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

Implementation Guide

Release 2.5

E15775-03

January 2012

ORACLE®

Oracle Design to Release Integration Pack for Agile Product Lifecycle Management and JD Edwards EnterpriseOne Implementation Guide, Release 2.5

E15775-03

Primary Author: Oracle Corporation

Contents

Preface

This preface discusses:

- Additional resources
- Oracle Application Integration Architecture Concepts and Technologies
- Oracle Application Integration Architecture Core Components
- Oracle Application Integration Architecture Developer's Guide
- Oracle Application Integration Architecture Process Integration Packs

Additional Resources

The following resources are available:

Resource	Location
JD Edwards EnterpriseOne product documentation	Oracle's Technology Network http://www.oracle.com/technology/documentation/jdedent.html
Installation Guide	My Oracle Support https://metalink.oracle.com
Documentation updates	My Oracle Support https://metalink.oracle.com
User Guide	Oracle Technology Network http://www.oracle.com/technology http://www.oracle.com/technology/documentation/agile.html
Release Notes	Oracle Technology Network http://www.oracle.com/technology http://www.oracle.com/technology/documentation/agile.html
Known issues, workarounds, and most current list of patches	My Oracle Support https://metalink.oracle.com

Oracle Application Integration Architecture Concepts and Technologies

Oracle Application Integration Architecture (AIA) Concepts and Technologies is a companion volume to *Oracle Application Integration Architecture Core Components and Oracle Application Integration Architecture Developer's Guide.*

Oracle Application Integration Architecture Concepts and Technologies provides definitions of fundamental Oracle AIA concepts and discusses the following:

- Enterprise Business Objects (EBO) and Enterprise Business Messages (EBM).
- Enterprise Business Services (EBS).
- Application Business Connector Services.
- Interaction patterns.
- Extensibility.
- Versioning.
- Business Processes.
- Batch Processing.
- Infrastructure services.
- Security.

Oracle Application Integration Architecture Core Components

Oracle Application Integration Architecture (AIA) Core Components Guide is a companion volume to *Oracle Application Integration Architecture Concepts and Technologies* and *Oracle Application Integration Architecture Developer's Guide.*

Oracle Application Integration Architecture Core Components discusses how to:

- Work with the Composite Application Validation System (CAVS).
- Work with the Business Service Repository (BSR).
- Set up and use error handling and logging.
- Work with the diagnostics framework.

Oracle Application Integration Architecture Developer's Guide

Oracle Application Integration Architecture (AIA) Developer's Guide is a companion volume to *Oracle Application Integration Architecture Concepts and Technologies Guide* and *Oracle Application Integration Architecture Core Infrastructure Components Guide.*

The *Oracle Application Integration Architecture Developer's Guide* discusses how to:

- Create an integration scenario.
- Define business service patterns.
- Design and develop Enterprise Business Flows.

- Design and construct application business connector services.
- Work with message transformation, enrichment, and configuration.
- Develop custom XPath functions.
- Design and construct JMS Adapter services.
- Work with Enterprise Message Headers.
- Work with message routing.
- Work with transactions.
- Develop Oracle AIA services to work with the Composite Application Validation System (CAVS).
- Configure Oracle AIA processes to be eligible for error handling and logging.
- Extend Enterprise Business Objects.

Oracle Application Integration Architecture Process Integration Packs

A Process Integration Pack (PIP) is a pre-built set of integrated orchestration flows, application integration logic, and extensible Enterprise Business Objects and services required to manage the state and execution of a defined set of activities or tasks between specific Oracle applications associated with a given process. A PIP provides everything you need to deploy a selected integrated Business Process area. The PIP product offering is suited to those customers seeking to rapidly implement a discreet Business Process.

This guide discusses the PIP named ‘Oracle Design to Release Integration Pack for Agile Product Lifecycle Management and JD Edwards EnterpriseOne’. Through the remainder of this document, this PIP is referred to as **‘*Design to Release***: ***Agile PLM - JDE E1*’**.

Chapter 1: Understanding the Oracle Design to Release Integration Pack for Agile Product Lifecycle Management and JD Edwards EnterpriseOne

This chapter provides an overview of the PIP ‘Design to Release: Agile PLM - JDE E1’ and discusses the:

- Overview
- Architecture of Agile PLM Integration
- Business Process Task Flows
- Solution Assumptions and Constraints
- Agile PLM Interfaces
- JD Edwards EnterpriseOne 9.0 Interfaces
- Components of Agile PLM - JDE E1 PIP
- Environment

Overview

Agile PLM enables companies to manage individual product life cycles, complete product portfolios and programs tied to product conception, design, launch, maturity and phase-outs. The primary users of Agile PLM (over 90%) are directly or indirectly involved in the process of designing products. The content managed by these users in Agile PLM often has direct bearing on processes managed in other enterprise systems, such as Enterprise Resource Planning systems (ERP), Manufacturing Execution Systems (MES), Customer Relationship Management systems (CRM) and so forth.

Timely and accurate propagation of product design information from the Product Lifecycle Management (PLM) system to the Manufacturing system is critical for companies to ensure that products are built to the right specs, 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:

a) Delays in product launch (thereby compromising market share and profit margins)

b) Compromised product quality, leading to higher service costs and further loss in market share

c) Expensive inventory write-offs that directly impact the bottom-line

The integration between Agile PLM and JD Edwards EnterpriseOne shall therefore be designed to address the primary use cases around synchronization of product content information between Agile Product Collaboration and JD Edwards EnterpriseOne.

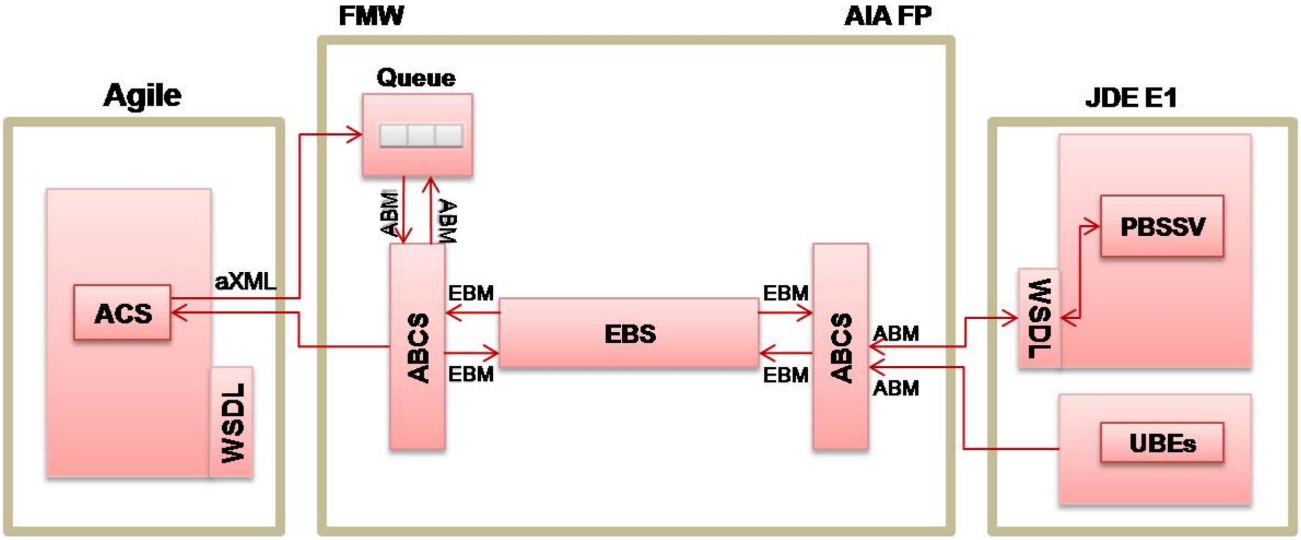
This is the first new generation integration solution between Agile PLM and JD Edwards EnterpriseOne Manufacturing.

The Oracle Design to Release Integration Pack for Agile Product Lifecycle Management and JD Edwards EnterpriseOne consists of the following integration flows:

1. Initial Load of Item and Bill Of Materials from JD Edwards EnterpriseOne to Agile PLM
2. Engineering Change Order Release from Agile PLM to JD Edwards EnterpriseOne through the Engineering Change Order (ECO) process
3. ECO Status from JD Edwards EnterpriseOne to Agile PLM
4. Item Attributes & Cost Updates from JD Edwards EnterpriseOne to Agile PLM
5. Item Balance Updates from JD Edwards EnterpriseOne to Agile PLM

Architecture of Agile PLM Integration

The figure shows the Agile PLM to JDE E1 integration architecture:



Agile PLM to JDE E1 Integration Architecture

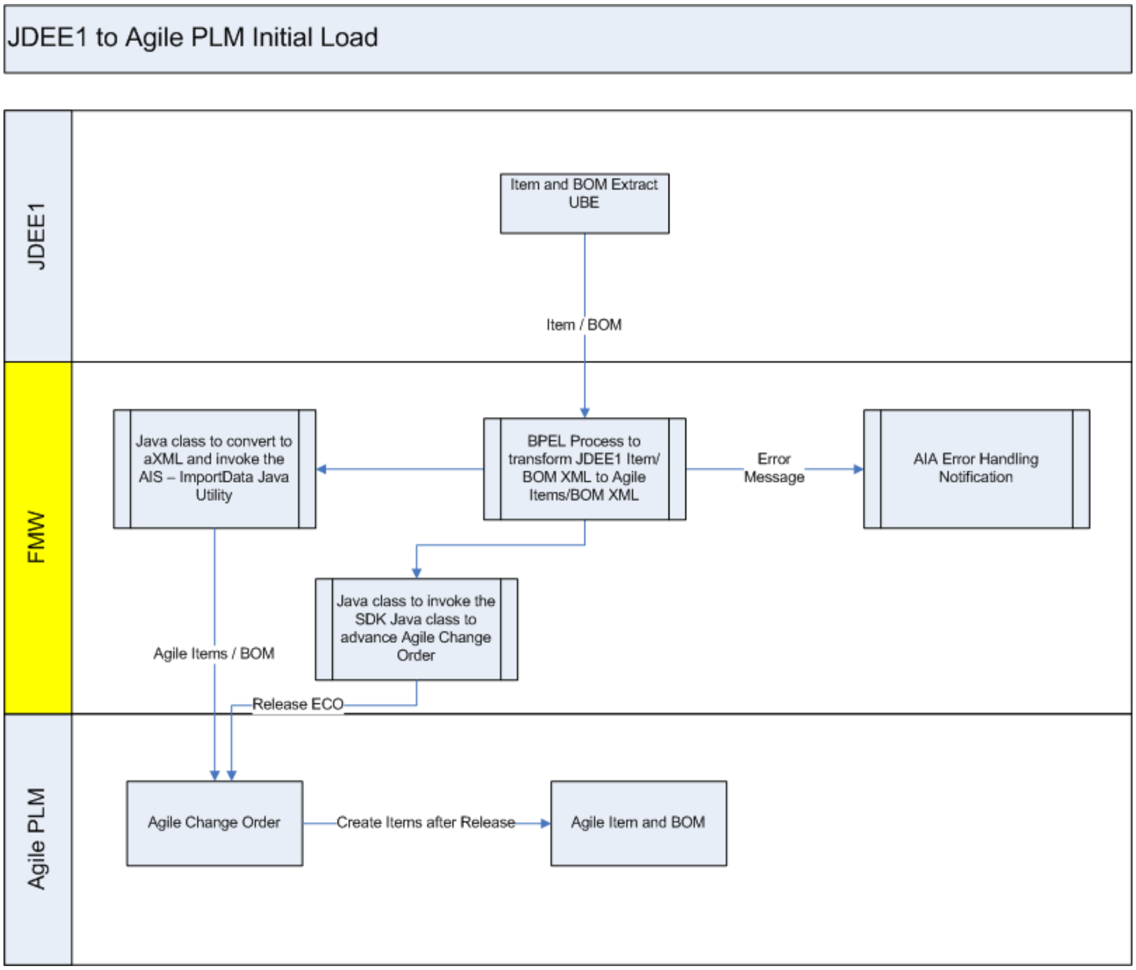
Business Process Task Flows

The integration covers the following business flows:

JDE E1 to Agile PLM Initial Load

Initial Load of Items and Bill of Materials from JDE E1 to Agile PLM: The Initial Load is a process to synchronize Items and BOMs from JD Edwards EnterpriseOne to Agile PLM. This process will be used to synchronize data from an existing JD Edwards EnterpriseOne install to a new installation of Agile PLM. The assumption is that no Items or BOMs exist within the Agile PLM system at the time that the Initial Load is performed. After the initial load, Items and BOMs will be maintained in the Agile PLM system. As the Items and BOMs are modified in Agile PLM, these changes will be pushed to the JD Edwards EnterpriseOne system through using the Engineering Change Order Release integration flow.

This diagram shows the JDE E1 to Agile PLM Initial Load process flow:


JDEE1 to Agile PLM Initial Load
JDEE1
FMW
Agile PLM
Item and BOM Extract UBE
Item / BOM
Java class to convert to aXML and invoke the AIS – ImportData Java Utility
BPEL Process to transform JDEE1 Item/ BOM XML to Agile Items/BOM XML
Error Message
AIA Error Handling Notification
Java class to invoke the SDK Java class to advance Agile Change Order
Agile Items / BOM
Release ECO
Agile Change Order
Create Items after Release
Agile Item and BOM

JDE E1 to Agile PLM Initial Load

Agile PLM to JDE E1 Processes

Engineering Change Order Release

During a product design phase, new products/parts are introduced and/or existing parts go through design changes. When the authoring of a part's attributes and design information is complete and is ready for publishing to the manufacturing system, it is released by means of Engineering Change Orders. The Change Order Release Process constitutes of New Part/Product Release (PREL) and Product Design Modification flows of Agile PLM.

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

Both these processes use the same integration sequence.

For more information, see Chapter 3: Integration Flow for Product Design Changes from Agile PLM to JD Edwards EnterpriseOne.

JDE E1 to Agile PLM Processes

These are the JDE E1 process flows:

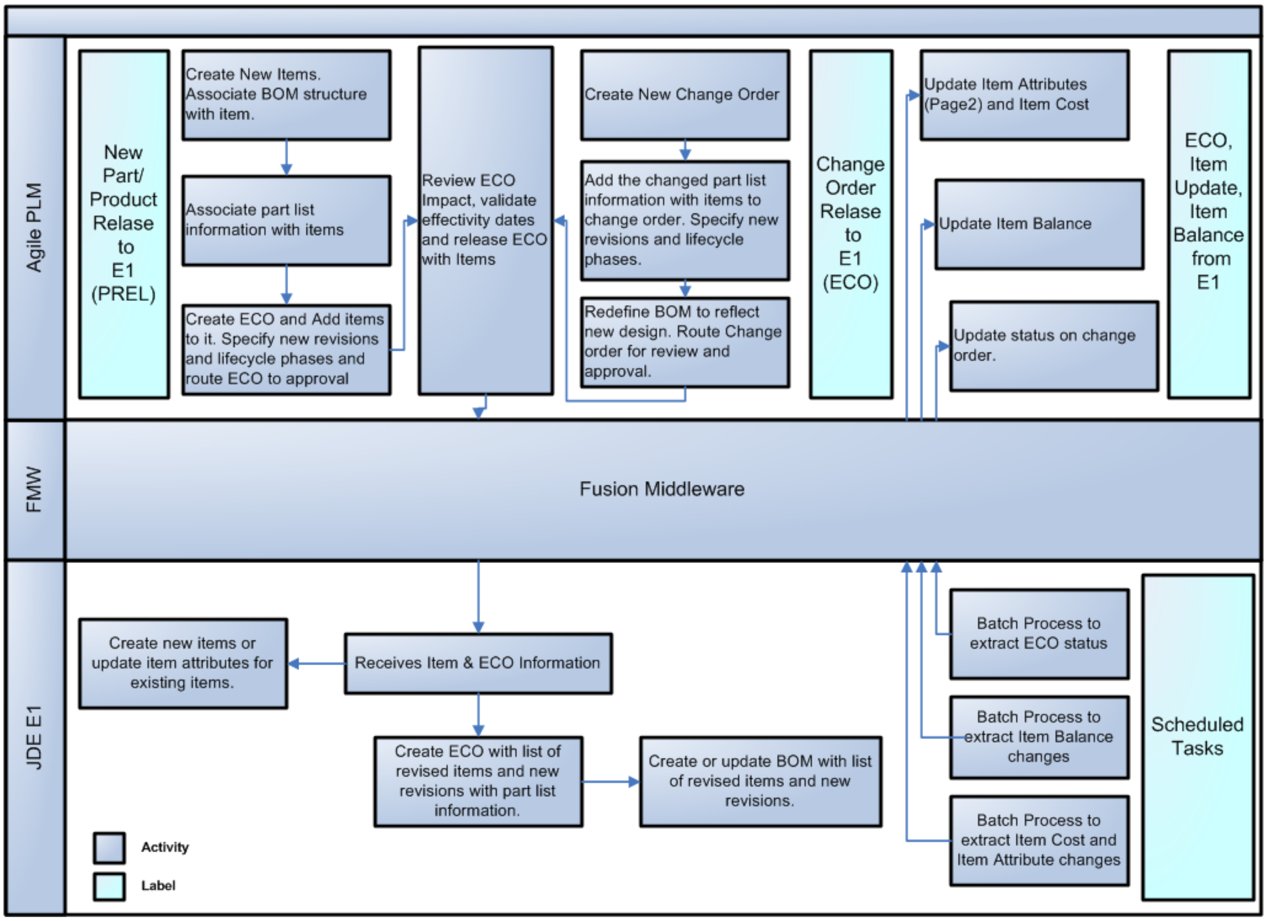
1. **JDE E1 Change Order Status Information to Agile PLM (Batch)**: The Engineering Change Order Update from JDE E1 to Agile PLM is a communication of change in status of the Engineering Change Order in the ERP system (also part of the Manufacturing Update business flow). It constitutes a key requirement for keeping users in Agile PLM apprised of the lifecycle of an Engineering Change Order. This involves communicating the ECO status in the ERP system to a configurable field on the Change Order in the Agile PLM system, and/or changing the status of the Change Order in the Agile PLM system.

   As a necessary part of the Manufacturing Update process, the ability to update a Change Order Line in Agile PLM with updates on ECO Status from the JD Edwards EnterpriseOne system is a key component of the bidirectional synchronization capability of the integration.

2. **JDE E1 Item and Unit Cost Information to Agile PLM (Batch)**: Item and Cost UBE have a Processing Option called ‘Cost Method’. The user has to specify the Cost Method in the Processing Option, so that the UBE extracts the Unit Cost of an Item.

3. **JDE E1 Item Balance Information to Agile PLM (Batch)**: The Item Balance information in the JD Edwards EnterpriseOne system is stored in three fields - Reserved Quantity, Available Quantity and On-hand Quantity. Further, an Item in the JD Edwards EnterpriseOne system can exist in more than one Branch Plants.

Any change in any of the three types of quantities may take place in just one or in a few or in all the Branch Plants. These changes are updated in Agile PLM. Similarly, whenever any change is carried out in the Item information, for example, cost of an item in the JD Edwards EnterpriseOne system, it calls for a corresponding update in Agile PLM system.


Agile PLM
New Part/ Product Relase to E1 (PREL)
Create New Items. Associate BOM structure with item.
Associate part list information with items
Create ECO and Add items to it. Specify new revisions and lifecycle phases and route ECO to approval
Review ECO Impact, validate effectivity dates and release ECO with Items
Create New Change Order
Add the changed part list information with items to change order. Specify new revisions and lifecycle phases.
Redefine BOM to reflect new design. Route Change order for review and approval.
Change Order Release to E1 (ECO)
Update Item Attributes (Page2) and Item Cost
Update Item Balance
Update status on change order.
ECO, Item Update, Item Balance from E1
FMW
Fusion Middleware
JDE E1
Create new items or update item attributes for existing items.
Receives Item & ECO Information
Create ECO with list of revised items and new revisions with part list information.
Create or update BOM with list of revised items and new revisions.
Batch Process to extract ECO status
Batch Process to extract Item Balance changes
Batch Process to extract Item Cost and Item Attribute changes
Scheduled Tasks
Activity
Label

Agile PLM to JDE E1 flow

Solution Assumptions and Constraints

Design Assumptions

- Agile Content Server will be the used for Events to trigger the payload from Agile PLM to the Integration.
- This design assumes that the following statements are true:
  1. There will be pre-defined blank templates made available for Custom fields
  2. Transformation logic for classification elements will be pre-coded in the OOTB XSL, but you may need to modify it to suit your Agile PLM implementation
- This design leverages AIA error handling framework.
- AML (Approved List of Manufacturers) information will not be integrated. Any AML information passed to JDE E1 in this PIP will not be imported into JDE E1.

Design Constraints

- In a few cases configuration driven XSLT may not reflect the changes immediately and may require a restart since the main XSL sheet is cached after a successful compilation

- Error handling capabilities of this integration are constrained by the capabilities of the AIA framework.

Agile PLM Interfaces

These are the Agile PLM interfaces:

Web Services Definition Language (WSDLs)

ProcessEngineeringChangeOrderAgileReqABCS	
ChangeABS.wsdl	Used to send the released ECO to be created as an EngineeringChangeOrder in JDE E1
UpdateItemListAgileProvABCSImpl	
ItemABS.wsdl	Used to update back any Item Attribute changes done in JDE E1 into Agile PLM
UpdateItemBalanceListAgileProvABCSImpl	
ItemABS.wsdl	Used to update back any Item Availability changes done in JDE E1 into Agile PLM

XML Schema Definition (XSDs)

UpdateItemListAgileProvABCSImpl	
ItemABM.xsd ItemABO.xsd	Contains the Update Item List Request and Response Application Business Message (ABM)

JD Edwards EnterpriseOne 9.0 Interfaces

These are the JDE E1 interfaces:

Web Services Definition Language (WSDLs)

CreateEngineeringChangeOrderListJDEE1ProvABCSImpl	
EngineeringChangeOrderManager.wsdl	Used to create an ECO in JDE E1
UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl	
IntegrationTimeStampManager.wsdl	Used to update an Time Stamp file F0095 in JDE E1
UpdateItemListJDEE1ReqABCSImpl	

IntegrationTimeStampManager.wsdl	Used to update an Time Stamp file F0095 in JDE E1
UpdateItemBalanceListJDEE1ReqABCSImpl	
IntegrationTimeStampManager.wsdl	Used to update an Time Stamp file F0095 in JDE E1

XML Schema Definition (XSDs)

InitialLoadItemListJDEE1toAgileImpl	
R4101D3.xsd	Contains Item ABM
InitialLoadBillOfMaterialsListJDEE1toAgileImpl	
R3002D.xsd	Contains BOM ABM
UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl	
R3013D.xsd	Contains the Update ECO Request ABM
UpdateItemListJDEE1ReqABCSImpl	
R4102D.xsd	Contains the Update Item List Request ABM
UpdateItemBalanceListJDEE1ReqABCSImpl	
R41021D.xsd	Contains Update Item Balance List Request ABM

Components of Agile PLM - JDE E1 PIP

Oracle Fusion Middleware

Oracle Fusion Middleware (OFM) is a portfolio of software products, produced by Oracle, which spans multiple services, including J2EE and developer tools, integration services, business intelligence, collaboration, and content management.

Oracle Fusion Middleware is designed to support development, deployment, and management of Service-Oriented Architecture. It includes what Oracle calls "Hot-Pluggable" architecture, which allows users to leverage existing investments in applications and systems from other software vendors such as IBM, Microsoft, and SAP.

Business Process Execution Language (BPEL)

BPEL is the technology that integrates and assembles the Web Services. BPEL is a XML-based workflow definition language that allows businesses to describe inter or intra enterprise business processes, which are connected using Web services. BPEL opens a completely new way or at least enhanced way, for software development for mainstream business applications to allow a programmer to describe a business process that will take place across the Internet.

BPEL provides an XML-based grammar for describing the logic to control and coordinate Web services participating in a process flow.

Enterprise Service Bus (ESB)

An Enterprise Service Bus (ESB) is a software architecture for middleware that provides fundamental services for more complex architectures and can be thought of as a mechanism, which manages access to applications and services.

Oracle Service Registry (OSR)

Oracle Services Repository is a Web Services Registry and Repository for building your Service-Oriented Architectures (SOA). It provides a UDDI v3-compliant platform for publishing, categorizing and discovering Web services and related resources across the enterprise. OSR enables service providers to expose and advertise service offerings, allows service consumers to find access and invoke services that meet defined criteria, and provides critical features for SOA governance.

Application Integration Architecture (AIA)

Oracle Application Integration Architecture provides an open-standards-based framework for creating cross-application business processes that support the way you run your business today, while paving the way for your long term, strategic, business transformation plans. Its application independent framework enables you to utilize the applications of your choice to create composite business processes unique to your business, on a flexible service-oriented architecture.

Agile PLM

Agile Content Service (ACS)

Agile Content Service is an event-driven XML-based publishing service that makes the product record 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, Agile Content Service can be configured to automatically publish 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.

The output generated by an ACS module is an aXML file or a PDX package.

Agile Integration Services (AIS)

Agile Integration Services (AIS) are a collection of predefined Web services in the Agile Integration Framework to enable communication between the Agile PLM server and disparate systems, including Enterprise Resource Planning (ERP) systems, Customer Resource Management (CRM) systems, Business-to-Business Integration systems (B2Bi), 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 can be used to:

- Make product content available to Enterprise Application Integration (EAI) systems.
- Share product content with product design, manufacturing planning, shop floor, Enterprise Resource Planning (ERP), and Customer Relationship Management (CRM) applications.

- Make product content available to Business-to-Business (B2B) systems, which 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 (SDK)

The **Agile SDK** is a **software development kit** that contains a collection of tools, application programming interfaces (APIs), sample applications, and documentation. It is used to build custom applications that access the Agile Application Server functionality. By using the Agile SDK, you can create programs that perform tasks automatically against the Agile PLM system.

The Agile SDK enables the following operations:

- Integrate the Agile PLM system with JDE E1 applications or other custom applications.
- Develop applications to process product data.
- Perform batch operations against the Agile Application Server.

The Agile SDK has the following three modules:

- **Agile API** — A Java API with interfaces that expose Agile PLM business objects. The Agile API is used to create additional Agile PLM clients, or used as part of an extension developed using WSX or PX.
- **Process extensions (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 custom actions, custom tools, and custom auto number sources.
- **Web service extensions (WSX)** — A framework that allows Agile PLM customers to extend the functionality of the Agile PLM server and expose customer-specific solutions using Web service.

JD Edwards EnterpriseOne

Published Business Service

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 and in JD Edwards EnterpriseOne, web services are referred to 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 up 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 up the signature for the business service methods. These public methods can be called from other business service classes and published business service classes.

Universal Batch Engine (UBEs)

UBE is a Universal Batch Engine for data extraction, transformation, publication and distribution. UBEs are also used to generate various outputs like Operational Document Creation, Customer-facing Documentation, Ad-Hoc Reporting, Financial Reporting, Regulatory Reporting and Analytics.

