with-param name="varSourceValue" select="$VarUOMCODE" />
    <xsl:with-param name="varTargetValueColumnName" select="'COMMON'" />
  </xsl:call-template> </corecomEBO:BaseUOMCode>
</xsl:if>
```

**Note:** This sample code is applicable for the Unit of Measure. The variables names may be different from what is listed here. Use appropriate variable names.
This appendix lists a number of common issues and their solutions.

**C.1 Resolving Common Issues**

When creating an ECO, if the system displays the message, "The SQL Exception is: "javax.resource.ResourceException: RollbackException: Transaction has been marked for rollback: Timed out", perform the following steps to resolve the issue:

Solution: Increase the JTA timeout values from the FMW console.

To increase JTA timeout:
1. Log in to the FMW admin console.
2. Navigate to soa_domain > Services > JTA.
3. Set the timeout value.

To increase syncMaxWaitTime:
1. Log in to the FMW Oracle Enterprise Manager console.
2. Expand SOA and right-click soa-infra.
3. Select SOA Administration > BPEL Properties.
4. Click the More BPEL Configuration Properties link and find syncMaxWaitTime.
5. Change it to some higher value and save.

**Issue**: In ECO forward flow, after the ECO is processed successfully the transfer status attribute (flex) in the ECO in Agile PLM is not being updated.

Solution: Check that which flexfield attribute has been enabled corresponding to the change. Then, ensure that the same attribute has been configured in the AIAConfigurationProperties.xml for that property.

**Issue**: For the Item Cost update and Item Balance update flows, the attributes in Agile PLM are not getting updated.

Solution: First, check that whether the Multisite_Enabled property is set to True or False. Based on this given value, ensure that the Cost and Quantity attributes in AIAConfigurationProperties.xml are correctly set.

**Issue**: In Agile PLM ACS, the test for Destination fails with some error.

Solution: If the Agile PLM server and the FMW server are in different domains, then for the ACS to work, an entry should be made in the host file of the two servers.

For Example:
10.176.138.126 aia06.agile.agilesoft.com aia06 - this would go in the FMW server’s host file. 64.181.168.191 sdc78623svqe.corp.siebel.com - this would go in the Agile PLM server’s host file.