Environment

Agile PLM	9.2.2.6 on Oracle Application Server 10.1.3.4
Oracle SOA Suite	10.1.3.4
Oracle AIA Foundation Pack (AIA FP)	2.5
Oracle Database	Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 DB upgrade patch 10.2.0.3
Oracle Service Registry	10.1.3.1
Java 2 Platform Standard Edition (J2SE) Development Kit (JDK)	5.0, Update 16
JD Edwards EnterpriseOne	9.0 (JDE tools set 8.98.2)

Chapter 2: Integration Flow for Initial Load

This chapter provides an overview of initial data loads and discusses:

- Overview
- Prerequisites
- AIA Core Components
- Initial Load Assumptions/Constraints

Overview

The Design to Release PIP requires that Item and Bill of Materials data in the JD Edwards EnterpriseOne system is initially synchronized to the new instance of Agile PLM. This Initial Load is a one-way process that loads data from the JD Edwards EnterpriseOne system into the Agile PLM system in bulk.

The assumption is that no Items or BOMs exist within the Agile PLM system at the time that the Initial Load is performed. After the initial load, Items and BOMs will be maintained in the Agile PLM system. As the Items and BOMs are modified in Agile PLM, these changes will be pushed to the JD Edwards EnterpriseOne system through using the Engineering Change Order Release integration flow.

Note. The initial load process is strongly recommended for one time initial load only. After the successful initial load from the JD Edwards EnterpriseOne system into the Agile PLM system, it is not recommended to rerun this process over the same data. However, after you have loaded business data initially, you can run the initial load process to load new data that has not been previously integrated.

To load data from JD Edwards EnterpriseOne to Agile PLM, extract programs in the JD Edwards EnterpriseOne system need to be run. These programs generate XML files that containthe data that will be uploaded into the Agile PLM system. These files are written to folders on theJD Edwards EnterpriseOne Enterprise 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 in later sections of this chapter.

After creating the XML files, there are two methods that can be used to pass that data to the Agile PLM system:

- File Transfer Protocol (FTP)
- Oracle Application Server

If the FTP method is being used, the Initial Load process for Item and Bill of Materials data requires that an FTP adapter be configured to monitor the folders for newly created extract files. When the FTP adapter locates a new file; the appropriate Enterprise Service Bus (ESB) process debatches the files into separate instances. Debatching is used to split large XML files into several smaller XML files.

Alternatively, the XML files can be moved to JDE E1 in the folder on the Oracle Application Server. If this method is used, 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 individual XML file to the appropriate Item/Bill of Materials Initial Load JDE E1 to Agile PLM BPEL Service. This BPEL service will then invoke a Java utility to execute Agile Integration Services (AIS) which will import the data through a change order. Then another Java utility will be invoked to execute yet another Java utility built with Agile SDK to release the Change Order.

Note. If the FTP method is used for Initial Load, an FTP Server should be configured on the JD Edwards EnterpriseOne Enterprise Server. The FTP adapter should be configured to use the FTP connection that is set up on the Enterprise Server.

Prerequisites

Before running the Initial Loads, the following needs to be completed:

1. Set up versions of each extract program in JD Edwards EnterpriseOne.
2. Set up Batch Processing information and Invoke utilities for Initial Load.
3. Create New Workflow for Initial Load Change Orders and set privileges.

AIA Core Components

Agile PLM & JD Edwards E1 Components for Initial Load

Services	BOM	Item
ABMs	R3002.xsd - JDE E1 BOM ABM aXML.xsd - Agile PLM BOM ABM	R4101D3.xsd - JDE E1 Item ABM aXML.xsd - Agile Item ABM
BPEL	InitialLoadBillOfMaterialsListJDEE1toAgileImpl	InitialLoadItemListJDEE1toAgileImpl
ESBs	BillOfMaterialsInitialLoadExtractJDEE1FileConsumer_RS BillOfMaterialsInitialLoadExtractJDEE1FTPConsumer_RS BillOfMaterialsInitialLoadExtractJDEE1FileConsumer BillOfMaterialsInitialLoadExtractJDEE1FileConsumer	ItemInitialLoadExtractJDEE1FileConsumer_RS ItemInitialLoadExtractJDEE1FTPConsumer_RS ItemInitialLoadExtractJDEE1FileConsumer ItemInitialLoadExtractJDEE1FTPConsumer

Component Locations

ABO, ABM & Common XSD files	http://<servername>:<portname>/AIAComponents/ApplicationObjectLibrary/JDEE1/V1/schemas http://<servername>:<portname>/AIAComponents/ApplicationObjectLibrary/JDEE1/V1/schemas/AgileInitialLoad

Initial Load Assumptions/Constraints

This design assumes that the following statements are true:

- When installing the PIP, the information from JDE E1 system does not exist in Agile PLM.
- All Items with Branch Plants have corresponding Sites set up in Agile PLM through 'AGILE_TARGET_SITE_MAPPING' Domain Value Map (DVM) when Multi Site is enabled in Agile PLM.
- Item Initial Load batch containing all not stock items will not be loaded into Agile PLM when Multi Site is enabled; The JDE E1 user can add a data selection of STKT != 'N' in **R4101D3** to filter out non-stock items if he/she chooses not to send them.
- **R3002D** – BOM Initial Load Extract UBE is run with appropriate processing options set up to retrieve the Parent Item Revision Level; otherwise PARENT_ITEM_REVISION_LEVEL Property in AIA Configuration Properties is populated with the value to be used as a Parent Item Revision Level.
- Only Bill of Materials type that matches with "DEFAULT_BOM_TYPE" Property in AIA Configuration Properties will be accepted. The JDE E1 user can add a data selection on TBM field to match with the value defined in the 'DEFAULT_BOM_TYPE' Property.
- Only Bill of Materials with a Batch Quantity of Zero (0) will be accepted. The JDE E1 user can add a data selection on BQTY = 0 to extract Bill of Materials with Batch Quantity equals to zero.
- Bill of Materials components that only have the same Branch Plant as Parent Branch Plant will be loaded into Agile PLM; other components will be filtered out. The JDE E1 user can filter these components by setting the **R3002D** Processing option "Selection for Components" with blank value.
- Non-stock Bill of Materials components will not be accepted when Multi-Site is enabled in Agile PLM. The JDE E1 user can choose not to send non-stock components in an Agile PLM No-Site configuration by adding a data selection for STKT != 'N' when running **R3002D** UBE.
- Multiple Bills of Material for the same Parent Item will be considered duplicate and will not be accepted when Multi-Site is not enabled in Agile PLM. The JDE E1 user can set appropriate data selections for Branch (MMCU), BOM Type (TBM), Batch Quantity (BQTY) and Batch UOM (UOM) such that only single Bill of Materials is selected for a parent Item.

Initial Load Flow

Here are integration load process flows:

Item Initial Load

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 the items that are extracted from JD Edwards EnterpriseOne and loaded into Agile PLM using selection criteria, which includes category codes.

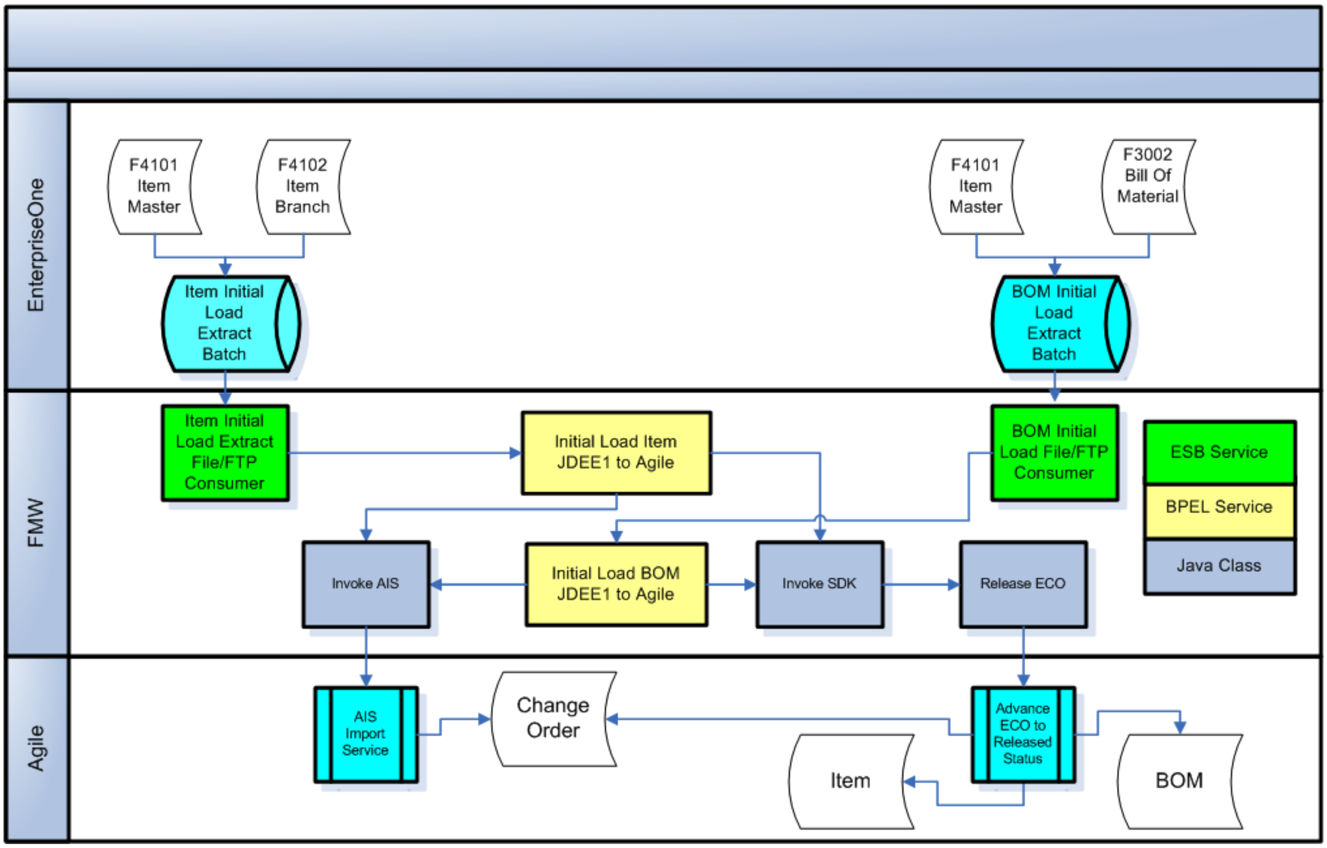
To extract Initial Load item records from the JD Edwards EnterpriseOne system, you should run the Item Initial Load Extract batch program (**R4101D3**). This program creates an XML file, and if it completes successfully, a PDF is created with a successful completion message and the number of items extracted.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

After the files have been written to the Enterprise Server, one of these consumer services retrieves and debatches the XML files:

- ItemInitialLoadExtractJDEE1FTPConsumer
  This consumer service is used when you configure the FTP adapter to locate files.

- ItemInitialLoadExtractJDEE1FileConsumer
  This consumer service is used when you move your files to the JDE E1In file on the Oracle Application Server.

After the file has been written to the Enterprise Server, the appropriate consumer service retrieves and debatches the XML file. 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 individual XML file to InitialLoadItemJDEE1toAgileImpl. This BPEL service will transform the list of items in R4101D3.xml into aXML.xsd format and write the file with the name agile<BPELInstance>.xml. The BPEL will then invoke a Java utility which will pick up the agile<BPELInstance >.xml file, zip it and save it as agile<BPELInstance >.axml. Then the utility will invoke 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. Then another Java utility will be invoked to call an ant script to invoke yet another Java utility built with Agile SDK to release the Change Order.


EnterpriseOne
FMW
Agile
F4101 Item Master
F4102 Item Branch
F4101 Item Master
F3002 Bill Of Material
Item Initial Load Extract Batch
BOM Initial Load Extract Batch
Item Initial Load Extract File/FTP Consumer
Initial Load Item JDEE1 to Agile
BOM Initial Load File/FTP Consumer
ESB Service
BPEL Service
Java Class
Invoke AIS
Initial Load BOM JDEE1 to Agile
Invoke SDK
Release ECO
AIS Import Service
Change Order
Advance ECO to Released Status
Item
BOM

Item Initial Load Process Flow

BOM Initial Load

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

Users can select the Bill of Material 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 the JD Edwards EnterpriseOne system, you should run the BOM Initial Load Extract batch program (**R3002D**). This program creates an XML file, and if it completes successfully, a PDF is created 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. You specify the location of the folder in the processing options of the **R3002D** program.

After the file has been written to the Enterprise Server, the appropriate consumer service retrieves and debatches the XML file. 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 individual XML file to InitialLoadBillOfMaterialsJDEE1toAgileImpl.

This BPEL service will transform the list of BOMs in R3002.xml into aXML.xsd format and write the file with the name agile<BPELInstance>.xml. Then the BPEL will invoke a Java utility which will pick up the agile< BPELInstance >.xml file, zip it and save as agile< BPELInstance >.axml. Then the utility will invoke an ant script to call an Agile Integration Service (AIS) to import the data stored in the agile<SEQ>.axml file.

Then another Java utility will be invoked to call an ant script to invoke yet another Java utility built with Agile SDK to release the Change Order.

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

- BillOfMaterialsInitialLoadExtractJDEE1FTPConsumer

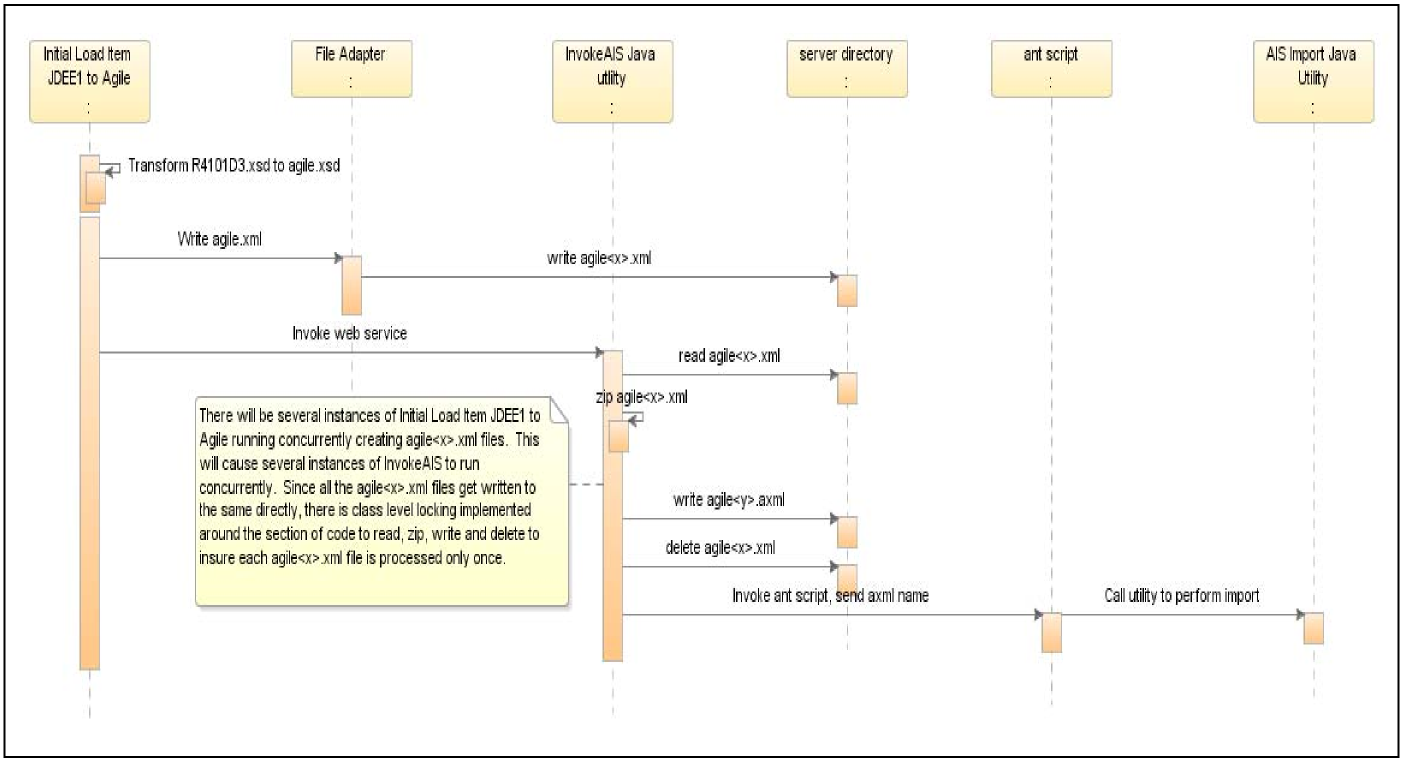
  This consumer service is used when you configure the FTP adapter to locate files.

- BillOfMaterialsInitialLoadExtractJDEE1FileConsumer

  This consumer service is used when you move your files to the JDEE1In file on the Oracle Application Server.

Initial Load Orchestration

This diagram shows the Initial Load Orchestration sequence:



There will be several instances of Initial Load Item JDEE1 to Agile running concurrently creating agile<x>.xml files. This will cause several instances of InvokeAIS to run concurrently. Since all the agile<x>.xml files get written to the same directly, there is class level locking implemented around the section of code to read, zip, write and delete to insure each agile<x>.xml file is processed only once.

Initial Load Orchestration

This table lists the activities involved in each flow:

#	Activity	Remarks
	Item Initial Load	
1	Run R4101D3 to export items	R4101D3_mmddyy_hhmmss.xml file will be placed on the location specified in the UBE Processing option.
2	ItemInitialLoadExtractConsumer ESB reads/debatches file	The batch size will be determined from

#	Activity	Remarks
		'BatchSize' Property in ItemInitialLoadExtractJDEE1FileConsumer
3	InitialLoadItemListJDEE1toAgileImpl BPEL service is invoked	ItemList ABM is transformed into ItemListAXML and agile<instanceid>.xml is written with the file adapter.
4	AIS Importer is invoked to create change order	AIS will add Affected Items into an ECO and import items in Redline mode.
5	AIS Result.xml file will be retrieved for error handling	Result<InstanceId>.xml fill will be retrieved through file adapter and stops processing in case of any errors.
6	SDK methods to release change order are invoked	ReleaseECO SDK will be invoked to advance the ECO to Released status
7	SDK Result.xml file will be retrieved for error handling	Result<InstanceId>.xml fill will be retrieved through file adapter and stops processing in case of any errors.
BOM Initial Load		
1	Run R3002D to export BOMs	R3002D _mmddyy_hhmmss.xml file will be placed on the location specified in the UBE Processing option.
2	BillOfMaterialsInitialLoadExtractConsumer ESB reads/debatches file	The batch size will be determined from 'BatchSize' Property in BillOfMaterialsInitialLoadExtractConsumer
3	InitialLoadBillOfMaterialsListJDEE1toAgileImpl BPEL service is invoked	ItemBOM ABM is transformed into ItemBOMAXML and agile<instanceid>.xml is written with the file adapter.
4	AIS Importer is invoked to create change order	AIS will add Affected Items and BOM into an ECO and import BOM in Redline mode.
5	AIS Result.xml file will be retrieved for error handling	Result<InstanceId>.xml fill will be retrieved through file adapter and stops processing in case of any errors.
6	SDK methods to release change order are invoked	ReleaseECO SDK will be invoked to advance the ECO to Released status
7	SDK Result.xml file will be retrieved for error handling	Result<InstanceId>.xml fill will be retrieved through file adapter and stops processing in case of any errors.

Chapter 3: Integration Flow for Product Design Changes from Agile PLM to JD Edwards EnterpriseOne

This chapter provides an overview of integration flow for product design changes from Agile PLM to JD EE1 and discuses:

- Overview
- Prerequisites
- AIA Core Components
- Integration Services
- Create ECO Solution Assumptions/Constraints

Overview

Both the New Part/Product Release process and the Product Design Modification process within Agile, result in an ECO being published. The ECO resulting from a new product will contain lines with all of the new components used on the BOM followed by the BOM details. Similarly, a product design modification will result in an ECO with all of the new or changed components used on the BOM followed by just the BOM changes.

Agile PLM can send all BOM information or just net changes to a BOM on an ECO. It is important that when a new product is released from Agile PLM, all the BOM information be included. Once the ECO is created and processed in JDE E1, only net change should be sent on the ECO.

The ECO EBM generated by Agile PLM will be sent to *EngineeringChangeOrderEBS*. *EngineeringChangeOrderEBS* does the routing to the JDE E1 Provider Application Business Connect Services (ABCS) – *CreateEngineeringChangeOrderListJDEE1ProvABCSImpl*. The *CreateEngineeringChangeOrderListJDEE1ProvABCSImpl* in turn, performs the following actions:

- Transforms ECO EBM to ECO ABM for JDE E1
- Invokes the JDE E1 ECO Processor Web Services

The JDE E1 ECO Processor provides a response to the requestor by invoking the *EngineeringChangeOrderEBSResponse.*

Note. The Create ECO flow is a one way flow from Agile PLM to JDE E1. Any ECOs created in JDE E1 will not be sent to Agile PLM.

Prerequisites

Before running the Create ECO flow, the following needs to be completed:

1. Set up UDCs.
2. Set up JDE E1 Web Services.
3. Set up OWSM security information.

AIA Core Components

Core AIA Components for Create ECO

The Process Integration for Create ECO uses the following industry components:

Enterprise Business Objects (EBOs)	EngineeringChangeOrderEBO
Enterprise Business Messages (EBMs)	CreateEngineeringChangeOrderEBM CreateEngineeringChangeOrderResponseEBM
Enterprise Business Services (EBSs)	EngineeringChangeOrderEBS

Core Components Locations

EBO & EBM XSD files	http://[HOST:PORT]/AIAComponents/EnterpriseObjectLibrary/Core/EBO/
WSDL files	http://[HOST:PORT]/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/

- For detailed documentation of individual EBOs, the EBO Name link can be clicked on the *Integration Scenario Summary* page in the *Oracle AIA Console*. The *Integration Scenario Summary* page can also be used to search for and view integration scenarios that utilize a particular EBO or EBS.

  For more information, see Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide, “Using the BSR,” Using the BSR UI to View Integration Scenarios.

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

For more information, see Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide, "Extensibility for Oracle AIA Artifacts."

Agile PLM & JD Edwards E1 Components for Create ECO

Services	Agile PLM (Requester)	JD Edwards EnterpriseOne (Provider)
ABMs	AgileCreateItemABM AgileCreateItemResponseABM	EngineeringChangeOrder
ABCS	ProcessEngineeringChangeOrderAgileReqABCS	CreateEngineeringChangeOrderListJDEE1ProvABCSImpl
EBS	EngineeringChangeOrderEBS	EngineeringChangeOrderResponseEBS

Component Locations

ABO, ABM & Common XSD files	http://<servername>:<portname>/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas
WSDL files	http://<servername>:<portname>/AIAComponents/ApplicationObjectLibrary/Agile/V1/wsdls

Integration Services

These are the delivered services:

1. EngineeringChangeOrderEBS

EngineeringChangeOrderEBS is the Enterprise Business Service, which exposes the operations related to the EngineeringChangeOrder Integration on the EngineeringChangeOrder EBO. The following are the routing rules:

EngineeringChangeOrderEBS ESB service

1. ProcessEngineeringChangeOrderAgileReqABCSImpl

   Route CreateECOEBM to CreateEngineeringChangeOrderListJDEE1ProvABCSImpl

EngineeringChangeOrderResponseEBS ESB service

2. CreateEngineeringChangeOrderListJDEE1ProvABCSImpl

   Route CreateECOResponseEBM to ProcessEngineeringChangeOrderAgileReqABCSImpl

2. ProcessEngineeringChangeOrderAgileReqABCSImpl

ProcessEngineeringChangeOrderAgileReqABCSImpl transforms the Agile PLM message (AgileCreateECOABM) into CreateECOEBM & invokes the EBS service to Create ECO(s) in JDE. JDE processes these ECO(s) and sends a CreateECOResponseEBM response back to Agile PLM. ProcessEngineeringChangeOrderAgileReqABCSImpl receives the CreateECOResponseEBM and transforms it to Agile ABM message (AgileCreateECOResponseABM) before invoking Agile Content Server (ACS).

Here is the process flow:

1. Receives *AgileCreateECOABM* from ProcessEngineeringChangeOrderAgileReqABCSImpl, which gets the same from Agile Content Server.
2. Transforms this message into the CreateECOEBM -

a. Populates EBM header

Determines Target System ID(s) and adds into the EBM header to control routing

b. Validates Required fields

c. Transforms ABM to EBM

Note. Refer mapping sheet in Appendices for complete details of field mappings.

3. Call Target 'EngineeringChangeOrderEBS' Service with operation CreateECO
4. Receives CreateECOResponseEBM.
5. Transforms Response EBM to AgileCreateECOResponseABM message.

Note. Refer mapping sheet for complete details of field mappings.

Invokes ACS for propagating the response to Queue and the Agile PLM system.

ProcessEngineeringChangeOrderAgileReqABCSImpl has the following transformation:

AgileCreateECOListABM_to_CreateECOListEBM
CreateECOListResponseEBM_to_AgileCreateECOListResponseABM

3. CreateEngineeringChangeOrderListJDEE1ProvABCSImpl

CreateEngineeringChangeOrderListJDEE1ProvABCSImpl receives the *CreateECOListEBM* message from ECO*EBS*, transforms the EBM into JDE E1 specific *CreateECOABM*. It then invokes the JDE E1 EngineeringChangeOrderManager service for creating/updating Items and creating ECO(s). JDEE1EngineeringChangeOrderManager service returns the list of item(s) and ECO(s) created in case of success or throws the appropriate fault in case of a failure.

After this It then transforms JDE E1 specific response message *CreateECOResponseABM* to *CreateECOResponseEBM* and invokes the routing service *'ECOResponseEBS'* for sending the response message back to Agile PLM.

Here is the process flow:

1. Receives CreateECOEBM from EngineeringChangeOrderEBS.
2. Transforms CreateECOEBM into JDE E1 specific message CreateECOABM.
   - Populates EBM header.

     Determines Target System ID(s) and adds into the EBM header to control routing.

   - Validates required fields.

3. Assigns “Implemented” status if the payload contains “Items” alone. If the payload contains BOM information also, the status is not tampered with.
4. Uses DynamicPartnerlink to determine TargetEndpointLocation.
5. Invokes EngineeringChangeOrderManager web service.
6. Transforms JDE E1 specific message CreateECOResponseABM into the CreateECOResponseEBM.
7. Invokes 'ECOResponseEBS' Service to send the response message (CreateECOResponseEBM) to ProcessEngineeringChangeOrderAgileReqABCSImpl

ProcessEngineeringChangeOrderListJDEE1ProvABCSImpl has the following transformation:

- Xform_EngineeringChangeOrderListEBM_To_EngineeringChangeOrderManagerABM
- Xform_EngineeringChangeOrderManagerRespMsg_To_EngineeringChangeOrderEBSResponseMsg

Create ECO Solution Assumptions/Constraints

This design assumes that the following statements are true:

- JDE E1 system should NOT be used to create any New Item or BOM
- A change order in Agile PLM is a generic term which is used to describe ECO, Manufacturing Change Order (MCO) or Site Change Order (SCO).
- MCOs are not supported in this PIP. This integration supports SCOs / ECOs with BOM information only.
- AML (Approved List of Manufacturers) information will not be integrated. Any AML data passed to JDE E1 in this PIP will not be imported into JDE E1.
- ECO should be used to transfer the new part/update part information from Agile PLM to the JDE E1 system. The Agile PLM user needs to ensure that the New/Updated items are available on the ‘Affected Items’ tab before the ECO is released to JDE E1.
- A mechanism should be provided for error handling/reporting to the end user.

1. **For more information**, see **Handling Errors**

- Use ECOs to redline BOM for associating new revision(s), Effectivity Date changes and inclusion of new items on the Bill of Material. When new items are created, the Agile PLM user needs to ensure that the New item(s) are available on the ‘Affected Items’ tab before the ECO is released to JDE E1.
- Agile PLM user cannot redline a BOM and add a new site in one Change Order. A separate ECO needs to be created for propagating new site information.
- If multiple sites are associated in Agile PLM for a single ECO, JDE E1 will have multiple ECOs created for each Branch Plant (site).
- ECOs have to be approved in Agile PLM before the start of the integration flow.

- As these ECOs are pre-approved in Agile PLM, an approval layer in JDE E1 again is not required before these ECOs get implemented through the scheduled Batch Process – R30510 for creating BOMs. However if the JDE E1 user decides to add an approval layer, any rejection in this approval process is not supported.
- If BOM creation through R30510 fails for any reason, the message is not relayed back to Agile PLM automatically.
- If the JDE E1 user decides to notify Agile PLM about “Rejection” of an ECO or any failures on BOM creation, the user has to manually update the appropriate status code on the ECO. The subsequent run of the Update ECO flow from JDE E1 to Agile PLM will relay the status message to Agile PLM.
- As these processes are synchronous, a process would start its execution only after the execution of the process before it completes.
- As Agile PLM maintains Effectivity Dates at the parent level, JDE E1 pushes the same date to all related component items.
- The BOMs which get loaded from JDE E1 as part of Initial Load could have a mixture of stock and non-stock items as components. The stock items on the BOM can be modified and sent successfully to JDE E1 via ECO. The only restriction is that the non-stock components cannot be changed.
- The maximum description length of ECOs and Items in JDE E1 is 30. If a user in Agile PLM sends a description which exceeds this length, characters exceeding this length will be truncated and the first 30 characters will be inserted / updated into JDE E1.

Configuration Parameters

In the AIA config file, a module level property DEFAULT_BOM_TYPE is set, which allows the user to specify the BOM type that needs to be passed to JDE E1 as this information does not flow in from Agile PLM. If this property is not set, for the forward flow from Agile PLM to JDE E1, JDE will pick up the default BOM type from the DD item’s default value.

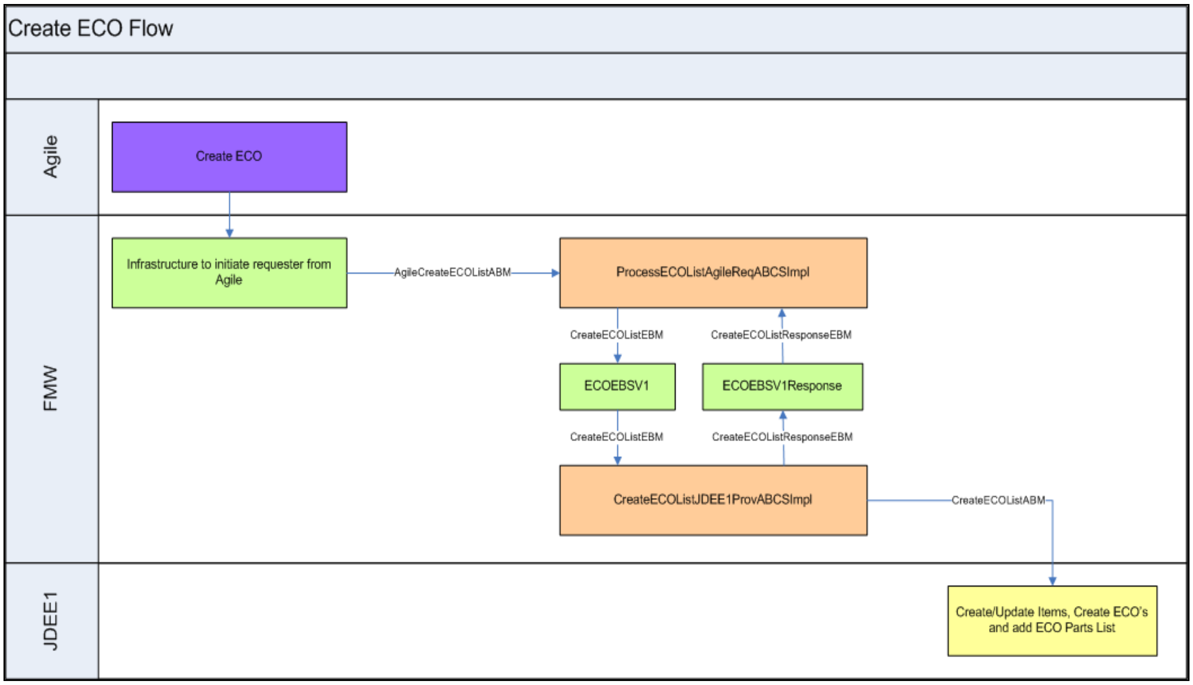
Note. This property is maintained at the module level as the same property is being used by BOM Initial Load.

UDC value of the “Implemented” state in JDE needs to go into the “ECO_Implemented_Status” config property. This status would be updated on the ECO which gets created for an ECO which does not hold any BOM information. Such ECOs would carry only the item(s) alone which are Created/Updated from Agile PLM.

Default_ECO_Branch_Plant needs to be set on the AIA config property to specify the Branch Plant under which the ECO needs to be created for facilitating non-stock items modification from Agile PLM. Ideally, Non-Stock items would not be created in Agile PLM. However, as Non-Stock items flow in from JDE E1 to Agile PLM as part of Initial Loads, any changes made to these items are brought back as ECOs from Agile PLM holding no Branch Plant information. Since the integration demands that the ECOs need to be created with Branch Plant information the default Branch Plant is used in this scenario.

Sequence

This sequence diagram shows the Create ECO flow from Agile PLM to JDE E1:



Create ECO Flow

Create ECO Flow

Here are the steps for Create ECO flow:

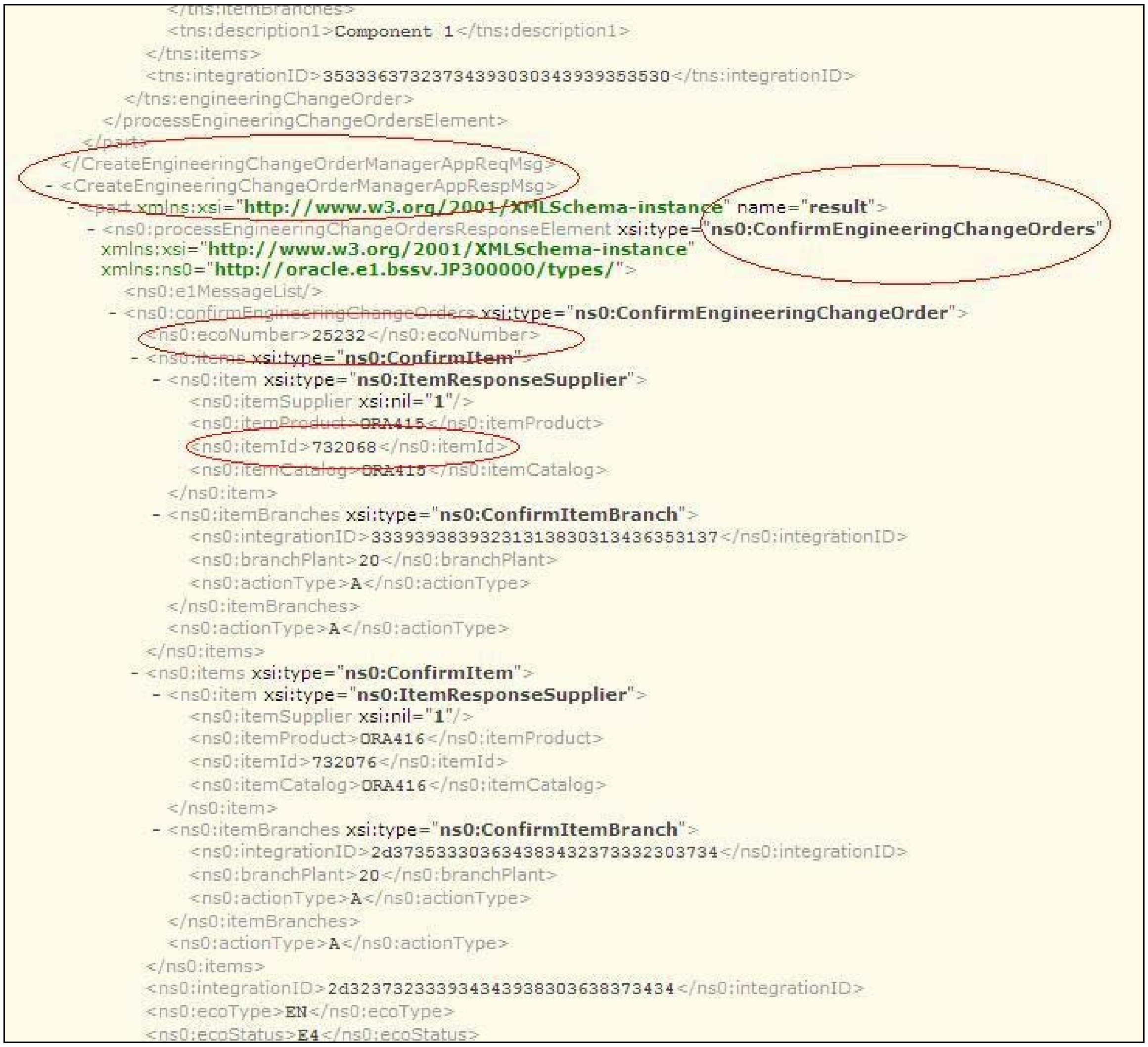
1. The ECO is “Submitted” for Approval (workflow step).
2. An ACS Workflow Event is generated when the ECO is “Released” from Agile PLM which happens after the approval of the ECO to trigger the ECO process flow.
3. The Queue framework captures the ACS payload (aXML) generated for the event and adds 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 EBS which routes the EBM to JDE E1 with ECO EBM as the input.
6. The EBS will invoke the Provider ABCS with EBM as input; provider then transforms EBM to JDE E1 ABM before invoking the EngineeringChangeOrderManager PBSSV (Web Service).
7. The ECO Business Flow -
   - Creates/Updates items in JDE E1 and links them to the Branch/Plants.
   - Creates an Engineering Change Order in JDE E1.
   - Associates list of revised items with new revisions and Effectivity Dates and schedules the ECO for implementation.

8. The status of Queue Message is updated in the Integration Queue, for monitoring.

For monitoring each ECO process:

1. On the BPEL Console, login with user name and password.
2. Click the Instances tab to view the successful ✓ and unsuccessful ⚠ runs.
3. Click Create EngineeringChangeOrderList.
4. On the next window that displays, click on the Flow tab.
5. Scroll down and click on InvokeEngineeringChangeOrderManager.

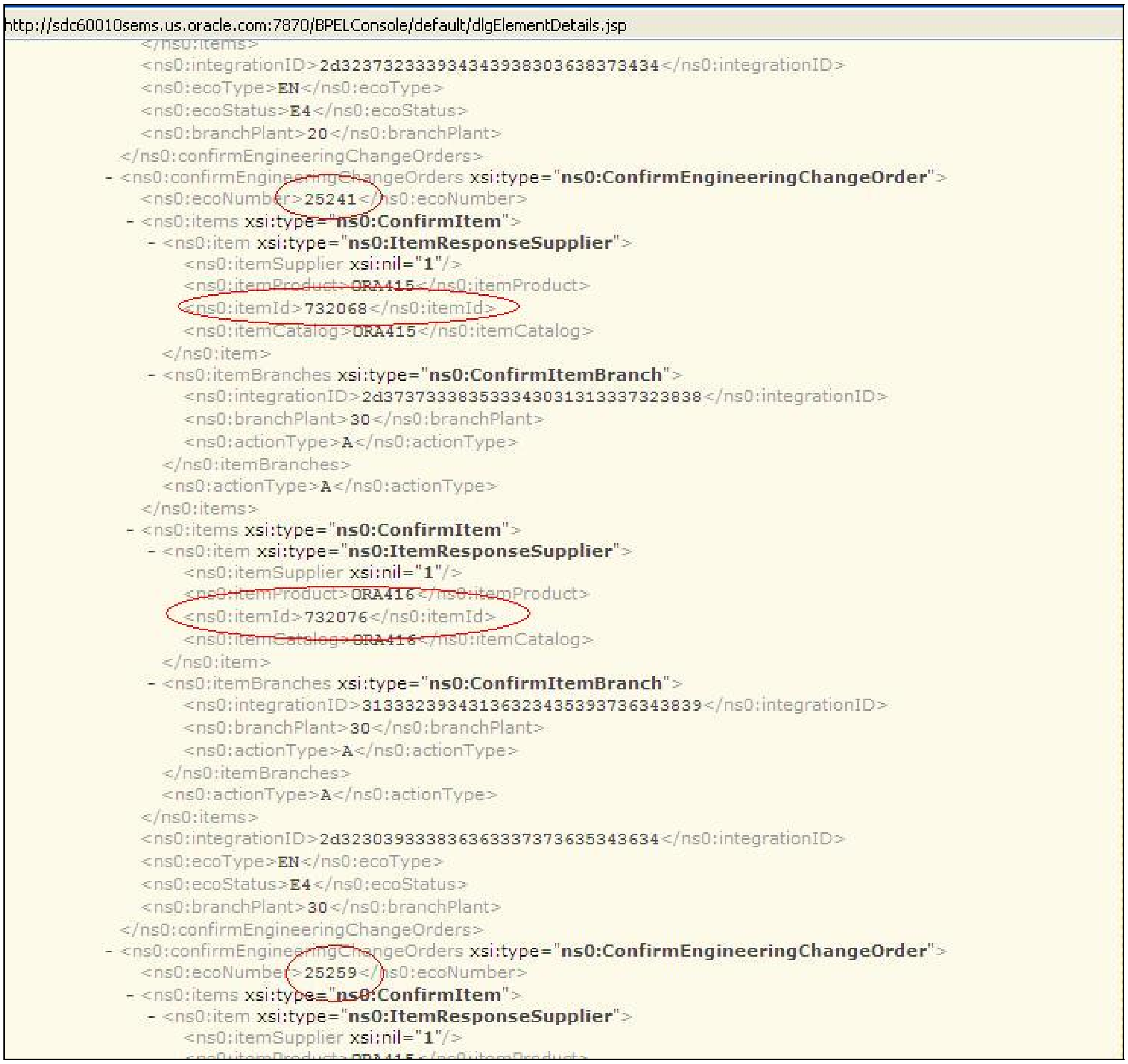
This brings up the Activity Audit Trail page. Scroll down to find the ECO number under the '*CreateECOManagerAppRespMsg*' tag > '*ConfirmEngineeringChangeOrders*'.



```
          </tns:itemBranches>
          <tns:description1>Component 1</tns:description1>
        </tns:items>
        <tns:integrationID>353336373237343930303439393353530</tns:integrationID>
      </tns:engineeringChangeOrder>
    </processEngineeringChangeOrdersElement>
  </part>
</CreateEngineeringChangeOrderManagerAppReqMsg>
- <CreateEngineeringChangeOrderManagerAppRespMsg>
  - <part xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="result">
    - <ns0:processEngineeringChangeOrdersResponseElement xsi:type="ns0:ConfirmEngineeringChangeOrders"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:ns0="http://oracle.e1.bssv.JP300000/types/">
        <ns0:e1MessageList/>
      - <ns0:confirmEngineeringChangeOrders xsi:type="ns0:ConfirmEngineeringChangeOrder">
          <ns0:ecoNumber>25232</ns0:ecoNumber>
        - <ns0:items xsi:type="ns0:ConfirmItem">
          - <ns0:item xsi:type="ns0:ItemResponseSupplier">
              <ns0:itemSupplier xsi:nil="1"/>
              <ns0:itemProduct>ORA415</ns0:itemProduct>
              <ns0:itemId>732068</ns0:itemId>
              <ns0:itemCatalog>ORA415</ns0:itemCatalog>
            </ns0:item>
          - <ns0:itemBranches xsi:type="ns0:ConfirmItemBranch">
              <ns0:integrationID>33393938393231313830313436353137</ns0:integrationID>
              <ns0:branchPlant>20</ns0:branchPlant>
              <ns0:actionType>A</ns0:actionType>
            </ns0:itemBranches>
            <ns0:actionType>A</ns0:actionType>
          </ns0:items>
        - <ns0:items xsi:type="ns0:ConfirmItem">
          - <ns0:item xsi:type="ns0:ItemResponseSupplier">
              <ns0:itemSupplier xsi:nil="1"/>
              <ns0:itemProduct>ORA416</ns0:itemProduct>
              <ns0:itemId>732076</ns0:itemId>
              <ns0:itemCatalog>ORA416</ns0:itemCatalog>
            </ns0:item>
          - <ns0:itemBranches xsi:type="ns0:ConfirmItemBranch">
              <ns0:integrationID>2d37353330363438343237332303734</ns0:integrationID>
              <ns0:branchPlant>20</ns0:branchPlant>
              <ns0:actionType>A</ns0:actionType>
            </ns0:itemBranches>
            <ns0:actionType>A</ns0:actionType>
          </ns0:items>
          <ns0:integrationID>2d32373233393434393830363873434</ns0:integrationID>
          <ns0:ecoType>EN</ns0:ecoType>
          <ns0:ecoStatus>E4</ns0:ecoStatus>
```

Activity Audit Trail page with ECO numbers 1

If multiple ECOs are created in this flow, all of them would be displayed here in order, one below the other.



http://sdc60010sems.us.oracle.com:7870/BPELConsole/default/dlgElementDetails.jsp

```
              </ns0:items>
              <ns0:integrationID>2d32373233393434393830363837343 4</ns0:integrationID>
              <ns0:ecoType>EN</ns0:ecoType>
              <ns0:ecoStatus>E4</ns0:ecoStatus>
              <ns0:branchPlant>20</ns0:branchPlant>
            </ns0:confirmEngineeringChangeOrders>
          - <ns0:confirmEngineeringChangeOrders xsi:type="ns0:ConfirmEngineeringChangeOrder">
              <ns0:ecoNumber>25241</ns0:ecoNumber>
            - <ns0:items xsi:type="ns0:ConfirmItem">
              - <ns0:item xsi:type="ns0:ItemResponseSupplier">
                  <ns0:itemSupplier xsi:nil="1"/>
                  <ns0:itemProduct>ORA415</ns0:itemProduct>
                  <ns0:itemId>732068</ns0:itemId>
                  <ns0:itemCatalog>ORA415</ns0:itemCatalog>
                </ns0:item>
              - <ns0:itemBranches xsi:type="ns0:ConfirmItemBranch">
                  <ns0:integrationID>2d373733383533343031313337323838</ns0:integrationID>
                  <ns0:branchPlant>30</ns0:branchPlant>
                  <ns0:actionType>A</ns0:actionType>
                </ns0:itemBranches>
                <ns0:actionType>A</ns0:actionType>
              </ns0:items>
            - <ns0:items xsi:type="ns0:ConfirmItem">
              - <ns0:item xsi:type="ns0:ItemResponseSupplier">
                  <ns0:itemSupplier xsi:nil="1"/>
                  <ns0:itemProduct>ORA416</ns0:itemProduct>
                  <ns0:itemId>732076</ns0:itemId>
                  <ns0:itemCatalog>ORA416</ns0:itemCatalog>
                </ns0:item>
              - <ns0:itemBranches xsi:type="ns0:ConfirmItemBranch">
                  <ns0:integrationID>31333239343136323435393736343839</ns0:integrationID>
                  <ns0:branchPlant>30</ns0:branchPlant>
                  <ns0:actionType>A</ns0:actionType>
                </ns0:itemBranches>
                <ns0:actionType>A</ns0:actionType>
              </ns0:items>
              <ns0:integrationID>2d32303933383636333737363535343634</ns0:integrationID>
              <ns0:ecoType>EN</ns0:ecoType>
              <ns0:ecoStatus>E4</ns0:ecoStatus>
              <ns0:branchPlant>30</ns0:branchPlant>
            </ns0:confirmEngineeringChangeOrders>
          - <ns0:confirmEngineeringChangeOrders xsi:type="ns0:ConfirmEngineeringChangeOrder">
              <ns0:ecoNumber>25259</ns0:ecoNumber>
            - <ns0:items xsi:type="ns0:ConfirmItem">
              - <ns0:item xsi:type="ns0:ItemResponseSupplier">
                  <ns0:itemSupplier xsi:nil="1"/>
                  <ns0:itemProduct>ORA415</ns0:itemProduct>
```

Activity Audit Trail page with ECO numbers 2

The Item Product and Item Catalog (2nd and 3rd Item Number) holds the part number that comes in from Agile PLM.

The Item ID (Short Item Number) is generated in JDE E1. This is the number given to the user for reference.

Create ECO Orchestration

Here is the flow to create ECO orchestration:

#	Activity	Remarks
1	Agile ACS transmits Agile Engineering Change Order Data in payload in the form of predefined XML format known as aXML. This file will get queued up for the further processing.	Agile ACS acts as a trigger for ECO Use case.
2	The QueueController Framework reads the highest priority Queue Message and transforms the payload (aXML) to AgileCreateEngineeringChangeOrderListABM.	*QueueController* processes the payload
3	QueueController invokes the ProcessEngineeringChangeOrderAgileReqABCS with AgileCreateEngineeringChangeOrderListABM as input.	
4	AgileCreateEngineeringChangeOrderListABM is transformed into CreateEngineeringChangeOrderListEBM.	*ProcessEngineeringChangeOrderAgileReqABCS* will make call backs to Agile Web services, if needed.
5	ProcessEngineeringChangeOrderAgileReqABCS invokes the CreateEngineeringChangeOrder operation on EngineeringChangeOrderEBS with CreateEngineeringChangeOrderEBM as input	
6	EngineeringChangeOrderEBS routes CreateEngineeringChangeOrderListEBM to CreateEngineeringChangeOrderListJDEE1ProvABCSImpl	
7	CreateEngineeringChangeOrderList JDEE1ProvABCSImpl transforms CreateEngineeringChangeOrderListEBM into EngineeringChangeOrderManagerABM and invokes the JDE E1 Web service with this input payload.	Creates/Updates items in JDE E1 and links them to the Branch/Plants, creates an ECO, associate revised items to it and creates BOM.
8	CreateEngineeringChangeOrderList JDEE1ProvABCSImpl invokes CreateEngineeringChangeOrderListResponse operation on EngineeringChangeOrderEBSResponse with CreateEngineeringChangeOrderListResponseEBM as input.	
9	The EngineeringChangeOrderEBS routes CreateEngineeringChangeOrderListResponseEBM to ProcessEngineeringChangeOrderAgileReqABCS	Response message routing
10	ProcessEngineeringChangeOrderAgileReqABCS sends the status back to the Queue Controller to update the queue.	This status is updated against the Queue message in the database by the QueueController
11	ProcessEngineeringChangeOrderAgileReqABCS transforms CreateEngineeringChangeOrderResponseEBM into AgileUpdateEngineeringChangeOrderListABM AgileUpdateEngineeringChangeOrderListABM is sent as	The web services update the status on the Change Order in Agile PLM which will be predefined P2 or P3 attributes on ECO object in Agile PLM.

#	Activity	Remarks
	an input to the Agile Web Service.	

Chapter 4: Integration Flow for ECO Implementation from JD Edwards EnterpriseOne to Agile PLM

This chapter provides an overview and discusses:

- Prerequisites
- AIA Core Components
- Integration Services
- Update ECO Solution Assumptions/Constraints

Overview

There is a use case in which Engineering Change Order information needs to flow from JDE E1 system to the PLM system. This case has restrictions in a multi-organization environment, simply because of the centralized (Agile PLM) to decentralized (JDE E1) nature of the interface.

Engineering Change Order gets completely implemented in the JD Edwards EnterpriseOne system. This step may be completed manually or set on the scheduler to update the Engineering Change Order in the Agile PLM system. Since an Engineering Change Order may be created in multiple branch/plants in JDE E1 and can have different status in each Branch Plant, this operation is carried out only when the Engineering Change Order is completely implemented in all the Branch Plants that it has been created in the JDE E1 system.

To transfer Change Order status information from JD Edwards EnterpriseOne to Agile PLM, an extract program (**R3013D**) in the JD Edwards EnterpriseOne system needs to be run. This program generates an XML file that contains the data that will be uploaded into the Agile PLM system.

R3013D will retrieve ECOs 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 given in the processing option.

This file is written 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, there are two methods that can be used to pass that data to the Agile PLM system:

- File Transfer Protocol (FTP)
- Oracle Application Server

If the FTP method is being used, the Change Order Implementation process requires an FTP adapter be configured to monitor the folders for newly created extract files. When the FTP adapter locates a new file; the appropriate enterprise service bus (ESB) process debatches the files into separate instances. Debatching is used to split large XML files into several smaller XML files.

Alternatively, the XML files can be moved to JDE E1 in the folder on the Oracle Application Server. If this method is used, the appropriate EBS process detects the file and debatches it into separate instances.

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

The *UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl* in turn performs the following actions:

- Transform ECO ABM to ECO EBM
- Invoke EngineeringChangeOrderEBS which does the routing to Provider ABCS – UpdateEngineeringChangeOrderListAgileProvABCSImpl

The Agile ABCS provides a Call back response to *EngineeringChangeOrderEBSResponse.*This response is received by *UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl* and this updates the IntegrationTimeStamp table with the last successful run date/time.

Note. If the FTP method is used, an FTP Server should be configured on the JD Edwards EnterpriseOne Enterprise Server. The FTP adapter should be configured to use the FTP connection that is set up on the Enterprise Server.

Prerequisites

Before running the Update ECO flow, the following needs to be completed:

1. Set up versions of each extract program in JD Edwards EnterpriseOne.
2. Set up Batch Processing information.

AIA Core Components

Core AIA Components for Update ECO

The Process Integration for Update ECO uses the following industry components:

EBOs	EngineeringChangeOrderEBO
EBMs	UpdateEngineeringChangeOrderEBM UpdateEngineeringChangeOrderResponseEBM
EBSs	EngineeringChangeOrderEBS

Core Components Locations

EBO & EBM XSD files	http://[HOST:PORT]/AIAComponents/EnterpriseObjectLibrary/Core/EBO/
WSDL files	http://[HOST:PORT]/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/

- For detailed documentation of individual EBOs, the **EBO Name** link needs to be selected on the *Integration Scenario Summary* page in the *Oracle AIA Console*. The *Integration Scenario Summary* page can also be used to search for and view integration scenarios that utilize a particular EBO or EBS.

For more information, see Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide, “Using the BSR,” Using the BSR UI to View Integration Scenarios.

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

  For more information, see Oracle Application Integration Architecture – Foundation Pack: Integration Developer’s Guide, “Extensibility for Oracle AIA Artifacts.”

Agile PLM & JD Edwards E1 Components for Update ECO

Services	JD Edwards EnterpriseOne (Requester)	Agile PLM (Provider)
ABMs	R3013D	AgileCreateItemABM AgileCreateItemResponseABM
ABCS	UpdateEngineeringChangeOrderListJDEE1ReqABCS	UpdateEngineeringCangeOrderListAgileProvABCSImpl
EBS	EngineeringChangeOrderEBS	EngineeringChangeOrderResponseEBS

Component Locations

ABO, ABM & Common XSD files	http://<servername>:<portname>/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas
WSDL files	http://<servername>:<portname>/AIAComponents/ApplicationObjectLibrary/Agile/V1/wsdls

Integration Services

These are the delivered services:

1. EngineeringChangeOrderEBS

EngineeringChangeOrderEBS is the Enterprise Business Service, which exposes the operations related to the EngineeringChangeOrder Integration on the EngineeringChangeOrder EBO.

The following are the routing rules:

EngineeringChangeOrderEBS ESB service

a. UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl

Route UpdateECOListEBM to UpdateEngineeringChangeOrderListAgileProvABCSImpl

EngineeringChangeOrderResponseEBS ESB service

b. UpdateEngineeringChangeOrderListAgileProvABCSImpl

Route UpdateECOListResponseEBM to UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl

2. UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl

UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl transforms the JDE E1 message (UpdateECOABM) into *UpdateECOListEBM* & calls the routing service to Update ECO and gets the *UpdateECOResponseEBM* response from the Agile PLM system. It then invokes the Integration Time Stamp Manager service to update the Integration Time Stamp table with the last run time and date.

Here is the flow for this integration service:

1. Picks up the xml file and based on the properties set in the file/ftp consumer.
2. Receives UpdateECOABM from file/ftp consumer.
3. Transforms JDE E1 specific UpdateECOABM into UpdateECOListEBM.
    - Populates EBM header.
4. Call 'ECOEBS' Service to send message UpdateECOListEBM to UpdateEngineeringChangeOrderAgileProvABCSImpl.
5. Receives UpdateECOListResponseEBM from ECOResponseEBS.
6. Calls the Integration Time Stamp Manager service to update the Integration Time Stamp table with the last run time and date.

UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl has the following transformation:

- Xform_EngineeringChangeOrderListABMReqMsg_To_EngineeringChangeOrderList ListEBMReqMsg

3. UpdateEngineeringChangeOrderListAgileProvABCSImpl

The UpdateEngineeringChangeOrderListAgileProvABCSImpl updates an Engineering Change Order in Agile PLM. In the Integration, the UpdateEngineeringChangeOrderListAgileProvABCSImpl is used to update the Status related fields. It is implemented as Asynchronous Process.

Here is the flow for this integration service:

1. UpdateEngineeringChangeOrderListAgileProvABCSImpl is invoked by EngineeringChangeOrderEBS with UpdateEngineeringChangeOrderListReqMsg which contains UpdateEngineeringChangeOrderListEBM as input.
2. Transform operation is called to convert the UpdateEngineeringChangeOrderListEBM into AgileUpdateEngineeringChangeOrderListABM.
3. AgileUpdateEngineeringChangeOrderListABM is passed as input to the web service operation which carries out the following functionalities for this integration.

4. Update Status related attributes on ECO (For Agile PLM to JD Edwards EnterpriseOne flow)
5. AgileUpdateEngineeringChangeOrderListResponseABM is received on successful execution of Coarse Grained API UpdateChange.
6. If the UpdateChange service operation fails on the Agile PLM side, a new Fault is generated and sent across with appropriate error message.

Update ECO Solution Assumptions/Constraints

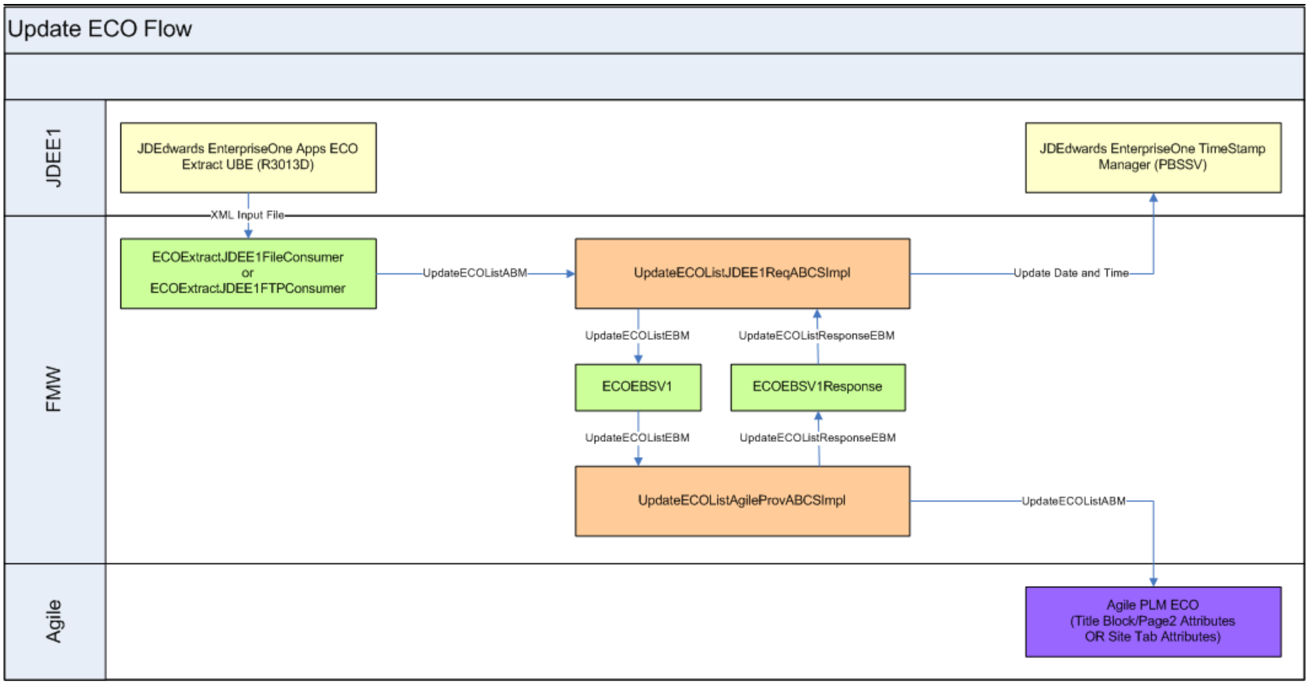
This design assumes that the following statements are true:

- The ECO update flow is from JD Edwards EnterpriseOne to Agile PLM only**.** Updates to ECOs in Agile are not supported.
- Sending ECO Parts List and Related Items information from JDE E1 is not supported in this integration.
- The JDE E1 IntegrationTimeStampManager Business service will update the last batch extract date/time as the Enterprise Server Date/Time.
- Any status change on the ECO in JDE E1 will be updated back to Agile PLM. When Agile PLM is configured for Multi-Site (and when multiple sites are attached to an ECO), a single ECO in Agile PLM creates multiple pre-approved ECOs in JDE E1. 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, i.e.; if any of the ECOs fails to get implemented, the user should ensure that the *failed* ECOs get implemented before the ‘Update ECO’ flow is triggered, because Agile PLM will consider the ECO at it’s end as implemented if any one of the corresponding ECOs at the JDE E1 end goes into an *'Implemented'* state.

  Alternatively, if the user wants to relay the failure status back to Agile PLM, the user has to manually configure the status on **P48020** to reflect the failure. This can be done by creating a new Status UDC Item for the error and configuring the ECO_STATUS_CODE DVM; then the changes can be relayed during the subsequent run of the ‘Update ECO’ flow. The failure status will be reflected only on the 'Page 2' Status Flex Field (this needs to be configured) while the workflow status for the ECO may still appear as *'Implemented'*.
- Any Effectivity Date changes in JDE E1 will not be sent back to Agile PLM.

Sequence

This diagram shows the Update ECO flow from JDE E1 to Agile PLM:


Update ECO Flow
JDEE1
FMW
Agile
JDEdwards EnterpriseOne Apps ECO Extract UBE (R3013D)
JDEdwards EnterpriseOne TimeStamp Manager (PBSSV)
XML Input File
ECOExtractJDEE1FileConsumer or ECOExtractJDEE1FTPConsumer
UpdateECOListABM
UpdateECOListJDEE1ReqABCSImpl
Update Date and Time
UpdateECOListEBM
UpdateECOListResponseEBM
ECOEBSV1
ECOEBSV1Response
UpdateECOListEBM
UpdateECOListResponseEBM
UpdateECOListAgileProvABCSImpl
UpdateECOListABM
Agile PLM ECO (Title Block/Page2 Attributes OR Site Tab Attributes)

Update ECO flow from JDE E1 to Agile PLM

ECO Update Flow

The Engineering Change Order Update Process flows as follows:

1. The requester ABCS, defined as an “asynchronous” process, receives a list of ABMs from the JDE E1 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, whose IDs are present in the XREF tables in the Fusion Middleware (FMW) layer. This provides a list of change orders that were actually from the Agile PLM system alone. The original list of ABMs may be those that are from non-Agile PLM sources as well.
3. The BPEL process then makes an ESB 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 EngineeringChangeOrderEBS with the EngineeringChangeOrderListEBM. This will route to the appropriate provider.
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.

   There is no impact on performance since it is a scheduled call and is invoked from back-end and not a UI invoke.

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

Change Status Operation

The Change Status operation in Agile PLM for Update ECO flow from JDE E1 to Agile PLM updates the status of the workflow when JDE E1 passes the “Implemented” status to Agile PLM.

The Update ECO Provider Service in Agile PLM system takes care of the following:

1. Whenever there is change in the status change on any ECO in JDE E1, JDE E1 sends the ECO status to the Agile PLM system.
2. *UpdateEngineeringChangeOrderListAgileProvImpl* Service checks the status of the ECO which is sent by JDE E1 and updates the *Status* field on the ECO in the Agile PLM system.
3. If the ECO which comes in from JDE E1 have the status as 'Implemented', the change in the Agile PLM system for the ECO needs to be pushed to 'Implemented' status, besides updating the Status field on ECO in the Agile PLM system.
4. For this, the entry in the *CHANGE_STATUS* table is used. If there is no entry corresponding to event 'Change Implementation' and Change type same as change type of ECO in Agile PLM like ECO or SCO, only then the change is pushed to next status otherwise the change won't be pushed to next status at all.
5. The *CHANGE_STATUS* table is located in the AIA Schema created on the SOA Server. The default User ID/ Password are plmpip / plmpip. The DB details, such as URL, port, SID, etc. can be found in the properties file - *AIA_HOME/config/deploy.properties*.

CHANGE_STATUS Sample data

EVENT	OUTCOME	SUBCLASS	WORKFLOW	NEXT_STATUS
Change Implemented	SUCCESS	ECO	Default Change Orders	Implemented

- EVENT column key is 'Change Implemented' and 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. For example, when a Change is implemented (identified by event - this is a key. do not change it) and outcome is SUCCESS (identified by outcome, gets checked while changing status here) and workflow chosen in Default Change Orders the set up is asking the change to be moved to Implemented status.

Example:

1. Release an ECO, C0001, from Agile PLM to JDE E1.
2. In JDE E1, create this ECO in three Branch Plants; say BP10, BP20, and BP30 using the New Part/Product Release process.
3. Change the status of this ECO in 10 to 'Implemented' within the JD Edwards EnterpriseOne system.

4. JDE E1 pushes this implemented ECO status to Agile PLM.
5. When it is received on 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 JDE E1, move the ECO in Agile PLM to 'Implemented' status.
   - The status field in Agile PLM is also updated with the “Implemented” status.
6. If there is an entry in the CHANGE_STATUS table corresponding to the event 'Change Implemented' & SubClass 'ECO', and workflow mentioned on ECO C0001 in Agile PLM, read the next status and push it to next status mentioned.
7. If there is no entry in table for the data mentioned above, no attempt will be made to push ECO C0001 to next status.

ECO Update Orchestration

This table describes the activities involved in ECO Update Orchestration process flow:

#	Activity	Remarks
1.	Run R3013D to export ECOs	This activity takes place when ECO related information is updated in JDE
2.	EngineeringChangeOrderListExtractJDEE1Consumer ESB reads/debatches file	EngineeringChangeOrderListExtractJDEE1Consumer is triggered.
3.	Invoke UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl process	*UpdateEngineeringChangeOrderEBM* is created inside *UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl*.
4.	UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl invokes the EngineeringChangeOrderEBS with UpdateEngineeringChangeOrderList operation	An invoke activity in *UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl* invokes the *UpdateEngineeringChangeOrderList* operation on *EngineeringChangeOrderEBS* with UpdateEngineeringChangeOrderListEBM as the input.
5.	EngineeringChangeOrderEBS routes UpdateEngineeringChangeOrderListEBM to UpdateEngineeringChangeOrderListAgileProvABCSImpl	*EngineeringChangeOrderEBS* routes *UpdateEngineeringChangeOrderListEBM* to *UpdateEngineeringChangeOrderListAgileProvABCSImpl.*
6.	UpdateEngineeringChangeOrderListAgileProvABCSImpl does the transformation	*UpdateEngineeringChangeOrderListAgileProvABCSImpl* transforms *UpdateEngineeringChangeOrderListEBM* into *AgileUpdateEngineeringChangeOrderListABM.*
7.	UpdateEngineeringChangeOrderListAgileProvABCSImpl invokes Agile Web Services	Agile Web Services are invoked with AgileUpdateEngineeringChangeOrderListABM as input.The status of ECO is updated in Agile PLM. AgileUpdateEngineeringChangeOrderListResponseABM is sent back to the UpdateEngineeringCUpdateEngineeringChangeOrderListAgileProvABCSImpl.

Chapter 5: Integration Flow for Item Attributes & Cost Updates from JD Edwards EnterpriseOne to Agile PLM

This chapter provides an overview and discusses:

- Prerequisites
- AIA Core Components
- Integration Services
- Update Item Attributes Solution Assumptions/Constraints

Overview

The update of Item Attribute & Unit Cost information from JD Edwards EnterpriseOne to Agile PLM is completed as a Batch Process. A new Batch Program (UBE) will extract the list of items (including obsolete items if not filtered in data selection) whose attribute values and unit cost need to be updated in Agile PLM. Obsolete items shall be marked as obsolete in Agile's Item Life cycle field.

Item Data Extract UBE **R4102D** will be executed on a scheduled basis. This program generates an XML file that contains the data of items along with values for all the attributes and unit cost information which will get updated into the Agile PLM system.

Since it is an update to Agile PLM, the items in the XML input file should already exist in Agile PLM; this is determined by checking the XREF record. If an XREF record exists for the item, Agile PLM will update the corresponding item record with Item Attribute information from JD Edwards EnterpriseOne. If an XREF record does not exist, then it will not be updated to Agile PLM (it is skipped).

The extract will include basic and additional item information, Unit of Measure (UOMs), sales, purchasing classifications; inventory processing information and so forth, from Item Master **(F4101)**, Item Branch **(F4102)** along with Unit Cost information **(F4105)**.

Item cost information is a one-way flow from JDE E1 to Agile PLM. Agile PLM uses this data strictly for informational purposes to help with product design. Cost should not be changed in Agile PLM, but if it is, it will not be sent back to JDE E1. **R4102D** will retrieve Items that have 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 given in the Processing Option.

This file is written 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, there are two methods that can be used to pass that data to the Agile PLM system:

- FTP

- Oracle Application Server

If the FTP method is being used, the Update Item Attributes and Unit Cost process require an FTP adapter be configured to monitor the folders for newly created extract files. When the

FTP adapter locates a new file; the appropriate Enterprise Service Bus (ESB) process debatches the files into separate instances. Debatching is used to split large XML files into several smaller XML files.

Alternatively, the XML files can be moved to JDE E1 in the folder on the Oracle Application Server. If this method is used, the appropriate EBS process detects the file and debatches it into separate instances.

After the XML files are debatched, routing services route each individual XML file to UpdateItemListJDEE1ReqABCSImpl*.*

This in turn performs the following actions:

- Transform Item ABM to Item EBM
- Invoke ItemEBS which does the routing to Provider ABCS – UpdateItemListAgileProvABCSImpl

The Agile ABCS provides a Call back response to *ItemEBSResponse.* This response is received by *UpdateItemListJDEE1ReqABCSImpl* and this updates the IntegrationTimeStamp table with the last successful run date/time.

Note. If the FTP method is used, an FTP Server should be configured on the JD Edwards EnterpriseOne Enterprise Server. The FTP adapter should be configured to use the FTP connection that is set up on the Enterprise Server.

Prerequisites

Before running the Update Item flow, the following needs to be completed:

1. Set up versions of each extract program in JD Edwards EnterpriseOne.
2. Set up Batch Processing information.

AIA Core Components

Core AIA Components for Update Item Attributes

The Process Integration for Update Item Attributes uses the following industry components:

EBOs	ItemEBO
EBMs	UpdateItemEBM UpdateItemResponseEBM
EBSs	ItemEBS

Core Components Locations

EBO & EBM XSD files	http://[HOST:PORT]/AIAComponents/EnterpriseObjectLibrary/Core/EBO/
WSDL files	http://[HOST:PORT]/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/

- For detailed documentation of individual EBOs, the **EBO Name** link needs to be selected on the *Integration Scenario Summary* page in the *Oracle AIA Console*. The *Integration Scenario Summary* page can also be used to search for and view integration scenarios that utilize a particular EBO or EBS.

For more information, see Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide, “Using the BSR,” Using the BSR UI to View Integration Scenarios.

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

For more information, see Oracle Application Integration Architecture – Foundation Pack: Integration Developer’s Guide, “Extensibility for Oracle AIA Artifacts.”

Agile PLM & JD Edwards E1 Components for Update Item Attributes

Services	JD Edwards EnterpriseOne (Requester)	Agile PLM (Provider)
ABMs	R4102D	AgileCreateItemABM AgileCreateItemResponseABM
ABCS	UpdateItemListJDEE1ReqABCS	UpdateItemListAgileProvABCSImpl
EBS	ItemEBS	ItemResponseEBS

Component Locations

ABO, ABM & Common XSD files	http://<servername>:<portname>/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas
WSDL files	http://<servername>:<portname>/AIAComponents/ApplicationObjectLibrary/Agile/V1/wsdls

Integration Services

These are the delivered services:

1. ItemEBS

ItemEBS is the Enterprise Business Service, which exposes the operations related to the Item Integration on the Item EBO. The following are the routing rules:

ItemEBS ESB service

1. UpdateItemListJDEE1ReqABCSImpl

Route UpdateItemListEBM to UpdateItemListAgileProvABCSImpl

ItemResponseEBS ESB service

2. UpdateItemListAgileProvABCSImpl

Route UpdateItemResponseEBM to UpdateItemJDEE1ReqABCSImpl

2. UpdateItemListJDEE1ReqABCSImpl

UpdateItemListJDEE1ReqABCSImpl transforms the JDE E1 message (UpdateItemListABM) into *UpdateItemListEBM* & calls the routing service to Update Item List Attributes and Item Cost Attributes. It then gets the *UpdateItemListResponseEBM* response from the Agile PLM system and invokes the Integration Time Stamp Manager service to update the Integration Time Stamp table with the last run time and date.

Here is the flow for this integration service:

1. Picks up the xml file based on the properties set in the file/ftp consumer.
2. Receives UpdateItemListABM from file/ftp consumer.
3. Transforms JDE E1 specific UpdateItemListABM into UpdateItemListEBM.
   - Populates EBM header.
4. Call 'ItemEBS' Service to send message UpdateItemListEBM to UpdateItemListAgileProvABCSImpl.
5. Receives UpdateItemListResponseEBM from ItemResponseEBS.
6. Calls the Integration Time Stamp Manager service to update the Integration Time Stamp table with the last run time and date.

   UpdateItemListJDEE1ReqABCSImpl has the following transformation:

   - Xform_ItemListABMReqMsg_To_ItemListEBMReqMsg

3. UpdateItemListAgileProvABCSImpl

UpdateItemListAgileProvABCSImpl is used to facilitate the communication between ItemEBS and Agile Web Service used for updating the Items' cost information in a batch mode in the Agile PLM system.

- Receives UpdateItemListReqMsg that contains UpdateItemListEBM
- Transform operation is called to convert the UpdateItemListEBM into AgileUpdateItemListABM.
- AgileUpdateItemListABM is sent as input to the Web Service operation UpdateItems (Coarse Grained APIs on Agile PLM side) to update Items in the Agile PLM system.
- AgileUpdateItemListResponseABM is received on successful execution of Coarse Grained API.
- Transform operation is called to convert the AgileUpdateItemListABM to UpdateItemListResponseEBM, which is returned as an output of this BEPL process.
- If the UpdateItems service operation fails on the Agile PLM side, a new Fault is generated and sent across with the appropriate error message.

Update Item Attributes Solution Assumptions/ Constraints

This design assumes that the following statements are true:

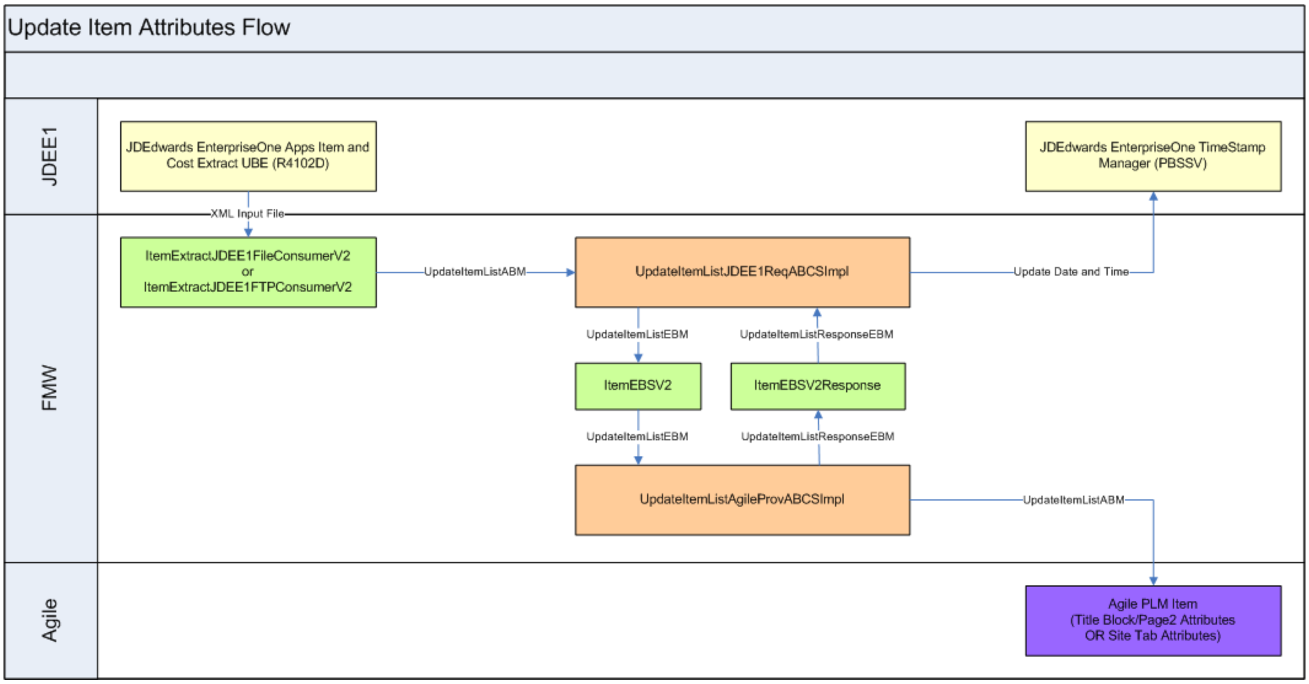
- If item information needs to be retrieved from multiple Branch/Plants in JDE E1, the following constraints should be met in order to support this process:
    - Agile PLM Multi-Sites is configured
    - There is a one-to-one mapping between Agile PLM Sites and JDE E1 Branch/Plants. This will be accomplished through DVMs.
    - The attribute(s) to be updated with JDE E1 data are on the Sites tab of the item if Multi-Site is configured.
- It is not necessary to have Agile PLM Multi-Sites active. Any Title Block or Page Two attribute will be updated with JDE E1 data.
- There can be multiple Item cost records retrieved for any Item when Item Cost level is 3. In such case, the UBE extracts the first record to XML and suppresses the processing of subsequent records.
- If Currency processing is enabled in JDE E1 and if the unit cost is not zero then the currency code will be sent back to Agile PLM.
- If the cost is maintained at Item Level then the Default Company Currency code will be sent back to Agile PLM.
- The JDE E1 IntegrationTimeStampManager Business service will update the last batch

extract date/time as the Enterprise Server Date/Time.

Note. New Item creation in JDE E1 is not part of the scope for this activity. It is assumed that the item exists in both Agile PLM and JDE E1, by means of a prior New Part Introduction or independent Initial Load processes.

Sequence

This diagram shows the Update Item Attribute flow:


Update Item Attributes Flow
JDEE1
FMW
Agile
JDEdwards EnterpriseOne Apps Item and Cost Extract UBE (R4102D)
JDEdwards EnterpriseOne TimeStamp Manager (PBSSV)
XML Input File
ItemExtractJDEE1FileConsumerV2 or ItemExtractJDEE1FTPConsumerV2
UpdateItemListABM
UpdateItemListJDEE1ReqABCSImpl
Update Date and Time
UpdateItemListEBM
UpdateItemListResponseEBM
ItemEBSV2
ItemEBSV2Response
UpdateItemListEBM
UpdateItemListResponseEBM
UpdateItemListAgileProvABCSImpl
UpdateItemListABM
Agile PLM Item (Title Block/Page2 Attributes OR Site Tab Attributes)

Update Item Attribute Flow

Update Item Attributes Flow

To extract the Updated Item Attribute & Unit Cost information in JD Edwards EnterpriseOne system, the user should run the Item and Cost Extract batch program (**R4102D**). **R4102D** will retrieve 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 given in the processing option.

This program retrieves these Items and creates an XML file, and if it completes successfully, a PDF is created 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. The user should specify the location of the folder in the processing options of the **R4102D** program.

After the files have been written to the Enterprise Server, one of these consumer services retrieves and debatches the XML files:

- ItemListExtractJDEE1FTPConsumer

This consumer service is used when you configure the FTP adapter to locate files.

- ItemListExtractJDEE1FileConsumer

This consumer service is used when you move your files to the JDE E1In file on the Oracle Application 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 individual XML file to UpdateItemListJDEE1ReqABCSImpl. This BPEL service will transform UpdateItemListABM to UpdateItemListEBM. As part of this transformation, if the BPEL service finds any Items which are created in JDE they would be dropped by looking up against the cross reference table.

It then makes an asynchronous request-delayed response call to ItemEBS with UpdateItemListEBM. This will be routed to UpdateItemListAgileProvABCSImpl.

UpdateItemListAgileProvABCSImpl updates the Item in Agile and sends a response UpdateItemListResponseEBM.

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

Update Item Attributes Orchestration

This table describes the activities involved in the Update Item Attributes Orchestration process flow:

#	Activity	Remarks
1	Run R4102D to export items	This activity takes place when Item related information is updated in JDE
2	ItemListExtractJDEE1Consumer ESB reads/debatches file	ItemListExtractJDEE1Consumer is triggered.
3	JDE E1 Item Operational Attribute Update Action Trigger	JDE E1 Item Operational Attribute Update Action trigger Invokes the *UpdateItemListJDEE1ReqABCSImpl*
4	UpdateItemListJDEE1ReqABCSImpl invokes ItemEBS	An invoke activity in *UpdateItemListJDEE1ReqABCSImpl* invokes the *UpdateItemList* operation on *ItemEBS* with *UpdateItemListEBM* as input
5	ItemEBS routes the UpdateItemListEBM to UpdateItemListAgileProvABCSImpl	*ItemEBS* routes *UpdateItemListEBM* as input to *UpdateItemListAgileProvABCSImpl*
6	UpdateItemListAgileProvABCSImpl invokes Agile's Item Operational Attribute Update Web Service	*ItemEBS* routes *UpdateItemListEBM* as input to *UpdateItemListAgileProvABCSImpl*. *UpdateItemListAgileProvABCSImpl* transforms the *UpdateItemListEBM* to *AgileUpdateItemListABM* and invokes *UpdateItem* service operation on Agile Web Service to update Item cost related attribute information for from Oracle to Agile PLM. The *AgileUpdateItemListResponseABM* is returned

#	Activity	Remarks
		back to *UpdateItemListAgileProvABCSImpl*
7	UpdateItemListAgileProvABCSImpl sends response to ItemResponseEBS	*UpdateItemListAgileProvABCSImpl* transforms the *AgileUpdateItemListResponseABM* to *UpdateItemListListResponseEBM and sends response to ItemResponseEBS*
8	ItemResponseEBS sends the UpdateItemListListResponseEBM to UpdateItemListJDEE1ReqABCSImpl	*ItemResponseEBS* sends the *UpdateItemListListResponseEBM* to *UpdateItemListJDEE1ReqABCSImpl*

Chapter 6: Integration Flow for Item Balance Updates from JD Edwards EnterpriseOne to Agile PLM

This chapter provides an overview and discusses:

- Process flow for item balance updates
- Prerequisites
- Solution assumptions and constraints
- Core AIA components
- Integration components

Overview

The update of Item Balance information from JD Edwards EnterpriseOne to Agile PLM is completed as a Batch Process. A new Batch Program (UBE) will extract the list of items whose quantity values need to be updated in Agile PLM.

Item Balance Data Extract UBE **R41021D** will be executed on a scheduled basis. This program generates an XML file that contains records which include Calculated Quantity information for items summarized at the Branch Plant level, based on the information from the Item Location File and User Defined Availability Constants which will get updated into the Agile PLM system.

Since, it is an update to Agile PLM, the items in the XML input file should already exist in Agile PLM; this is determined by checking the XREF record. If an XREF record exists for the item, Agile PLM will update the corresponding item record with Item Balance information from JD Edwards EnterpriseOne. If an XREF record does not exist, then it will not be updated to Agile PLM (it is skipped).

Item balance information is a one-way flow from JDE E1 to Agile PLM. Agile PLM uses this data strictly for informational purposes to help with product design. These values should not be changed in Agile PLM but if it is, it will not be sent back to JDE E1.

R41021D is used to retrieve those list of Items, whose Availability information has changed since the last successful run (the last successful run date and time is stored in an IntegrationTimeStampTable - **F0095** for this UBE) or As of Date given in the Processing Option for the purpose of updating Item Availability information from JD Edwards EnterpriseOne to Agile PLM.

This file is written 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 the later sections of this chapter.

After creating the XML files, there are two methods that can be used to pass that data to the Agile PLM system:

- FTP

- Oracle Application Server

If the FTP method is being used, the Update Item Balance process requires an FTP adapter to be configured to monitor the folders for newly created extract files. When the FTP adapter locates a new file, the appropriate Enterprise Service Bus (ESB) process debatches the files into separate instances. Debatching is used to split large XML files into several smaller XML files.

Alternatively, the XML files can be moved to JDE E1, in the folder on the Oracle Application Server. If this method is used, the appropriate EBS process detects the file and debatches it into separate instances.

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

See [Chapter 7 - Setting up Batch Processing Information].

The *UpdateItemBalanceListJDEE1ReqABCSImpl* in turn performs the following actions:

- Transforms Item Balance ABM to Item Balance EBM
- Invokes ItemBalanceEBS which does the routing to Provider ABCS – UpdateItemBalanceListAgileProvABCSImpl

The Agile ABCS provides a Call back response to *ItemBalanceEBSResponse.* This response is received by *UpdateItemBalanceListJDEE1ReqABCSImpl* and it updates the IntegrationTimeStamp table with the last successful run date/time.

Note. If the FTP method is used, an FTP Server should be configured on the JD Edwards EnterpriseOne Enterprise Server. The FTP adapter should be configured to use the FTP connection that is set up on the Enterprise Server.

Prerequisites

Before running the Update Item Balance flow, the following needs to be completed:

1. Set up versions of each extract program in JD Edwards EnterpriseOne.
2. Set up Batch Processing information.

AIA Core Components

Core AIA Components for Update Item Balance

The Process Integration for Create ECO uses the following industry components:

EBOs	ItemBalanceEBO
EBMs	UpdateItemBalanceEBM UpdateItemBalanceResponseEBM

EBSs	ItemBalanceEBS

Core Components Locations

EBO & EBM XSD files	http://[HOST:PORT]/AIAComponents/EnterpriseObjectLibrary/Core/EBO/
WSDL files	http://[HOST:PORT]/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/

- For detailed documentation of individual EBOs, the **EBO Name** link needs to be selected on the *Integration Scenario Summary* page in the *Oracle AIA Console*. The *Integration Scenario Summary* page can also be used to search for and view integration scenarios that utilize a particular EBO or EBS.

  For more information, see Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide, “Using the BSR,” Using the BSR UI to View Integration Scenarios.

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

  For more information, see Oracle Application Integration Architecture – Foundation Pack: Integration Developer’s Guide, “Extensibility for Oracle AIA Artifacts.”

Agile PLM & JD Edwards E1 Components for Update Item Balance

Services	JD Edwards EnterpriseOne (Requester)	Agile PLM (Provider)
ABMs	R41021D	AgileCreateItemABM AgileCreateItemResponseABM
ABCS	UpdateItemBalanceListJDEE1ReqABCS	UpdateItemBalanceListAgileProvABCSImpl
EBS	ItemBalanceEBS	ItemBalanceResponseEBS

Component Locations

ABO, ABM & Common XSD files	http://<servername>:<portname>/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas
WSDL files	http://<servername>:<portname>/AIAComponents/ApplicationObjectLibrary/Agile/V1/wsdls

Integration Services

These are the delivered services:

1. ItemBalanceEBS

ItemBalanceEBS is the Enterprise Business Service, which exposes the operations related to the Item Integration on the Item EBO. The following are the routing rules:

ItemBalanceEBS ESB service

1. UpdateItemBalanceListJDEE1ReqABCSImpl

   Route UpdateItemBalanceListEBM to UpdateItemBalanceListAgileProvABCSImpl

ItemBalanceResponseEBS ESB service

2. UpdateItemBalanceListAgileProvABCSImpl

   Route UpdateItemBalanceListResponseEBM to UpdateItemBalanceListJDEE1ReqABCSImpl

2. UpdateItemBalanceListJDEE1ReqABCSImpl

UpdateItemBalanceListJDEE1ReqABCSImpl transforms the JDE E1 message (UpdateItemBalanceListABM) into *UpdateItemBalanceListEBM* & calls the routing service to Update Item Balance List Attributes. It then gets the *UpdateItemBalanceListResponseEBM* response from the Agile PLM system and invokes the Integration Time Stamp Manager service to update the Integration Time Stamp table with the last run time and date.

Here is the flow for this integration service:

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 JDE E1 specific UpdateItemBalanceListABM into UpdateItemBalanceListEBM.
   - Populates EBM header.
4. Call 'ItemBalanceEBS' Service to send message UpdateItemBalanceListEBM to UpdateItemBalanceListAgileProvABCSImpl.
5. Receives UpdateItemBalanceListResponseEBM from ItemResponseEBS.
6. Calls the Integration Time Stamp Manager service to update the Integration Time Stamp table with the last run time and date.

   UpdateItemBalanceListJDEE1ReqABCSImpl has the following transformation:

   - Xform_ItemBalanceListABMReqMsg_To_ItemBalanceListEBMReqMsg

3. UpdateItemBalanceListAgileProvABCSImpl

UpdateItemBalanceListAgileProvABCSImpl is used to facilitate the communication between ItemBalanceEBS and Agile web service used for updating the Items' Quantity information in batch mode in Agile PLM.

Here is the flow for this integration service:

1. Receives UpdateItemBalanceListReqMsg, which contains UpdateItemBalanceListEBM
2. Transform operation is called to convert the UpdateItemBalanceListEBM into AgileUpdateItemBalanceListABM.
3. AgileUpdateItemBalanceListABM is passed as input to the web service operation UpdateItems (Coarse Grained APIs on Agile PLM side) to update Items in Agile PLM.
4. AgileUpdateItemBalanceListResponseABM is received on successful execution of Coarse Grained API.
5. Transforms AgileUpdateItemBalanceListResponseABM to UpdateItemBalanceListResponseEBM, which is returned as output of this BEPL process.

   If the UpdateItems service operation fails on the Agile PLM side, a new Fault is generated and the same will be sent across with an appropriate error message.

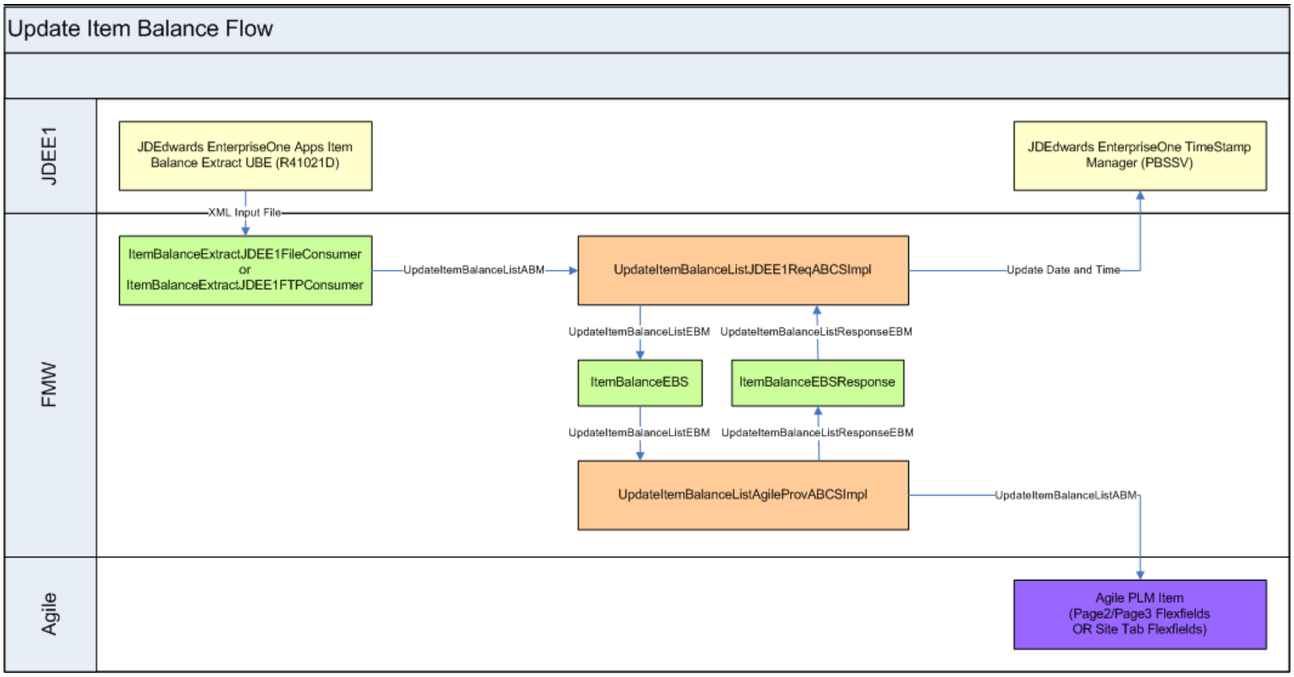
Update Item Balance Solution Assumptions/ Constraints

This design assumes that the following statements are true:

1. If item balance information needs to be retrieved from multiple Branch/Plants in JDE E1, then the Agile PLM Multi-Sites must be configured.
2. There is a one-to-one mapping between Agile PLM Sites and JDE E1 branch/plants
3. The attribute(s) to be updated with JDE E1 data are on the Sites tab of the item
4. It is not necessary to have Agile PLM Multi-Sites. Any Title Block or Page Two attribute may be updated with JDE E1 data.
5. The JDE E1 IntegrationTimeStampManager Business service will update the last batch extract date/time as the Enterprise Server Date/Time.

Sequence

This diagram shows Update Item Balance flow:


Update Item Balance Flow
JDEE1
FMW
Agile
JDEdwards EnterpriseOne Apps Item Balance Extract UBE (R41021D)
XML Input File
ItemBalanceExtractJDEE1FileConsumer or ItemBalanceExtractJDEE1FTPConsumer
UpdateItemBalanceListABM
UpdateItemBalanceListJDEE1ReqABCSImpl
Update Date and Time
JDEdwards EnterpriseOne TimeStamp Manager (PBSSV)
UpdateItemBalanceListEBM
UpdateItemBalanceListResponseEBM
ItemBalanceEBS
ItemBalanceEBSResponse
UpdateItemBalanceListEBM
UpdateItemBalanceListResponseEBM
UpdateItemBalanceListAgileProvABCSImpl
UpdateItemBalanceListABM
Agile PLM Item (Page2/Page3 Flexfields OR Site Tab Flexfields)

Update Item Balance Flow

Item Balance Update Orchestration

This table describes the activities involved in the Item Balance Update Orchestration flow:

#	Activity	Remarks
1	Run R41021D to export items	This activity takes place when Item Balance related information is updated in JDE
2	ItemBalanceExtractJDEE1Consumer ESB reads/debatches file	ItemBalanceExtractJDEE1Consumer is triggered.
3	Oracle Item Balance Action triggers	Oracle Item Balance Action Invokes the *UpdateItemBalanceListJDEE1ReqABCS*
4	UpdateItemBalanceListJDEE1ReqABCSImpl invokes ItemBalanceEBS	An invoke activity in *UpdateItemBalanceListJDEE1ReqABCSImpl* invokes the *UpdateItemBalanceList* operation on *ItemBalanceEBS* with *UpdateItemBalanceList* as input
5	ItemBalanceEBS invokes the UpdateItemBalanceListAgileProvABCSImpl	*ItemBalanceEBS* sends *SyncItemBalanceListEBM* as input to *UpdateItemBalanceListAgileProvABCSImpl*
6	UpdateItemBalanceListAgileProvABCSImpl invokes Agile's Item Balance Web Service	*UpdateItemBalanceListAgileProvABCSImpl* transforms the *UpdateItemBalanceListEBM* to *AgileUpdateItemBalanceListABM* and invokes the updateItem service operation on Agile Web Service to update Item On-Hand Quantity information from Oracle to Agile PLM. The *AgileUpdateItemBalanceListResponseABM* is

#	Activity	Remarks
		returned to *UpdateItemBalanceListAgileProvABCSImpl*
7	UpdateItemBalanceListAgileProvABCSImpl sends response back to the ItemBalanceEBS	*UpdateItemBalanceListAgileProvABCSImpl* transforms *AgileUpdateItemBalanceListResponseABM* to *UpdateItemBalanceListResponseEBM* and returns it to *ItemBalanceEBS*.
8	ItemBalanceEBS sends the UpdateItemBalanceListResponseEBM to UpdateItemBalanceListJDEE1ReqABCSImpl	*ItemBalanceEBS* sends the *UpdateItemBalanceListResponseEBM* to *UpdateItemBalanceListJDEE1ReqABCSImpl*

Chapter 7: Configuring the Process Integration for Oracle Design to Release Integration Pack for Agile Product Lifecycle Management and JD Edwards EnterpriseOne

This chapter discusses how to:

- Setting Up Agile PLM Applications
- Setting Up JD Edwards EnterpriseOne 9.0
- Setting Up Oracle Web Services Manager Security Information
- Setting Up Batch Processing Information
- Working with Domain Value Maps
- Working with Cross-References
- Handling Errors
- Viewing EBO Implementation Maps (EIMs)
- Setting Configuration Properties
- Queue Management
- Setting Up for Change Order Creation in Released Status in EBS

Setting Up Agile PLM Applications

After installation of Agile PLM PIP, the Agile Administrator is required to set up Agile Content Services.

To set up Agile PLM, follow these steps:

1. Create New Destinations
2. Create New Events for ECO and SCO
3. Define Filters
4. Create New Subscribers for ECO, SCO
5. Set Privileges
6. Modify Flex Fields
7. Create New ECO Workflow for Initial Load Change Orders
8. Set Privileges.

Creating JMS Destination

JMS Destination is created in Agile *Java Client*.

1. Create New JMS Destination

a. In Admin Tab, go to System Settings > Agile Content Service > Destinations

b. Select Protocol JMS

c. Enter or set the following essential values:

Field	Value/Setting
Name	Define your own
Response Expected	No
User Name	oc4jadmin (SOA server admin user name)
Password	welcome1
Provider Context Factory	com.evermind.server.rmi.RMIInitialContextFactory
Connection Factory	java:comp/resource/EcoRP/QueueConnectionFactories/QCF
Default Provider URL	opmn:ormi://<SOAServer>:<OPMNPort>:<instance-name>
Destination Name	java:comp/resource/EcoRP/Queues/PLM_ECO_QUEUE

d. Click **Test** to validate.

2. Create New Events for ECO and SCO

a. In Admin Tab, go to System Settings > Agile Content Service > Events

b. Enter/set the following:

Field	Value/Setting	
	ECO	SCO
Name	Define your own	
Event Type	Workflow	
Workflow	Default Change Order	Default Site Change Orders
Workflow Status	Released	

3. Define Filters

a. In Admin Tab, go to System Settings > Agile Content Service > Filters

b. Modify Default Item Filter to set the following:

Field	Value/Setting

View Tabs	Add Sites
BOM Options	Tabs and Items
BOM Levels	Select **All Levels** check box
AML Options	Tabs and Manufacturer Parts
Attachment Options	Tab only

c. Modify the following filters to set the given fields:

Field	Value/Setting	
	Default Change Order Filter	Default Site Change Order Filter
Affected Items Options	Tab & Items	Tabs & Items
Redline Changes only	No	No

4. Create New Subscribers for ECO and SCO

a. In Admin Tab, go to System Settings > Agile Content Service > Subscribers

b. Create new Subscribers, one each for ECO and SCO and set the following:

Field	Value/Setting	
	For ECO	For SCO
Name	Define your own	
Subclass	ATO	
Workflow	Default ATOs	
Criteria	All Change Orders	All Site Change Orders
Event	Select the Name of the Event that you created #2 for ECO	Select the Name of the Event that you created #2 for SCO

c. Enter/set the **Subscriber Details** for each (ECO, SCO), by adding a new row, as follows:

Field	Value/Setting	
	For ECO	For SCO
Filter	Default Change Order Filter	Default Site Change Order Filter
	Default Item Filter	Default Item Filter
Roles	All	
Format	aXML	
Language	English	

Field	Value/Setting
Site	All

d. Enable all the newly created Subscribers

5. Set Privileges

a. In Admin Tab, go to User Settings > Privileges > Modify.

b. Create new Modify Privileges for ECO and SCO.

c. Set Privilege to Modify.

d. Select the Criteria that correspond to each of the ECO and SCO.

e. Select all the Attributes, including the invisible/disabled attributes in the Applied to field and Save them.

f. In Where Used Tab, add Roles to all the created privileges. Default role is Admin user.

Note. The user should have privileges to modify the ‘released’ items and ‘released’ changes.

6. Modify Flex Fields

a. In Admin Tab, go to Data Settings > Classes.

b. For both Parts and Document classes, enable the flexfields on Page2, Page3 or Site tab according to the MultiSite_Enabled property value in AIAConfigProperties.xml for Agile PLM module.

c. These field names in Agile PLM reflect the following fields from JDE E1:

- Unit Cost
- Available Quantity
- On Hand Quantity
- Reserved Quantity

d. These fields should be the same as those entered as values for the following properties:

- Item.UnitCostAttribute
- Item.AvailableQuantityAttribute
- Item.OnHandQuantityAttribute
- Item.ReservedQuantityAttribute

e. Make the created fields as Visible

Note. Make sure that these attributes have Read and Modify privileges.

f. Click OK.

7. Create New Workflow for Initial Load Change Orders

a. In Admin Tab, go to Workflow Settings > Workflows

b. Create new Workflow for the Initial Load Change Orders and set the following:

Field	Value/Setting
Name	Define your own
Workflow Criteria Matching Type	All
Status Criteria Matching Type	Same
Object Type	Changes
Matching Criteria	All Change Orders

c. Go to Status tab and add two new workflow status and set the following:

Field	Value/Setting
Name	Define your own
Status Type	Pending
Status Stamp Color	Define your own

Field	Value/Setting
Name	Define your own
Status Type	Released
Status Stamp Color	Define your own

d. Add Workflow Criteria for each of the status created earlier and set the “Select Criteria” as “All Change Orders”

e. Enable the newly created Workflow

8. Set Privileges

a. In Admin Tab, go to User Settings > Privileges > Change Status.

b. Create new Change Status Privilege for Initial Load Change Orders workflow and set the following:

Field	Value/Setting
Name	Define your own
Description	Define your own

Field	Value/Setting
Enabled	Yes
Privilege	Change Status
Criteria	All Change Orders
Workflow	New workflow created in step 9 earlier
Status – From	Status defined in step 9 with Status type as “Pending”
Status – To	Status defined in step 9 with Status type as “Released”

c. In **Where Used** Tab, add Roles to all the created privilege. Default role is **Admin** user.

Setting Up JD Edwards EnterpriseOne 9.0

Before the Design to Release Agile PLM - JDE E1 PIP can be used; several applications should be set up in the JD Edwards EnterpriseOne systems that are used by the PIP. The integration processes and flows supported in this PIP require that JD Edwards EnterpriseOne system is set up correctly before any Initial Load or updates occur in the Agile PLM system. Therefore, it is critical that users understand how to accurately set up the JD Edwards EnterpriseOne system so that updates and synchronizations to Agile PLM are successful.

This section discusses how to:

- Set up Batch programs
- Set up the integration User Defined Code (UDC)
- Set up Business Services (BSSVs)

Setting up Batch programs

Setting up Initial Load Batch Extract Programs

Before using the Design to Release Agile PLM - JDE E1 PIP effectively, it is recommended that the Initial Load item and BOM information are extracted from JD Edwards EnterpriseOne into the Agile PLM system.

Before the Initial Load is processed using batch programs, versions for each of them need to be configured.

Initial Load programs include:

- Item Initial Load Extract to XML (R4101D3)
- BOM Initial Load Extract to XML (R3002D)

See *JD Edwards EnterpriseOne Tools 8.98 Development Tools: Batch Versions Guide.*

Setting up the Item Initial Load Extract to XML (R4101D3)

This program is used to extract Initial Load item data from JD Edwards EnterpriseOne to Agile PLM. Before running this program, the processing options need to be set, to ensure that data is extracted correctly.

The following needs to be set on the Processing Options tab:

1. **Path where XML file will be written.**

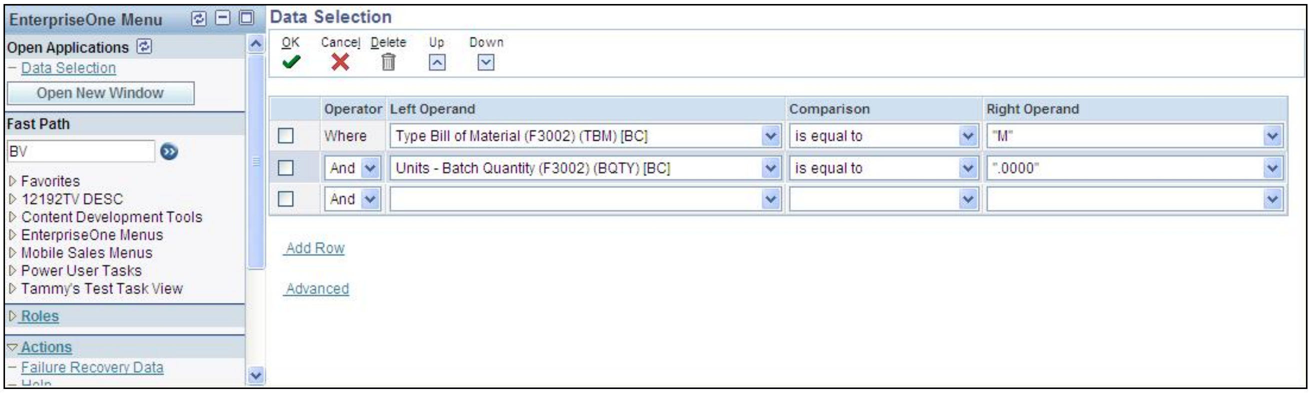
   a. This processing option is to be used to specify the file location that the system uses for storing the resulting XML file. The name of the resulting XML file will be in the format of R4101D3_MMDDYY_hhmmss.xml. The file name will be appended to the value entered in this processing option to determine the fully qualified path and name. All directories specified in the path should already exist. Also, it needs to be ensured that the value ends with appropriate '\' or '/' depending on the operating system. If this processing option is left blank, the resulting XML file will be written to the directory where this Report is running.

Setting up the BOM Initial Load Extract to XML (R3002D)

This program is used to extract Initial Load BOM data from JD Edwards EnterpriseOne to Agile PLM. Before running this program, the processing options need to be set to ensure that data is extracted correctly.

This program is provided with two default versions:

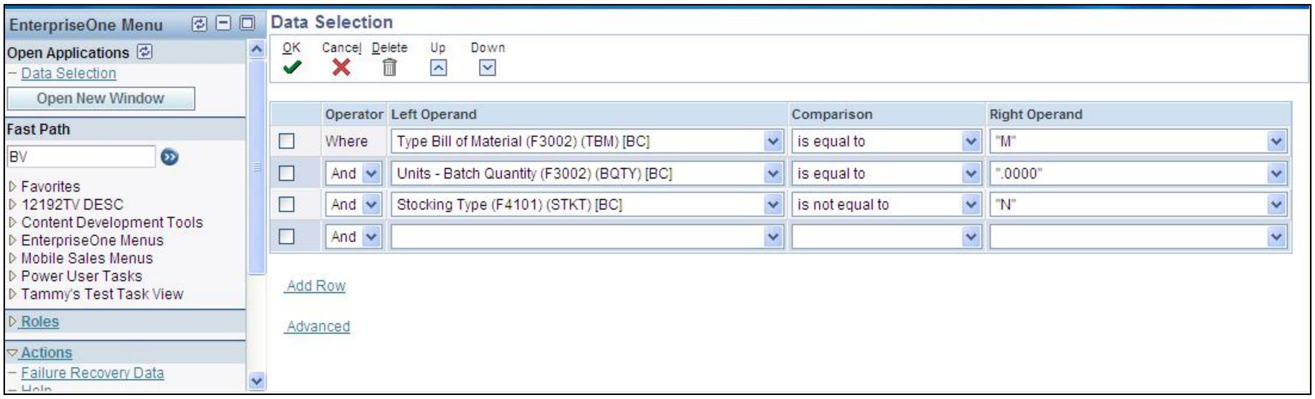
i. XJDE0001 – BOM Initial Load Extract to XML – It is recommended for Agile PLM System with Multi-Site OFF.



Data Selection

	Operator	Left Operand	Comparison	Right Operand
□	Where	Type Bill of Material (F3002) (TBM) [BC]	is equal to	"M"
□	And	Units - Batch Quantity (F3002) (BQTY) [BC]	is equal to	".0000"
□	And			

ii. XJDE0002 – BOM Initial Load – Exclude Non-Stock Components - It is recommended for Agile PLM System with Multi-Site ON.



	Operator	Left Operand	Comparison	Right Operand
☐	Where	Type Bill of Material (F3002) (TBM) [BC]	is equal to	"M"
☐	And	Units - Batch Quantity (F3002) (BQTY) [BC]	is equal to	".0000"
☐	And	Stocking Type (F4101) (STKT) [BC]	is not equal to	"N"
☐	And			

The following need to be set on the Processing Options tab:

- **Path where the XML File will be written**

This processing option is to be used to specify the file location that the system uses for storing the resulting XML file. The name of the resulting XML file will be in the format of R3002D_MMDDYY_hhmmss.xml. The file name will be appended 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 this processing option is left blank, the resulting XML file will be written to the directory where this Report is running.

- **As of Date**

This is the date used for effectivity checking. A specific date is to be entered to display documents (orders, bills of material, routings, as applicable) that are effective on or after that date. The current system date is the default, but any future or past date can be entered as well.

- **Selection for Components**

This processing option is to be used to specify if the system selects all the components associated with selected Bill of Materials. If left blank, the system will only select components that have the same Branch Plant as Parent. The skipped components detail will be written to the report output. The system will select all associated components for the Bill of Materials selected if it is set with the value '1'.

- **Extract Parent Item Revision Level**

This processing option is to be used to indicate whether the system extracts the BOM Parent Item Revision Level from Item Master (**F4101**). If left blank, the system does not extract Parent Item Revision Level. The system extracts Parent Item Revision Level if it is set with value '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, versions for each of them need to be configured.

Update ECO program includes:

- Update ECO Extract to XML (R3013D)

See *JD Edwards EnterpriseOne Tools 8.98 Development Tools: Batch Versions Guide.*

Setting up the ECO Extract to XML (R3013D)

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

This program is provided with one default version:

- **XJDE0001** – ECO Extract to XML.

There is no default Data Selection for this ECO Extract **R3013D**.

The following need to be set on the Processing Options tab:

1. Path where the XML File will be written
This processing option is to be used to specify the file location that the system uses for storing the resulting XML file. The name of the resulting XML file will be in the format of R3013D_MMDDYY_hhmmss.xml. All directories specified in the path should already exist. If this processing option is left blank, the resulting XML file will be written to the default JD Edwards EnterpriseOne system directory.

2. As of Date
This is the date used for effectivity checking. A specific date is to be entered to extract the updated ECOs from JDE E1 that are effective on or after that date. If the As of Date in the Processing Option is left blank, Integration Time Stamp Date will be used.

3. Integration Code
This processing option is to be used to specify Integration Code which is a Mandatory field with UDC (00/IE). If Integration Code is left blank then no data will be processed and XML file will not be written.

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, versions for each program should be configured.

Item List programs include:

- Item List Extract to XML **(R4102D)**

See *JD Edwards EnterpriseOne Tools 8.98 Development Tools: Batch Versions Guide.*

Setting up the Item List Extract to XML (R4102D)

This program is used to extract Item List data from JD Edwards EnterpriseOne to Agile PLM. Before running this program, the processing options are to be set to ensure that data is extracted correctly.

This program is provided with one default version:

1. XJDE0001 – Item List Extract to XML.

There is no Data Selection for this Item List Extract **R4102D**.

The following need to be set on the Processing Options tab:

1. Path where the XML File will be written

This processing option is to be used to specify the file location that the system uses for storing the resulting XML file. The name of the resulting XML file will be in the format of R4102D_MMDDYY_hhmmss.xml. All directories specified in the path should already exist. If this processing option is left blank, the resulting XML file will be written to the default JD Edwards EnterpriseOne system directory.

2. As of Date

This is the date used for effectivity checking. A specific date is to be entered to extract the updated ECOs from JDE that are effective on or after that date. If the As of Date in the Processing Option is left blank, Integration Time Stamp Date will be used.

3. Integration Code This processing option is to be used to specify Integration Code which is a Mandatory field with UDC (00/IE). If Integration Code is left blank then no data will be processed and XML file will not be written.

4. Ledger Type

This processing option is to be used to specify the Costing Method. If left as blank then cost method from Branch Plant constants will be used. If cost method is not set in Branch Plant constants then default value of Data dictionary item CSMT will be used.

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, versions for each program should be configured.

Item Balance programs include:

- Item Balance Extract to XML **(R41021D)**

See JD Edwards EnterpriseOne Tools 8.98 Development Tools: Batch Versions Guide.

Setting up the Item Balance Extract to XML (R41021D)

This program is used to extract Item Balance data from JD Edwards EnterpriseOne to Agile PLM. Before running this program, the processing options need to be set to ensure that data is extracted correctly.

This program is provided with two default versions:

XJDE0001 – Item Balance Extract to XML.

There is no Data Selection for this Item Balance Extract **R41021D**.

The following needs to be set on the Processing Options tab:

1. Path where the XML File will be written

This processing option is to be used to specify the file location that the system uses for storing the resulting XML file. The name of the resulting XML file will be in the format of R41021D_MMDDYY_hhmmss.xml. All directories specified in the path should already exist. If this processing option is left blank, the resulting XML file will be written to the default JD Edwards EnterpriseOne system directory.

2. **As of Date**

   This is the date used for effectivity checking. A specific date is to be entered to extract the updated ECOs from JDE that are effective on or after that date. If the As of Date in the Processing Option is left blank, Integration Time Stamp Date will be used.

3. **Integration Code**

   This processing option is to be used to specify Integration Code which is a mandatory field with UDC (00/IE). If Integration Code is left blank then no data will be processed and XML file will not be written.

Setting up UDC

A UDC is set up to store the information regarding the types of Integration codes used.

To set up UDC, follow these steps:

1. Go to UDC through fast path.
2. Enter Product Code as "00" and User Defined Codes as "IE"
3. Click Add. Enter the Code and Description as ‘AgileJDE’ and ‘Agile to JDE Integration’ respectively.
4. Click OK save the UDC values.

Setting up JD Edwards EnterpriseOne Web Services

These JD Edwards EnterpriseOne web services, also called business services, are used in the Design to Release PIP:

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

For more information, see JD Edwards EnterpriseOne Business Services 9.0 Reference Guide.

EngineeringChangeOrderManager (JP300000)

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

Operation	Description
EngineeringChangeOrdersProcessor (J3000010)	Use this operation to call other Processors internally to add records into Item, Item Branch and ECO files.
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.

See *JD Edwards EnterpriseOne Business Services 9.0 Reference Guide, for all the web services listed in the table.*

IntegrationTimeStampManager (JP300010)

The IntegrationTimeStampManager Published Business Service (PBSSV) manages the processing of the following web service operation:

Operation	Description
ProcessIntegrationTimeStamp (J3000030)	Use this operation to add/modify the Integration Time Stamp table.

See *JD Edwards EnterpriseOne Business Services 9.0 Reference Guide,* "IntegrationTimeStampManager*".*

EngineeringChangeOrdersProcessor (J3000010)

The EngineeringChangeOrdersProcessor web service (BSSV) is called by the EngineeringChangeOrderManager PBSSV from the provider when a user creates an ECO. For the Design to Release PIP, the external system is Agile PLM. The user creates an ECO in Agile PLM and releases the same to the JD Edwards EnterpriseOne system. This Processor calls Create ECO (**B3004100**) business function to add a header record in **F4801**. This generates an ECO Number, which is used to create a BOM in JD Edwards EnterpriseOne.

The EngineeringChangeOrdersProcessor Published Business Service (BSSV) manages the processing of the following web service operations:

Operation	Description
EngineeringChangeOrderPartsListProcessor (J3000020)	Use this operation to add ECOs.
InventoryItemsProcessor (J4100020)	Use this operation to add or modify Item records.
InventoryItemBranchProcessor (J4100030)	Use this operation to add or modify Item Branch records.

See *JD Edwards EnterpriseOne Business Services 9.0 Reference Guide, "EngineeringChangeOrdersProcessor".*

EngineeringChangeOrderPartsListProcessor (J3000020)

The EngineeringChangeOrderPartsListProcessor web service is called by the EngineeringChangeOrdersProcessor. This Processor calls Create ECO (**B3004100**) business function to add a parts list and related items in JD Edwards EnterpriseOne.

See *JD Edwards EnterpriseOne Business Services 9.0 Reference Guide, "EngineeringChangeOrderPartsListProcessor".*

ProcessIntegrationTimeStamp (J3000030)

The ProcessIntegrationTimeStamp web service is called by the IntegrationTimeStampManager PBSSV. This Processor will add or modify the Integration Time Stamp table (**F0095**) with the last successful run time.

See *JD Edwards EnterpriseOne Business Services 9.0 Reference Guide, "ProcessIntegrationTimeStamp".*

InventoryItemsProcessor (J4100020)

The InventoryItemsProcessor web service is called by the EngineeringChangeOrdersProcessor. This Processor calls **F4101** ItemMasterAddition (**B4101062**) business function to add items in JD Edwards EnterpriseOne.

See *JD Edwards EnterpriseOne Business Services 9.0 Reference Guide, "InventoryItemsProcessor".*

InventoryItemBranchProcessor (J4100030)

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

See *JD Edwards EnterpriseOne Business Services 9.0 Reference Guide, "InventoryItemsProcessor".*

Operation	Description
EngineeringChangeOrdersProcessor (J3000010)	Use this operation to call other Processors internally to add records into Item, Item Branch and ECO files.
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.

Setup for BSSVs:

To set up BSSVS, follow these steps:

1. Enter P951000, P951000 in the fast path.
2. Click Add.
3. Enter the values in the Key and Group fields as shown below:

Key	Value
J4100003_ITEM_MBF_VERSION	For Example: ZJDE0001
J4100003_ITEM_STOCKING_TYPE_CODE	For Example: S
J3000010_ECO_MBF_VERSION	For Example: ZJDE0001

Setting Up Oracle Web Services Manager Security Information

The Oracle Web Services Manager (OWSM) client agent needs to be set up to store the security credentials that are used by the EnterpriseOne web services.

To set up OWSM security information:

1. Sign in to the OWSM system and select the Policy Management section.
2. From the Manage Policies page, click Add New Component.
3. Enter a new component using the data show below:

Component Name	Enter a name for the component, such as *E1–WS–Client–Agent.*
Component Type	Select *Client Agent.*
Container Type	Select *OC4J.*

4. Click Register.
5. Record the component ID.
6. Create this directory: <ORACLE_HOME>/e1_client_agent.
7. Access the agent.properties in the <ORACLE_HOME>/owsm/bin directory.
8. Update the agent.properties so that:

```
agent.componentType=OC4JClientInterceptor
client.home=<ORACLE_HOME>/e1_client_agent
agent.component.id is equal to the value that you recorded in step 4.
```

9. Navigate to the directory listed in step 5 and run the command `./wsmadmin.sh installAgent.`

10. Copy the SetUsernamePasswordStep.xml file to your local machine. The file is located at <AIA_HOME>/PIPS/Core/JDEE1/Utilities/OWSM.

11. Return to the Manage Policies form in OWSM and select Steps.

12. Click Add New Steps to begin the process of adding a new step for the component ID that you created in step 4.

13. Browse to the SetUsernamePasswordStep.xml file that you copied to your local machine and click Upload.

14. In OWSM, access your component default policy by clicking the policies link, and then click Edit.

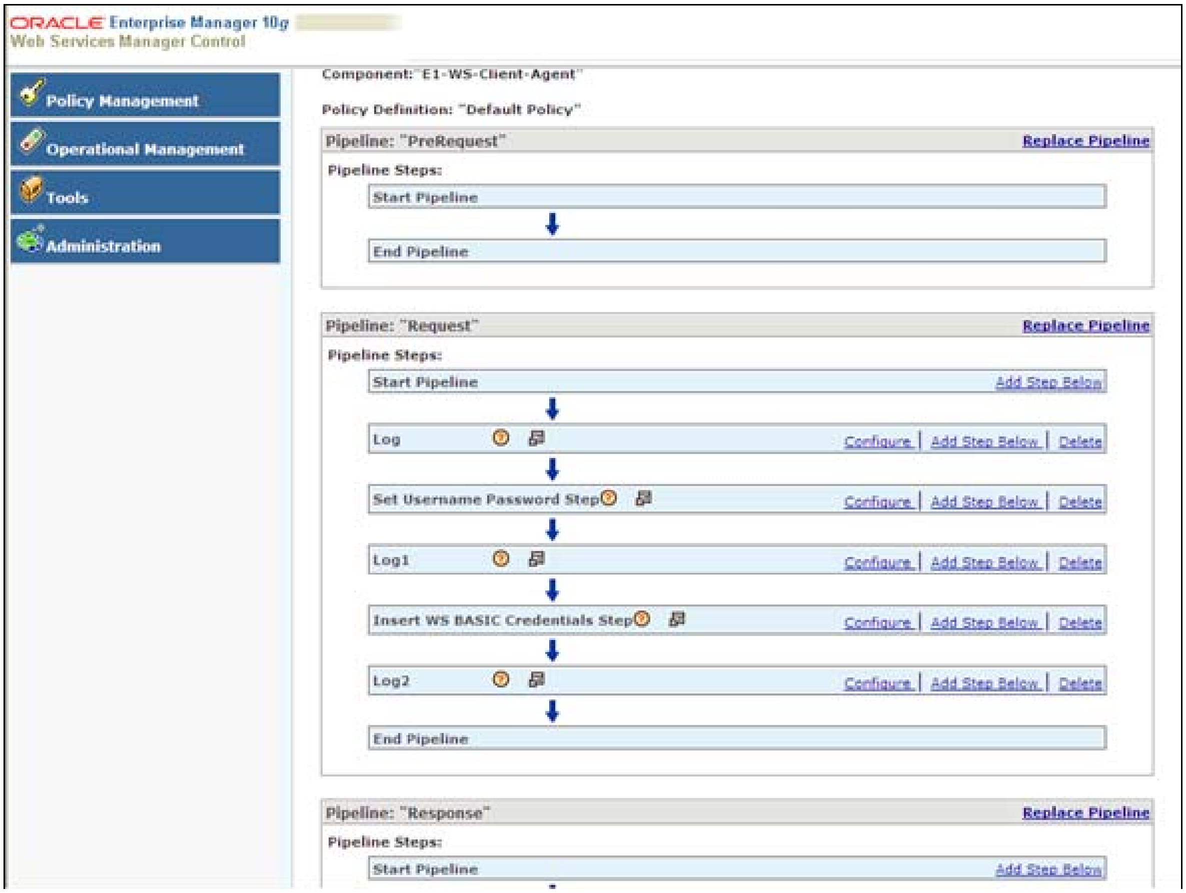
15. Edit the Request pipeline to set the user password and add WS security.

16. Add the steps listed here by clicking the Add Step Below link in the appropriate location. Oracle also recommends that you add a log step before and after each of these steps:

    a. Insert the Set Username Password Step.

    c. Insert the Insert WS Basic Credentials Step.

The request pipeline should match with the page shown here:



ORACLE Enterprise Manager 10g
Web Services Manager Control

Policy Management
Operational Management
Tools
Administration

Component: "E1-WS-Client-Agent"

Policy Definition: "Default Policy"

Pipeline: "PreRequest"

Pipeline Steps:

- Start Pipeline
- End Pipeline

Pipeline: "Request"

Pipeline Steps:

- Start Pipeline
- Log

Configure | Add Step Below | Delete
- Set Username Password Step

Configure | Add Step Below | Delete
- Log1

Configure | Add Step Below | Delete
- Insert WS BASIC Credentials Step

Configure | Add Step Below | Delete
- Log2

Configure | Add Step Below | Delete
- End Pipeline

Pipeline: "Response"

Pipeline Steps:

- Start Pipeline

Request Pipeline

Editing the Request Pipeline

To edit the request pipeline, follow these steps:

1. Configure the Set Username Password Step to enter the actual user name and password data, and save the configuration using the following information:
   - Step name is Set Username Password.
   - Set the basic properties as:
   - Enabled
   - Type: Boolean
   - Value: *True*
   - Set the User Credentials for Username (*) as:
   - Type: string

- Value: Enter the user name, for example, username: *DN=REGUSR01, ENV=PY812SOD, ROLE=TG2*

2. Set the User Credentials for Password (*) as:

   - Type: string
   - Value: Enter the password for the associated user name.

   No faults exist for this step.

3. Click the Commit Policy link.

4. Get the wsif-wsm-config.xml file from <AIA_HOME>/PIPS/Core/JDEE1/Utilities/OWSM. Update the file by replacing the server path for the client agent folder that was created and the component ID that was created in OWSM. Place the updated file in the following location: <ORACLE_HOME>/j2ee/<instance_name>/config directory

5. Confirm the oracle.wsm.agent shared library is present in the server.xml located at <ORACLE_HOME>/j2ee/<instance_name>/config directory. If the shared library is not present, follow the directions in the next step. Otherwise, it can be skipped.

6. To generate the oracle.wsm.agent shared library, create a temporary server agent. Follow steps 1-8 in this section except chose Server Agent instead of Client Agent in step 3. Update the <ORACLE_HOME>/owsm/bin/agent.properties file with:

   - agent.componentType: OC4JServerInterceptor
   - agent.comonent.id: C0009999 (use the identifier generated when the agent was created)

7. When running installAgent, use the following command:

   - wsmadmin.sh installAgent –Doc4jAdminPassword=<SOA Admin Password>

8. After the agent is installed, verify the oracle.wsm.agent library has been added to the server.xml file. The temporary server agent can be deleted.

9. Restart SOA to update the system with the changes.

Setting Up Batch Processing Information

To utilize the Initial Load and update flows from JDE E1, batch information needs to be set up first. This section discusses how to:

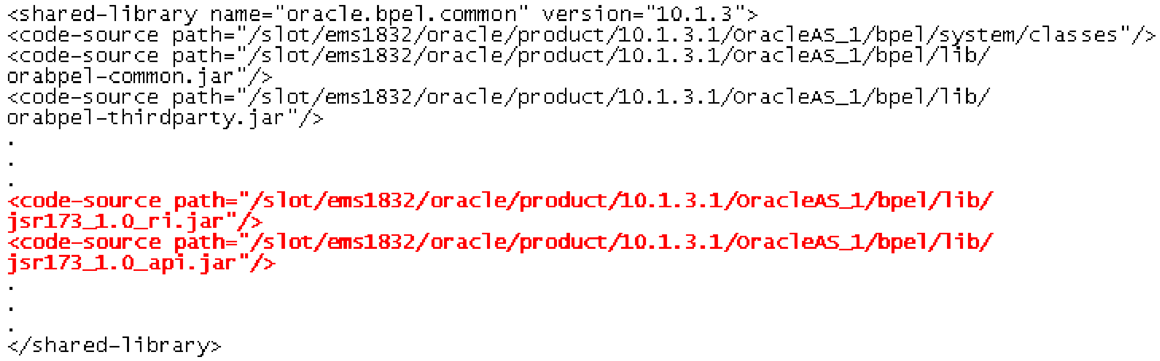
- Set up debatching information.
- Set up batch consumer properties in the ESB Console.
- Set up the FTP adapter.
- Set up Resequencer for Batch Processing
- Set up Invoke Utilities for Initial Load

Setting Up Debatching Information

The debatching process is used to split single files that contain a large number of records into smaller files, with fewer records, so that they can be consumed and processed in a more efficient manner.

For debatching to function properly, the server.xml file should be updated, which is located at: <ORACLE_HOME>/j2ee/<instance_name>/config/ by adding the jar files to the following section: <shared-library name="oracle.bpel.common" version="10.1.3">

Use this example for assistance with updating the server.xml file. The text that appears in red should be added to the file. It should be ensured that the root, /slot/ems1832/ etc in this case, is updated to reflect the actual server path.



```
<shared-library name="oracle.bpel.common" version="10.1.3">
<code-source path="/slot/ems1832/oracle/product/10.1.3.1/OracleAS_1/bpel/system/classes"/>
<code-source path="/slot/ems1832/oracle/product/10.1.3.1/OracleAS_1/bpel/lib/
orabpel-common.jar"/>
<code-source path="/slot/ems1832/oracle/product/10.1.3.1/OracleAS_1/bpel/lib/
orabpel-thirdparty.jar"/>
.
.
.
<code-source path="/slot/ems1832/oracle/product/10.1.3.1/OracleAS_1/bpel/lib/
jsr173_1.0_ri.jar"/>
<code-source path="/slot/ems1832/oracle/product/10.1.3.1/OracleAS_1/bpel/lib/
jsr173_1.0_api.jar"/>
.
.
.
</shared-library>
```

Example: server.xml file

Setting Up Batch Consumer Properties

1. For the **Initial Load** processes and other update processes to work correctly, ESB properties should be set up for these batch consumer services:

- ItemInitialLoadExtractJDEE1FTPConsumer

- ItemInitialLoadExtractJDEE1FileConsumer
- BillOfMaterialsInitialLoadExtractJDEE1FTPConsumer
- BillOfMaterialsInitialLoadExtractJDEE1FileConsumer

To set these properties, the ESB Console has to be opened, the services need to be located and then the Properties tab has to be selected.

The FileLocation property for the ItemInitialLoadExtractJDEE1FTPConsumer 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 FileLocation property should be \E900\DDP\PrintQueue. The FileLocation property for the BillOfMaterialsInitialLoadExtractJDEE1FTPConsumer should match the path specified in the processing options for the **R3002D** relative to the FTP server default directory.

The FileLocation property for the ItemInitialLoadExtractJDEE1FileConsumer and BillOfMaterialsInitialLoadExtractJDEE1FileConsumer should match the location of *<ORACLE_HOME>/JDEE1In/.* These services are delivered with an example path which is; */slot/ems2005/oracle/product/10.1.3.1/OracleAS_1/JDEE1In/.* This path should be updated to reflect the implementation. After the extract programs are executed in JD Edwards EnterpriseOne, the files should be moved to the JDEE1 In folder in order for these file consumer services to find and process the files.

All of the consumer services listed in this section have the following two properties, which can be configured as follows:

Property	Description
BatchSize	This property determines how many records are included in each batch. The default value for this property is *60.* If you use a value larger than 60, the consumer services will not function properly, as Agile Integration Services (AIS) would consume more time and cause time out in middleware. Oracle recommends that you do not change this value.
PollingFrequency	This property specifies the time interval, in seconds, when the file consumers check the fileLocation for new files to process. The default value for this property is *30.* Note: Oracle recommends that you do not change the pollingFrequency or batchSize properties for the FTP adapter.

2. For the **ECO, Item List and Item Balance** processes and other update processes to work correctly, ESB properties should be set up for these batch consumer services:

- EngineeringChangeOrderListExtractJDEE1FTPConsumer
- EngineeringChangeOrderListExtractJDEE1FileConsumer

- ItemListExtractJDEE1FTPConsumer
- ItemListExtractJDEE1FileConsumer
- ItemBalanceListExtractJDEE1FTPConsumer
- ItemBalanceListExtractJDEE1FileConsumer

To set these properties, the ESB Console has to be opened, the services are to be located and then the Properties tab needs to be selected.

The FileLocation property for the **EngineeringChangeOrderListExtractJDEE1FTPConsumer** should match the path specified in the **R3013D** processing option. The physicalDirectory property for the **ItemListExtractJDEE1FTPConsumer** should match the path specified in the processing options for the **R4102D**. The physicalDirectory property for the **ItemBalanceListExtractJDEE1FTPConsumer** should match the path specified in the processing options for the **R41021D**. These paths should be relative to the FTP setup path on the EnterpriseOne Enterprise Server.

The FileLocation property for the **EngineeringChangeOrderListExtractJDEE1FileConsumer**, **ItemListExtractJDEE1FileConsumer** and **ItemBalanceListExtractJDEE1FileConsumer** should match the location of *<ORACLE_HOME>/JDEE1In/.* These services are delivered with an example path which is; */slot/ems2005/oracle/product/10.1.3.1/OracleAS_1/JDEE1In/.* This path should be updated to reflect your implementation. After the extract programs are executed in JD Edwards EnterpriseOne, the files should be moved to the JDEE1 In folder in order for these file consumer services to find and process the files.

All consumer services listed in this section have two properties to be configured as follows:

Property	Description
BatchSize	This property determines how many records are included in each batch. The default value for this property is *40.*
PollingFrequency	This property specifies the time interval, in seconds, when the file consumers check the fileLocation for new files to process. The default value for this property is *10.* **Note:** Oracle recommends that you do not change the pollingFrequency or batchSize properties for the FTP adapter.

For more information, see *Enterprise Service Bus Quick Start Guide 10g*, “Creating, Configuring, and Managing an Oracle Enterprise Service Bus”.

Setting Up FTP Adapter Information

To support the bulk load processes for the Lead to Order PIP, the FTP adapter may be set up. The FTP adapter will allow the JDE E1 Batch jobs to be processed directly from the JDE E1 Enterprise Server. If FTP is not set up, the extract xml files will need to be manually moved to the FMW server for processing. To set up FTP adapter, the oc4j-ra.xml file needs to be modified at the following location:

<ORACLE_HOME>/j2ee/<instance_name>/application-deployments/default/FtpAdapter/

The host, port, username, password, ftpAbsolutePathBegin, ftpPathSeparator, and serverType values should be set according to the FTP setup on the EnterpriseOne Enterprise Server. Copy an existing connector factory and make the name eis/Ftp/JDEE1FtpAdapter.
The following text can be used as an example of how to modify this file:

```
<?xml version="1.0"?>
<oc4j-connector-factories xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://www.oracle.com/technology/oraclea
s/schema/
oc4j-connector-factories-10_0.xsd" schema-major-version="10" schema-
minor-version=□
"0" >
<imported-shared-libraries>
<import-shared-library name="oracle.bpel.common"/>
<import-shared-library name="oracle.xml"/>
</imported-shared-libraries>
.
.
.
.
<connector-factory location="eis/Ftp/JDEE1FtpAdapter" connector-
name="Ftp Adapter">
<config-property name="host" value="denmlsan011.mlab.JDEdwards.com"/>

<config-property name="port" value="21"/>
<config-property name="username" value="psft"/>
<config-property name="password" value="psftpsft"/>
<config-property name="ftpAbsolutePathBegin" value="Z:"/>
<config-property name="ftpPathSeparator" value="/"/>
<config-property name="keepConnections" value="true"/>
<config-property name="serverType" value="win"/>
<config-property name="serverLocaleLanguage" value=""/>
<config-property name="serverLocaleCountry" value=""/>
<config-property name="serverLocaleVariant" value=""/>
<config-property name="serverEncoding" value=""/>
<config-property name="useFtps" value="false"/>
<config-property name="useImplicitSSL" value="false"/>
<config-property name="walletLocation" value="C:\wallet\ewallet.p12"/>
<config-property name="walletPassword" value="welcome1"/>
<config-property name="channelMask" value="both"/>
<config-property name="securePort" value="990"/>
<config-property name="keyStoreProviderName" value=""/>
<config-property name="keystoreType" value=""/>
<config-property name="keystoreAlgorithm" value=""/>
```

```
<config-property name="enableCipherSuits" value=""/>
<config-property name="proxyHost" value=""/>
<config-property name="proxyPort" value=""/>
<config-property name="proxyUsername" value=""/>
<config-property name="proxyPassword" value=""/>
<config-property name="proxyType" value=""/>
<config-property name="proxyDefinitionFile" value=""/>
<config-property name="useProxy" value="false"/>
<config-property name="useSftp" value="false"/>
<config-property name="authenticationType" value="password"/>
<config-property name="preferredKeyExchangeAlgorithm" value="diffie-
hellman-
group1-sha1"/>
<config-property name="preferredCompressionAlgorithm" value="none"/>
<config-property name="preferredDataIntegrityAlgorithm" value="hmac-
md5"/>
<config-property name="preferredPKIAlgorithm" value="ssh-rsa"/>
<config-property name="privateKeyFile" value=""/>
<config-property name="preferredCipherSuite" value="blowfish-cbc"/>
<config-property name="transportProvider" value="socket"/>
<connection-pooling use="none">
</connection-pooling>
<security-config use="none">
</security-config>
</connector-factory>
</oc4j-connector-factories>

ftpAbsolutePathBegin should be set to the default directory of the FTP
server on the JDEE1 Enterprise Server.
```

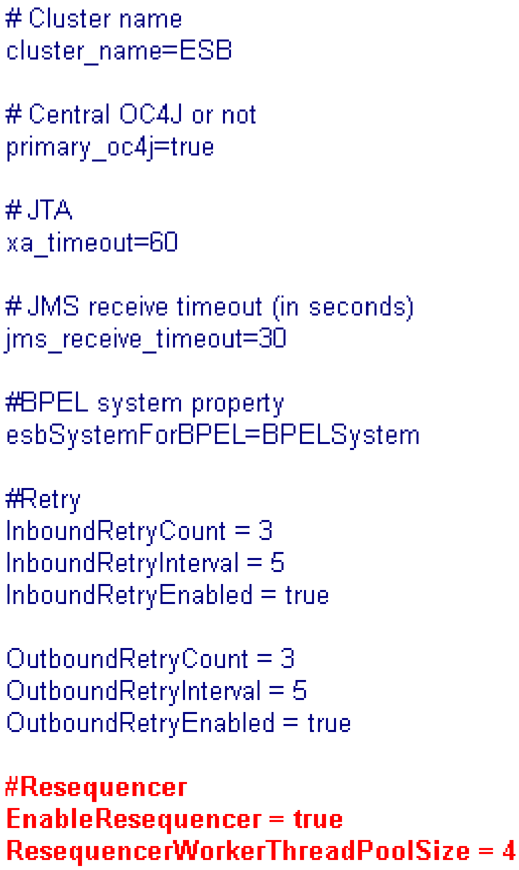
- If the JD Edwards enterprise server is an AS/400, serverType should be “unix”.
- If the JD Edwards Enterprise server is a Windows server, serverType should be “win”.

Setting Up Resequencer for Batch Processing

To use the ESB Resequencer for Initial Load error handling, the esb_config.ini file should be updated at <ORACLE_HOME>/integration/esb/config/ to enable Resequencer and set the number of threads for Initial Load processing.

This change requires a restart.

The #Resequencer section, depicted in red in the following illustration, should be added to the file, as it will not exist after installation of the PIP:



```
# Cluster name
cluster_name=ESB

# Central OC4J or not
primary_oc4j=true

# JTA
xa_timeout=60

# JMS receive timeout (in seconds)
jms_receive_timeout=30

#BPEL system property
esbSystemForBPEL=BPELSystem

#Retry
InboundRetryCount = 3
InboundRetryInterval = 5
InboundRetryEnabled = true

OutboundRetryCount = 3
OutboundRetryInterval = 5
OutboundRetryEnabled = true

#Resequencer
EnableResequencer = true
ResequencerWorkerThreadPoolSize = 4
```

Adjusting Timeout Settings

The xa_timeout setting in the esb_config.ini at <ORACLE_HOME>/integration/esb/config/ needs to be adjusted. The default value is set to 60 seconds; however, the same should be increased to at least 360, to accommodate batch processing. This setting might need to be increased, depending on the SOA setup and message size.

Setting Transaction Timeout Values

The values for the Configuration files should be as follows:

Configuration File	Timeout Value
<ORACLE_HOME>\j2ee\home\config\transaction-manager.xml	Transaction-timeout 420
<ORACLE_HOME>\j2ee\home\applicationdeployments\orabpel\ejb_ob_engine\orion-ejb-jar.xml	Transaction-timeout 380
<ORACLE_HOME>\bpel\domains\domain_name\config\domain.xml – see note below	SyncMaxWaitTime 360

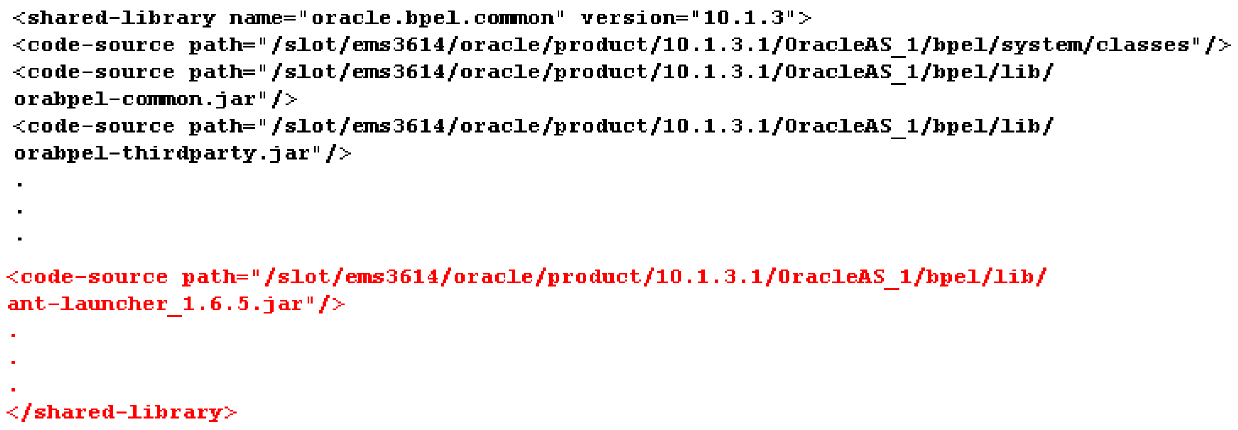
Note. The syncMaxWaitTime setting must be set using the configurations tab on the BPEL console and should not be manually updated. Also, the values here are suggested values and might need to be increased to accommodate large batches of data.

Setting Up Invoke utilities for Initial Load

The InvokeAIS and InvokeSDK Java utilities are used to invoke an ant script to call Agile Integration Service (AIS) and Agile SDK Java utilities. For the invoke utilities to invoke Ant scripts properly, the server.xml file located at: <ORACLE_HOME>/j2ee/<instance_name>/config/, should be updated by adding the jar file to the following section:

<shared-library name="oracle.bpel.common" version="10.1.3">

This example can be used for assistance with updating the server.xml file. The text that appears in red should be added to the file. It should be ensured that the root, /slot/ems1832/ in this case, are updated to reflect the actual server path.



```
<shared-library name="oracle.bpel.common" version="10.1.3">
<code-source path="/slot/ems3614/oracle/product/10.1.3.1/OracleAS_1/bpel/system/classes"/>
<code-source path="/slot/ems3614/oracle/product/10.1.3.1/OracleAS_1/bpel/lib/
orabpel-common.jar"/>
<code-source path="/slot/ems3614/oracle/product/10.1.3.1/OracleAS_1/bpel/lib/
orabpel-thirdparty.jar"/>
.
.
.
<code-source path="/slot/ems3614/oracle/product/10.1.3.1/OracleAS_1/bpel/lib/
ant-launcher_1.6.5.jar"/>
.
.
.
</shared-library>
```

```
        <shared-library name="oracle.bpel.common" version="10.1.3">
                <code-source
path="/slot/ems3614/oracle/product/10.1.3.1/OracleAS_1/bpel/system/classes"/>
                <code-source
path="/slot/ems3614/oracle/product/10.1.3.1/OracleAS_1/bpel/lib/orabpel-common.jar"/>
                <code-source
path="/slot/ems3614/oracle/product/10.1.3.1/OracleAS_1/bpel/lib/orabpel-thirdparty.jar"/>
                .
                .
                .
                <code-source
path="/slot/ems3614/oracle/product/10.1.3.1/OracleAS_1/bpel/lib/ant-launcher_1.6.5.jar"/>
                .
                .
                .
        </shared-library>
```

Example: server.xml file

Working with Domain Value Maps

Domain Value Maps templates are XML files that conform to the Oracle SOA Suite DVM schema. The DVMs are stored in the ESB database and are maintained using the ESB Console user interface. Domain values are used for static lookups. Using ESB administration, these XML files may be imported and then modified according to requirements.

Out-of-the-Box DVMs

During installation, the DVMs used for the PIP are imported with default data mappings. The values mapped by these DVMs have to be changed as needed. There are many DVMs that are seeded and need not be touched. Since most of the Agile PLM attributes being mapped are list values, the Agile PLM data is not seeded and should be changed accordingly.

Some of the most used DVMs, with their Out-of-the-Box values, are listed herein. Their values can be modified as per requirements. For complete list of available DVMs, including the most used ones, the List of DVMs below can be referred to.

Warning! More rows of value mappings may be added, but the DVM name, column names, or the number of columns must not be changed.

Sr.No	DVM Name	Description
1	ITEM_DUAL_UOM_TRACKING_INDICATOR.xml	ITEM_OBJ/DualUOMTrackingIndicator
2	ITEM_INDICATOR.xml	ITEM_OBJ/ItemPurchasingCharacteristics/TaxableIndicator ITEM_OBJ/ItemManufacturingCharacteristics/StructureAllowedIndicator
3	ITEM_INVENTORY_PLANNING_CODE.xml	ITEM_OBJ/ItemPlanningCharacteristics/InventoryPlanningCode
4	ITEM_INVOICINGENABLEDINDICATOR.xml	ITEM_OBJ/ItemOrderManagementCharacteristics/InvoicingEnabledIndicator
5	ITEM_ITEMSPECIFICATIONGROUP_CYCLECOUNTCATEGORY.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
6	ITEM_ITEMSPECIFICATIONGROUP_PURCHASINGREPORTCODE1.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
7	ITEM_ITEMSPECIFICATIONGROUP_PURCHASINGREPORTCODE2.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
8	ITEM_ITEMSPECIFICATIONGROUP_PURCHASINGREPORTCODE3.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
9	ITEM_ITEMSPECIFICATIONGROUP_PURCHASINGREPORTCODE4.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification

Sr.No	DVM Name	Description
10	ITEM_ITEMSPECIFICATIONGROUP_PURCHASINGREPORTCODE5.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
11	ITEM_ITEMSPECIFICATIONGROUP_PURCHASINGREPORTCODE6.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
12	ITEM_ITEMSPECIFICATIONGROUP_PURCHASINGREPORTCODE7.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
13	ITEM_ITEMSPECIFICATIONGROUP_PURCHASINGREPORTCODE8.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
14	ITEM_ITEMSPECIFICATIONGROUP_PURCHASINGREPORTCODE9.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
15	ITEM_ITEMSPECIFICATIONGROUP_PURCHASINGREPORTCODE10.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
16	ITEM_ITEMSPECIFICATIONGROUP_SALESREPORTINGCODE1.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
17	ITEM_ITEMSPECIFICATIONGROUP_SALESREPORTINGCODE2.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
18	ITEM_ITEMSPECIFICATIONGROUP_SALESREPORTINGCODE3.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
19	ITEM_ITEMSPECIFICATIONGROUP_SALESREPORTINGCODE4.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
20	ITEM_ITEMSPECIFICATIONGROUP_SALESREPORTINGCODE5.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
21	ITEM_ITEMSPECIFICATIONGROUP_SALESREPORTINGCODE6.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
22	ITEM_ITEMSPECIFICATIONGROUP_SALESREPORTINGCODE7.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
23	ITEM_ITEMSPECIFICATIONGROUP_SALESREPORTINGCODE8.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
24	ITEM_ITEMSPECIFICATIONGROUP_SALESREPORTINGCODE9.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
25	ITEM_ITEMSPECIFICATIONGROUP_SALESREPORTINGCODE10.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
26	ITEM_ITEMSPECIFICATIONGROUP_SHIPPINGCOMMODITYCLASS.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
27	ITEM_ITEMSPECIFICATIONGROUP_SHIPPINGCONDITIONSCODE.xml	ITEM_OBJ/ItemSpecificationGroup/SpecificationGroup/Specification
28	ITEM_MAKEORBUY_CODE.xml	ITEM_OBJ/ItemPlanningCharacteristics/MakeOrBuyCode
29	ITEM_PURCHASINGALLOWEDINDICATOR.xml	ITEM_OBJ/ItemPurchasingCharacteristics/PurchasingAllowedIndicator
30	ITEM_REPLENISHMENT_SOURCE_CODE.xml	ITEM_OBJ/ItemPlanningCharacteristics/ReplenishmentSourceCode
31	ITEM_SERIALIZATION_EVENT_CODE.xml	ITEM_OBJ/InventoryCharacteristics/SerializationEventCode
32	ITEM_STOCKINGALLOWEDINDICATOR.xml	ITEM_OBJ/InventoryCharacteristics/StockingAllowedIndicator

Sr.No	DVM Name	Description
33	ITEM_TYPE.xml	ITEM_OBJ/TypeCode
34	ITEM_UOM_CODE.xml	ITEM_OBJ/BaseUOMCode ITEM_OBJ/ShippingUOMCode ITEM_OBJ/SecondaryUOMCode ITEM_OBJ/ItemPhysicalCharacteristics/VolumeMeasure ITEM_OBJ/ItemPurchasingCharacteristics/IssueUOMCode ITEM_OBJ/ItemPhysicalCharacteristics/WeightMeasure
35	ITEM_STATUS_CODE	ITEM_OBJ/Status/Code
36	ECO_CLASSIFICATION_CODE.xml	ECO_OBJ/ClassificationCode
37	ECO_ECOSPECIFICATIONGROUP_CATEGORYCODE1.xml	ECO_OBJ/EngineeringChangeOrderSpecificationGroup/SpecificationGroup/Specification
38	ECO_ECOSPECIFICATIONGROUP_CATEGORYCODE2.xml	ECO_OBJ/EngineeringChangeOrderSpecificationGroup/SpecificationGroup/Specification
39	ECO_ECOSPECIFICATIONGROUP_CATEGORYCODE3.xml	ECO_OBJ/EngineeringChangeOrderSpecificationGroup/SpecificationGroup/Specification
40	ECO_ECOSPECIFICATIONGROUP_CATEGORYCODE4.xml	ECO_OBJ/EngineeringChangeOrderSpecificationGroup/SpecificationGroup/Specification
41	ECO_ECOSPECIFICATIONGROUP_CATEGORYCODE5.xml	ECO_OBJ/EngineeringChangeOrderSpecificationGroup/SpecificationGroup/Specification
42	ECO_ECOSPECIFICATIONGROUP_CATEGORYCODE6.xml	ECO_OBJ/EngineeringChangeOrderSpecificationGroup/SpecificationGroup/Specification
43	ECO_ECOSPECIFICATIONGROUP_CATEGORYCODE7.xml	ECO_OBJ/EngineeringChangeOrderSpecificationGroup/SpecificationGroup/Specification
44	ECO_ECOSPECIFICATIONGROUP_CATEGORYCODE8.xml	ECO_OBJ/EngineeringChangeOrderSpecificationGroup/SpecificationGroup/Specification
45	ECO_ECOSPECIFICATIONGROUP_CATEGORYCODE9.xml	ECO_OBJ/EngineeringChangeOrderSpecificationGroup/SpecificationGroup/Specification
46	ECO_ECOSPECIFICATIONGROUP_CATEGORYCODE10.xml	ECO_OBJ/EngineeringChangeOrderSpecificationGroup/SpecificationGroup/Specification
47	ECO_PRIORITY_CODE.xml	ECO_OBJ/PriorityCode
48	ECO_REASON_CODE.xml	ECO_OBJ/ReasonCode
49	ECO_STATUS_CODE.xml	ECO_OBJ/Status/Code
50	ECO_TYPECODE.xml	ECO_OBJ/TypeCode

Working with Cross-References

Cross-references map and connect the 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.

This table lists the cross-references:

Name	Purpose	Columns	Mapping Details
ITEM_ITEMID	Query/Create	JDEE1_01, COMMON, AGILE9226_01	
CHANGE_CHANGEID	Query/Create	JDEE1_01, COMMON, AGILE9226_01	

Handling Errors

The Design to Release PIP uses the ESB 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, review the product documentation for the programs or objects that are being used.

For more information, see Oracle Application Integration Architecture – Foundation Pack: Core Infrastructure Components Guide, “Setting Up and Using Error Handling and Logging.”

Resequencing and Error Handling

The Initial Load flow in Design to Release Agile PLM - JDE E1 PIP uses the ESB Resequencer feature to manage any errors or 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:

- ItemInitialLoadExtractJDEE1FileConsumer_RS
- ItemInitialLoadExtractJDEE1FTPConsumer_RS
- BillOfMaterialsInitialLoadExtractJDEE1FileConsumer_RS
- BillOfMaterialsInitialLoadExtractJDEE1FTPConsumer_RS

ItemInitialLoadExtractJDEE1FileConsumer_RS / ItemInitialLoadExtractJDEE1FTPConsumer_RS and BillOfMaterialsInitialLoadExtractJDEE1FileConsumer_RS / BillOfMaterialsInitialLoadExtractJDEE1FTPConsumer_RS are initiated before the Initial Load Item and BillOfMaterials JDEE1toAgile PLM services in their respective flows, and they protect the Item and Bill Of Materials flows from EnterpriseOne to Agile PLM. If an error occurs in either of these flows, the message remains in the resequencer store that is associated with the ESB 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, 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 throttle the Initial Load processing so that all messages are split into a number of threads defined in ESB Resequencing configuration. By default, the system creates 4 groups, which are called 1, 2, 3, and 4.
After unlocking the group, the message that contained the failed record will reprocess. Therefore, the error should be identified and corrected *before* unlocking the group. Once the failed message is successfully processed, the system processes any messages that are after it in the same group.

A group can be unlocked by connecting to the OAS database with user name *oraesb* and password *oraesb,* and running the following script:

```
update esb_group_status set error=0, lockConId='NONE'
where error!=0 and lockConId != 'NONE' and group_id='844334'
and service_guid in (select guid from wf_events where name ='
ItemInitialLoadExtractJDEE1FileConsumer_RS');
```

Note. These services each have configurable ESB properties, ResequencerType, InvocationMode and ResequencerGroupXPath. Oracle recommends that the value of these properties is not changed.

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:

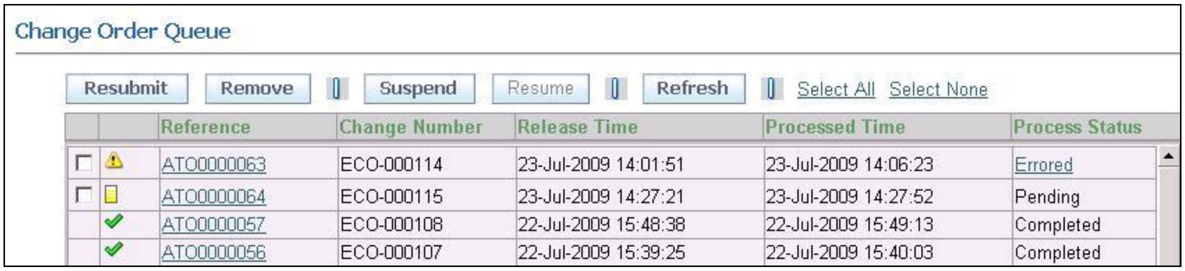
- If all four runtime values (SYSTEM_CODE, ERROR_CODE, SERVICE_NAME, and PROCESS_NAME) are available and they map to an error notification entry in this table, use the specified notification roles.
- If the ERROR_CODE, SERVICE_NAME, and PROCESS_NAME are available and map to an error notification entry 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 are available, the default values are fetched from the AIAConfigurationProperties.xml file.

Error Handling in PIP Queue Manager

Any failure in the processing of a Change Order is captured by the Queue Manager. This "Errored" process can be identified in the Queue Monitor, symbolized by⚠.

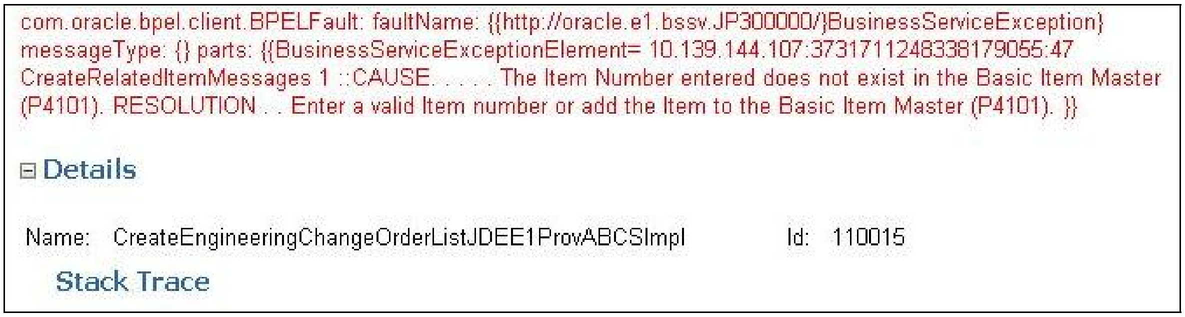
This is a sample screen showing Change Order process status:



		Reference	Change Number	Release Time	Processed Time	Process Status
☐	⚠	ATO0000063	ECO-000114	23-Jul-2009 14:01:51	23-Jul-2009 14:06:23	Errored
☐	▯	ATO0000064	ECO-000115	23-Jul-2009 14:27:21	23-Jul-2009 14:27:52	Pending
	✓	ATO0000057	ECO-000108	22-Jul-2009 15:48:38	22-Jul-2009 15:49:13	Completed
	✓	ATO0000056	ECO-000107	22-Jul-2009 15:39:25	22-Jul-2009 15:40:03	Completed

Sample Change Order Process Status page

To see the reason of error, click the 'Errored' link in the Process Status column. It pops-up an Error Message window, similar to the sample message given.



com.oracle.bpel.client.BPELFault: faultName: {{http://oracle.e1.bssv.JP300000/}BusinessServiceException} messageType: {} parts: {{BusinessServiceExceptionElement= 10.139.144.107:3731711248338179055:47 CreateRelatedItemMessages 1 ::CAUSE. . . . . The Item Number entered does not exist in the Basic Item Master (P4101). RESOLUTION . . Enter a valid Item number or add the Item to the Basic Item Master (P4101). }}

⊟ Details

Name: CreateEngineeringChangeOrderListJDEE1ProvABCSImpl Id: 110015

Stack Trace

Sample Error Message

This Error Message comprises of two parts:

1. Error Text - This is the text of error source, which can be from any participating ABCS that may have faulted.
2. Details - The 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 displayed in this UI. In addition to this, any failures in the flows would be captured in AIA's Error Logs. These can be seen from Enterprise Manager Console logs section.

In an event when a certain service is down and the error is not related to the payload, users may resubmit the change in the Queue Manager UI.

This table lists the errors that are thrown by the PIP services.

Error Code	Message Text
AIA_ERR_AIAAGILEJDEE1_0003	No error message has been specified for the input key
AIA_ERR_AIAAGILEJDEE1_0004	None of the Engineering Change Orders selected have been integrated

This table lists the errors that are thrown by Initial Load.

Error Code	Message Text
Invoke AIS Design	
AIA_ERR_AIAAGILEJDEE1_0013	File does not exist:
Release ECO Design	
AIA_ERR_AIAAGILEJDEE1_0010	Failed
AIA_ERR_AIAAGILEJDEE1_0011	Release ECO SDK Execution Failed:
AIA_ERR_AIAAGILEJDEE1_0012	Could not find Change Order workflow:
AIA_ERR_AIAAGILEJDEE1_0015	The Change Order Not found –
Initial Load BOM JDE E1 to Agile PLM design	
AIA_ERR_AIAAGILEJDEE1_0001	For additional information go to <SOA_HOME>/agile/ais/fileResult and review the following file:
AIA_ERR_AIAAGILEJDEE1_0006	All Bill Of Materials Batch Quantity should be equal to Zero.
AIA_ERR_AIAAGILEJDEE1_0007	All Bill Of Materials Type should be equal to 'DEFAULT_BOM_TYPE' AIA Configuration Property value.
AIA_ERR_AIAAGILEJDEE1_0008	Multiple Bill Of Materials for the same Parent Item is not allowed when 'MULTISITE_ENABLED' property is set to 'FALSE'
AIA_ERR_AIAAGILEJDEE1_0009	Bill Of Materials Components with 'Non-Stock' Stocking Type is not allowed when 'MULTISITE_ENABLED' property is set to 'TRUE'.
Initial Load Item JDE E1 to Agile PLM Design	
AIA_ERR_AIAAGILEJDEE1_0001	For additional information go to <SOA_HOME>/agile/ais/fileResult and review the following file:
AIA_ERR_AIAAGILEJDEE1_0002	There are Branch Plants in the input message that do not have a mapped value in the AGILE_TARGET_SITE_MAPPING DVM
AIA_ERR_AIAAGILEJDEE1_0014	None of the Items selected have associated Branch Plants
AIA_ERR_AIAAGILEJDEE1_0016	Invalid Item Revision Level: Make sure 'ITEM_REVISION_LEVEL' Property in AIA Configuration Properties is not Blank

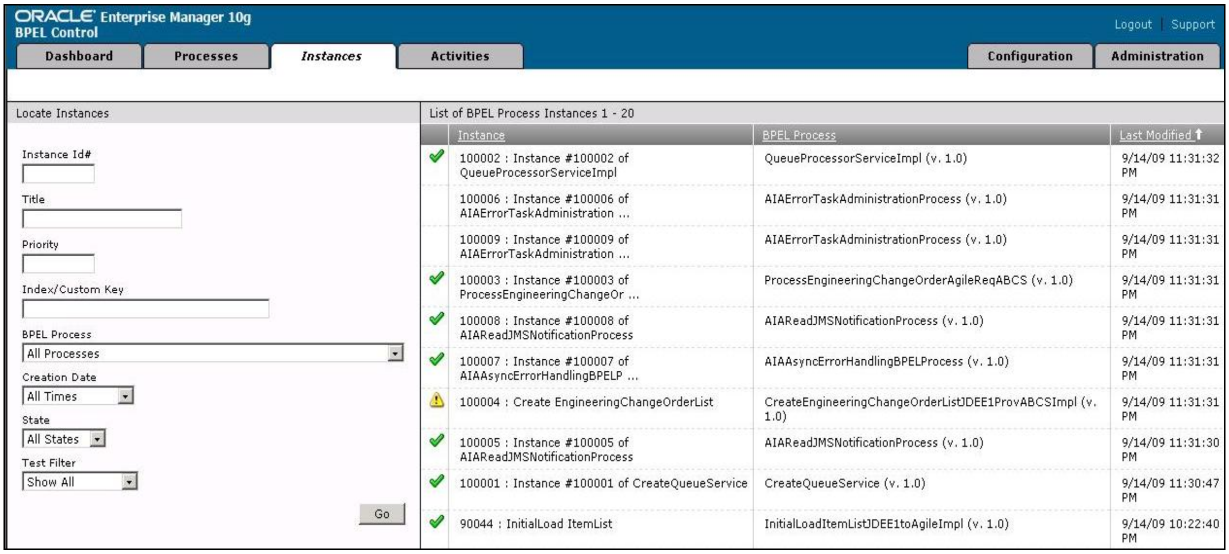
For more information about the errors thrown by Agile PLM or JD Edwards EBS, see the corresponding product’s documentation.

For more information about AIA error handling, see the *Oracle Application Integration Architecture – Foundation Pack: Core Infrastructure Components Guide*, "Setting Up and Using Error Handling and Logging".

Mechanism for Error Handling / Reporting

To view an error, follow these steps:

1. On the BPEL Console, the user needs to login with the user name and password.
2. Click on the Instances tab.
3. Click on the 'faulted' instance (the instance marked by the ⚠ icon).



Locate Instances

Instance Id#

Title

Priority

Index/Custom Key

BPEL Process: All Processes

Creation Date: All Times

State: All States

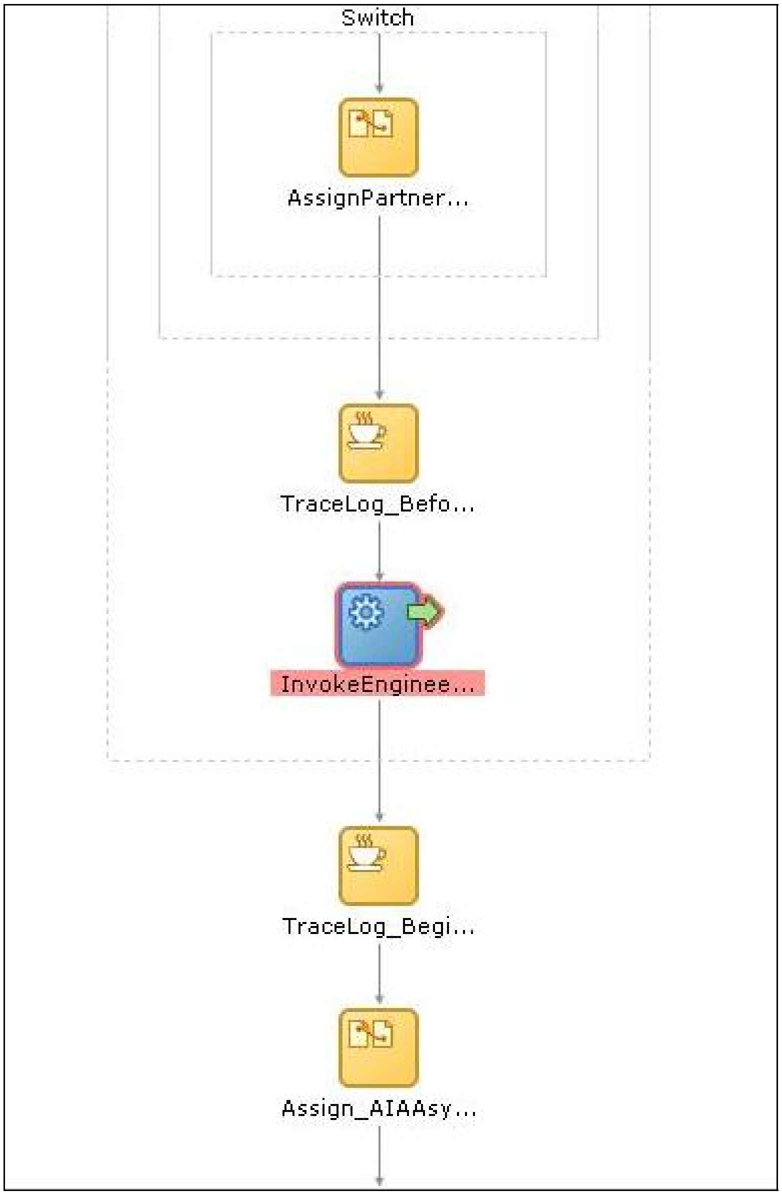
Test Filter: Show All

List of BPEL Process Instances 1 - 20

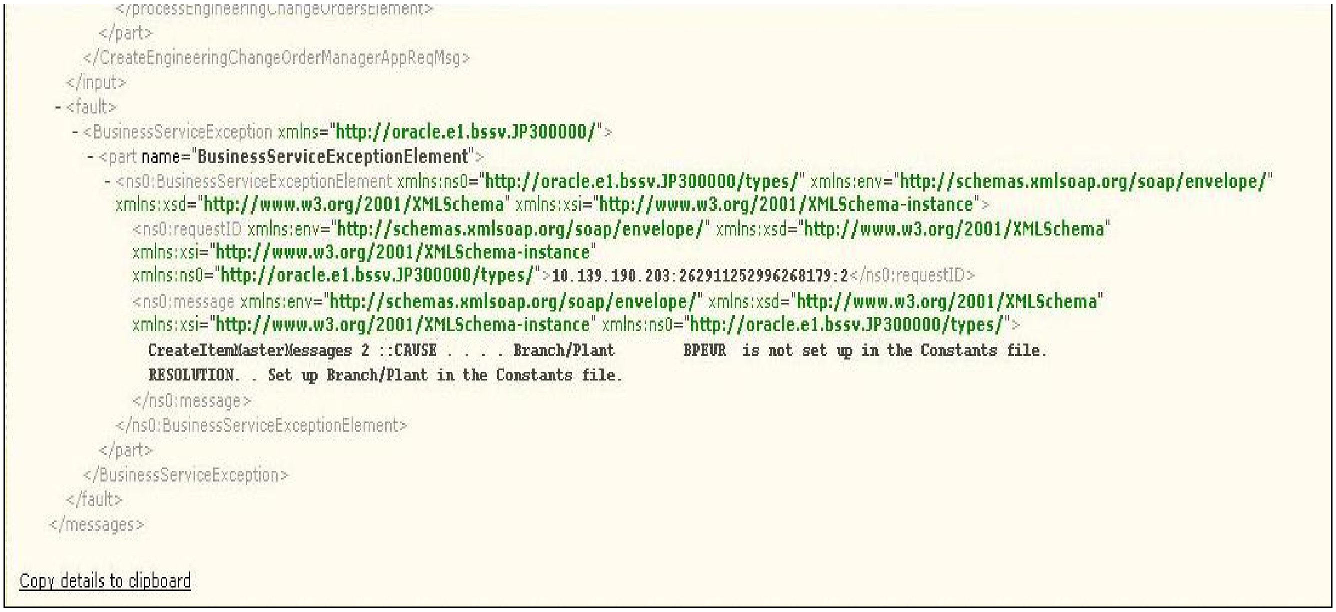
	Instance	BPEL Process	Last Modified
✓	100002 : Instance #100002 of QueueProcessorServiceImpl	QueueProcessorServiceImpl (v. 1.0)	9/14/09 11:31:32 PM
	100006 : Instance #100006 of AIAErrorTaskAdministration ...	AIAErrorTaskAdministrationProcess (v. 1.0)	9/14/09 11:31:31 PM
	100009 : Instance #100009 of AIAErrorTaskAdministration ...	AIAErrorTaskAdministrationProcess (v. 1.0)	9/14/09 11:31:31 PM
✓	100003 : Instance #100003 of ProcessEngineeringChangeOr ...	ProcessEngineeringChangeOrderAgileReqABCS (v. 1.0)	9/14/09 11:31:31 PM
✓	100008 : Instance #100008 of AIAReadJMSNotificationProcess	AIAReadJMSNotificationProcess (v. 1.0)	9/14/09 11:31:31 PM
✓	100007 : Instance #100007 of AIAAsyncErrorHandlingBPELP ...	AIAAsyncErrorHandlingBPELProcess (v. 1.0)	9/14/09 11:31:31 PM
⚠	100004 : Create EngineeringChangeOrderList	CreateEngineeringChangeOrderListJDEE1ProvABCSImpl (v. 1.0)	9/14/09 11:31:31 PM
✓	100005 : Instance #100005 of AIAReadJMSNotificationProcess	AIAReadJMSNotificationProcess (v. 1.0)	9/14/09 11:31:30 PM
✓	100001 : Instance #100001 of CreateQueueService	CreateQueueService (v. 1.0)	9/14/09 11:30:47 PM
✓	90044 : InitialLoad ItemList	InitialLoadItemListJDEE1toAgileImpl (v. 1.0)	9/14/09 10:22:40 PM

4. On the next page that appears, the details of the Fault are displayed. Click the Flow tab on this window.

   This displays the visual representation of the history of this BPEL business flow as shown below:



5. On the flow, click on the ‘red’ activity where the error has been caught. And that would bring up the ‘Activity Audit Trail’ which gives you details about the error thrown. This screen would look like the image below.



```
            </processEngineeringChangeOrdersElement>
          </part>
        </CreateEngineeringChangeOrderManagerAppReqMsg>
      </input>
    - <fault>
      - <BusinessServiceException xmlns="http://oracle.e1.bssv.JP300000/">
        - <part name="BusinessServiceExceptionElement">
          - <ns0:BusinessServiceExceptionElement xmlns:ns0="http://oracle.e1.bssv.JP300000/types/" xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
              <ns0:requestID xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xmlns:ns0="http://oracle.e1.bssv.JP300000/types/">10.139.190.203:262911252996268179:2</ns0:requestID>
              <ns0:message xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns0="http://oracle.e1.bssv.JP300000/types/">
                CreateItemMasterMessages 2 ::CAUSE . . . . Branch/Plant        BPEUR  is not set up in the Constants file.
                RESOLUTION. . Set up Branch/Plant in the Constants file.
              </ns0:message>
            </ns0:BusinessServiceExceptionElement>
          </part>
        </BusinessServiceException>
      </fault>
    </messages>
```

Viewing EBO Implementation Maps (EIMs)

Item EBO mappings

Agile PLM Entity: Attribute	Item EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	Name	Item: 2nd Item Number	
	Description	Item: Description Line1	
	ItemIdentification/BusinessComponentID	Item: Primary: Product Item Number (short)	
Part/Document. Title Block.Number	ItemIdentification/ID	Item: Primary: Product Item Number (short)	
	ItemIdentification/ContextID	Item: Branch/Plant	
	ItemIdentification/ApplicationObjectKey/ID	Item: SecondItem Number	
	ItemIdentification/ApplicationObjectKey/ContextID	Item: Branch/Plant	

Agile PLM Entity: Attribute	Item EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	ItemIdentification/AlternateObjectKey/ID	Item: Item Number (third)	
	ItemIdentification/Revision/Description	Item:Description Line 1	
	TypeCode	Item:StockingType	
	DualUOMTrackingIndicator	Item: Primary: Dual UOM Item	
	BaseUOMCode	Item: Primary: Primary Unit of Measure	
	SecondaryUOMCode	Item: Primary: Secondary Unit of Measure	
	ShippingUOMCode	Item : Shipping Unit Of Measure	
Change Orders Class.Affected Items.New Lifecycle	Status/Code	Item : Stocking Type	
	ItemSpecificationGroup/SpecificationGroup/Name	Text as "Category Codes"	
	ItemSpecificationGroup/SpecificationGroup/Specification/SalesReportingCode 1		
	ItemSpecificationGroup/SpecificationGroup/Specification/SalesReportingCode 2		

Agile PLM Entity: Attribute	Item EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	ItemSpecificationGroup/SpecificationGroup/Specification/SalesReportingCode3		
	ItemSpecificationGroup/SpecificationGroup/Specification/SalesReportingCode4		
	ItemSpecificationGroup/SpecificationGroup/Specification/SalesReportingCode5		
	ItemSpecificationGroup/SpecificationGroup/Specification/SalesReportingCode6		
	ItemSpecificationGroup/SpecificationGroup/Specification/SalesReportingCode7		
	ItemSpecificationGroup/SpecificationGroup/Specification/SalesReportingCode8		
	ItemSpecificationGroup/SpecificationGroup/Specification/SalesReportingCode9		

Agile PLM Entity: Attribute	Item EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	ItemSpecificationGroup/SpecificationGroup/Specification/SalesReportingCode10		
	ItemSpecificationGroup/SpecificationGroup/Specification/PurchasingReportingCode1		
	ItemSpecificationGroup/SpecificationGroup/Specification/PurchasingReportingCode2		
	ItemSpecificationGroup/SpecificationGroup/Specification/PurchasingReportingCode3		
	ItemSpecificationGroup/SpecificationGroup/Specification/PurchasingReportingCode4		
	ItemSpecificationGroup/SpecificationGroup/Specification/PurchasingReportingCode5		

Agile PLM Entity: Attribute	Item EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	ItemSpecificationGroup/SpecificationGroup/Specification/PurchasingReportingCode6		
	ItemSpecificationGroup/SpecificationGroup/Specification/PurchasingReportingCode7		
	ItemSpecificationGroup/SpecificationGroup/Specification/PurchasingReportingCode8		
	ItemSpecificationGroup/SpecificationGroup/Specification/PurchasingReportingCode9		
	ItemSpecificationGroup/SpecificationGroup/Specification/PurchasingReportingCode10		
	ItemPhysicalCharacteristics/VolumeMeasure(unitCode)	Item: Physical Attributes:Volume Unit Of Measure	
	ItemPhysicalCharacteristics/WeightMeasure(unitCode)	Item: Physical Attributes:Weight Unit Of Measure	

Agile PLM Entity: Attribute	Item EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	InventoryCharacteristics/SerializationEventCode	Item: Inventory: Serial Number Required	
	InventoryCharacteristics/ShelfLifeDuration	Item: Inventory: Shelf LifeDays	
	InventoryCharacteristics/StockingAllowedIndicator	Item: Inventory:Stock Non Stock	
	InventoryCharacteristics/UnitCost/Amount(currencyCode)	Item: Unit Cost	
	ItemPurchasingCharacteristics/DebitGLAccountCode	Item: Purchasing: G/L Class Code	
	ItemPurchasingCharacteristics/PurchasingAllowedIndicator	Item: Purchasing : Stocking Type	
	ItemPurchasingCharacteristics/TaxableIndicator	Item: Purchasing: Purchasing Sales Taxable (Y/N)	
	ItemPurchasingCharacteristics/IssueUOMCode	Item: Purchasing: Component UOM	
	ItemPlanningCharacteristics/LotSizeMultiplier	Item: Multiple Order Quantity	
	ItemPlanningCharacteristics/InventoryPlanningCode	Item: General PlanningType	
Make/Buy	ItemPlanningCharacteristics/MakeOrBuyCode	Item: Stocking Type	

Agile PLM Entity: Attribute	Item EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	ItemPlanningCharacteristics/ReplenishmentSourceCode	Item: General Planning:Policy Code	
	ItemPlanningCharacteristics/ShrinkageRate	Item: MPS/MRP Planning:Shrinkage Rate	
	ItemPlanningCharacteristics/ReorderCharacteristics/ReorderQuantity	Item: Reorder Quantity	
	ItemPlanningCharacteristics/ReorderCharacteristics/MaximumReorderQuantity	Item: Maximum Reorder Quantity	
	ItemPlanningCharacteristics/ReorderCharacteristics/MinimumInventoryQuantity	Item: Minimum Reorder Quantity	
	ItemManufacturingCharacteristics/StructureAllowedIndicator	Item: Stocking Type	
	ItemOrderManagementCharacteristics/BackOrderEnabledIndicator	Item: Backorder allowed (y/n)	
	ItemOrderManagementCharacteristics/InvoicingEnabledIndicator	Item:StockingType	

Agile PLM Entity: Attribute	Item EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
Site	ItemIdentification/ContextID (schemeID: OrganizationCode)		
Part/Document. Title Block. Description	ItemIdentification/Description		
Change Orders Class.Affected Items.New Revision	ItemIdentification/Revision/Code		
ECO / SCO. Affected Item. Effectivity Date	ItemIdentification/Revision/EffectiveDate		
Item Type	PrimaryClassificationCode		
Cost	ItemManufacturingCharacteristics/CostingEnabledIndicator		
Shippable Item	ItemOrderManagementCharacteristics/ShippableIndicator		
Commodity, Item Category, Product Line(s), Part Family	ItemClassification/ClassificationCode		

Agile PLM Entity: Attribute	Item EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
Part/Document. Manufacturers. Manufacturer	ItemManufacturer/ManufacturerPartyReference/OrganizationName		
Part/Document. Manufacturers. Mfr Part Number	ItemIdentification/ManufacturerItemID (SchemeAgencyID: <Manufacturer Name>)		

Item Balance EBO mappings

This table describes the Item Balance EBO mappings:

Agile PLM Entity: Attribute	Item Balance EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	InventoryBalance/ItemReference/ItemIdentification/BusinessComponentID	Item.IdentifierShortItem	
	ItemIdentification/ApplicationObjectKey/ID	Item.IdentifierShortItem	
Site	InventoryBalance/ItemReference/ItemIdentification/ContextID	Item.BranchPlant	
	InventoryBalance/OnHandQuantity	Item.On Hand Quantity	
	InventoryBalance/AvailableQuantity	Item.Available Quantity	
	InventoryBalance/ReservedQuantity	Item.Reserved Quantity	
Part/Document. Title Block.Number	InventoryBalance/ItemReference/ Identification/ID		

Engineering Change Order EBO Mappings

This table describes the Engineering Change Order EBO mappings:

Agile PLM Entity: Attribute	ECO EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	Identification.BusinessComponentID	EngineeringChangeOrder. ECO Number	
Number	Identification.ID	EngineeringChangeOrder. ECO Number	
	Identification.ApplicationObjectKey.ID	EngineeringChangeOrder. ECO Number	
	Identification/AlternateObjectKeyID	EngineeringChangeOrder. ECO Number	
	Identification.ApplicationObjectKey.ContextID	EngineeringChangeOrder. BranchPlant	
Site	Identification/ContextID (schemeID: OrganizationCode)	EngineeringChangeOrder. BranchPlant	
Description of Change	Description	EngineeringChangeOrder. ECO Description/ StandardDescription	
Date Originated	InitiationDate	EngineeringChangeOrder. Actual Design Date	
Date Released	ImplementationDate	EngineeringChangeOrder. Actual Incorporated Date	
	PriorityCode	EngineeringChangeOrder. Priority	
Reason Code	ReasonCode	EngineeringChangeOrder. Reason	
Change Type	TypeCode	EngineeringChangeOrder. ECO Type	
Change Category	ClassificationCode	EngineeringChangeOrder. PhaseIn	
Status	Status.Code	EngineeringChangeOrder. ECO Status	

Agile PLM Entity: Attribute	ECO EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
Disposition01	EngineeringChangeOrderLine/DispositionTypeCode	EngineeringChangeOrder. Existing Disposition	
Effective Date	EngineeringChangeOrderLine/EffectiveDate	EngineeringChangeOrder. Part list effective from	
Revised Item	EngineeringChangeOrderLine/RevisedItem	EngineeringChangeOrder. Revised Item	
	EngineeringChangeOrderLine/RevisedItem/Item/ItemId	EngineeringChangeOrder. Items. itemId	
	ItemIdentification/ApplicationObjectKey/Id	EngineeringChangeOrder. Items. itemCatalog	
	ItemIdentification/Id	EngineeringChangeOrder. Items. itemProduct	
	ItemClassification/CatalogReference/CatalogIdentification/ID	EngineeringChangeOrder. Items. itemCatalog	
	BaseUOMCode	EngineeringChangeOrder. ItemDimensions.unitOfMeasureCodePrimary	
	PhysicalCharacteristics/WeightMeasure(unitCode)	EngineeringChangeOrder. ItemDimensions.unitOfMeasureCodeWeight	

Agile PLM Entity: Attribute	ECO EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	PhysicalCharacteristics/VolumeMeasure(unitCode)	EngineeringChangeOrder.ItemDimensions.unitOfMeasureCodeVolume	
	ItemIdentification/Description	EngineeringChangeOrder.ItemDimensions.description1	
	ItemIdentification/Description	EngineeringChangeOrder.ItemDimensions.description2	
	InventoryCharacteristics/StockingAllowedIndicator	EngineeringChangeOrder.ItemDimensions.stockingTypeCode	
	ItemPurchasingCharacteristics/DebitGLAccountCode	EngineeringChangeOrder.ItemDimensions.glClassCode	
	InventoryCharacteristics/SerializationEventCode	EngineeringChangeOrder.ItemDimensions.serialNumberFlag	
	LotControlIndicator	EngineeringChangeOrder.ItemDimensions.lotStatusCode	

Agile PLM Entity: Attribute	ECO EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	InventoryCharacteristics/ShelfLifeDuration	EngineeringChangeOrder.ItemDimensions.daysShelfLife	
	ItemIdentification/ContextID	EngineeringChangeOrder.ItemBranch.branchPlant	
	InventoryCharacteristics/StockingAllowedIndicator	EngineeringChangeOrder.ItemBranch.stockingTypeCode	
	ItemPurchasingCharacteristics/DebitGLAccountCode	EngineeringChangeOrder.ItemBranch.glClassCode	
	InventoryCharacteristics/SerializationEventCode	EngineeringChangeOrder.ItemBranch.serialNumberFlag	
	Lot Process Type	EngineeringChangeOrder.ItemBranch.lotProcessCode	

Agile PLM Entity: Attribute	ECO EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	Lot Status Code	EngineeringChangeOrder.ItemBranch.lotStatusCode	
	InventoryCharacteristics/ShelfLifeDuration	EngineeringChangeOrder.ItemBranch.daysShelfLife	
	ItemIdentification/ContextID	EngineeringChangeOrder.PartsList.ItemBranchPlant	
	ItemIdentification/Revision/Number (schemeID: RevisionId)	EngineeringChangeOrder.PartsList.swapRevisionLevel	
	ItemManufacturer/EffectiveTimePeriod/StartDateTime	EngineeringChangeOrder.PartsList.datePartsListEffectiveFrom	
	ItemManufacturer/EffectiveTimePeriod/EndDateTime	EngineeringChangeOrder.PartsList.datePartsListEffectiveThru	
	ItemPlanningCharacteristics/MaximumProductionOrderQuantity	EngineeringChangeOrder.PartsList.batchQuantity	
	BaseUOMCode	EngineeringChangeOrder.PartsList.unitOfMeassureAsInput	

Agile PLM Entity: Attribute	ECO EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	ItemIdentification/Revision/Code	EngineeringChangeOrder.PartsList.revisionLevelFrom	
	ItemIdentification/Revision/Code	EngineeringChangeOrder.PartsList.revisionLevelTo	
	EngineeringChangeOrderLine/RevisedBillOfMaterial	Revised BOM/Structure	
	EngineeringChangeOrderLine/RevisedBillOfMaterial/ ContextID (schemeID: OrganizationCode)	EngineeringChangeOrder.relatedItems.relatedItemBranchPlant	
	EngineeringChangeOrderReference/EngineeringChangeOrderIdentification/ID	EngineeringChangeOrder.relatedItems.ECO Number	
	EngineeringChangeOrderLine/RevisedBillOfMaterial/ EffectiveTimePeriod/StartDateTime	EngineeringChangeOrder.relatedItems.dateRelatedItemEffectiveFrom	
	EngineeringChangeOrderLine/RevisedBillOfMaterial/ EffectiveTimePeriod/StartDateTime	EngineeringChangeOrder.relatedItems.dateRelatedItemEffectiveThru	
	EngineeringChangeOrderLine/RevisedBillOfMaterial/Quantity	EngineeringChangeOrder.relatedItems.relatedItemBatchQuantity	
	EngineeringChangeOrderLine/RevisedBillOfMaterial/StorageUnitCode	EngineeringChangeOrder.relatedItems.relatedItemBatchUnitOfMeassure	
	EngineeringChangeOrderLine/RevisedBillOfMaterial/TypeCode	EngineeringChangeOrder.relatedItems.RelatedItemBOMType	
	ItemReference/Identification/Revision/Code	EngineeringChangeOrder.relatedItems.relatedItemFromRevisionLevel	

Agile PLM Entity: Attribute	ECO EBO	JDE E1 Entity: Attribute Group: Attribute	Comments
	ItemReference/Identification/Revision/Code	EngineeringChangeOrder.relatedItems.relatedItemToRevisionLevel	
	ComponentItem/Quantity	EngineeringChangeOrder.relatedItems.quantity	
	ComponentItem/StorageUnitCode	EngineeringChangeOrder.relatedItems.quantityUOM	
	ItemReference/Identification/Revision/Code	EngineeringChangeOrder.relatedItems.relatedItemDrawingToRevisionLevel	
	CommonBillOfMaterialsReference/ ItemIdentification/ID	EngineeringChangeOrder.relatedItems.ItemNumber	
Originator	RequesterPartyReference/ PersonName		
Change Analyst, Component Engineer	OwnerPartyReference/ PersonName		
Flex Attributes	EngineeringChangeOrderLine / EngineeringChangeOrderLineSpecificationGroup/		
Revised BOM/Structure	EngineeringChangeOrderLine / RevisedBillOfMaterials		

Initial Load Mappings

This table describes the initial load mappings for Item:

JDE E1 Entity: Attribute	Agile PLM Entity: Attribute	Comments
Item.IdentifierShortItem	AgileData.Parts.TitleBlock.Number	if AIA Config property directs to use Short Item # in Agile PLM

JDE E1 Entity: Attribute	Agile PLM Entity: Attribute	Comments
Item.Identifier2ndItem	AgileData.Parts.TitleBlock.Number	if AIA Config property directs to use 2nd Item # in Agile PLM
Item.Identifier3rdItem	AgileData.Parts.TitleBlock.Number	if AIA Config property directs to use 3rd Item # in Agile PLM
Item.DescriptionLine1	AgileData.Parts.TitleBlock.Description	
Item.StockingType	AgileData.Parts.TitleBlock.LifecyclePhase	
Item.BranchPlant	AgileData.Parts.Sites.SiteName	If Multisite is enabled
Item.BranchPlant	AgileData.Parts.PageTwo.Multilist01.value	If Multisite is not enabled

This table describes the initial load mappings for Bill of Materials:

JDE E1 Entity: Attribute	Agile PLM Entity: Attribute	Comments
Item.IdentifierShortItem	AgileData.Parts.TitleBlock.Number	Agile Item Number is retrieved from ITEM_ITEMID XREF with reference to JDE E1 Short Item Number
Item.RevisionLevel	AgileData.Parts.TitleBlock.Rev	If JDE E1 attribute is blank, value from PARENT_ITEM_REVISION_LEVEL AIA Config property is mapped
Item.BOM.Components.IdentifierShortItem	AgileData.Parts.BOM.ItemNumber	Agile Item Number is retrieved from ITEM_ITEMID XREF with reference to JDE E1 Short Item Number
Item.BOM.Components.Branch	AgileData.Parts.BOM.Sites	Mapped only when Multisite is enabled and retrieved from ITEM_ITEMID XREF with reference to JDE E1 Branch
Item.BOM.Components.Quantity	AgileData.Parts.BOM.Qty	

For more information about how services are mapped, see the My Oracle Support document *EBO Implementation Maps (EIMs) 881022.1*.

Setting Configuration Properties

This PIP uses various configuration parameters that control the behavior of the flow. Standard AIA XML configuration file, AIAConfigurationProperties.xml, is used 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 PIPs running on the SOA suite. Service level parameters can be configured at the individual service level such as ABCS.

Note. The configuration properties from the Agile PLM Module and the JD Edwards EnterpriseOne Module are listed separately in this section, only for the purpose of identification. The actual AIAConfigurationProperties.xml file on AIA Server is a merger of both.

Configuration Parameters

This PIP uses the following type of 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.

Note. Whenever the AIAConfigurationProperties.xml file is updated, the file should be reloaded for updates to be reflected in the applications or services that use the updated properties. You can perform this reload by clicking the Reload button on the Configuration page in the Oracle AIA Console. Alternatively, you can perform the reload by rebooting the server.

For more information about updating SOA MDS with AIA metadata, see *Oracle Application Integration Architecture – Foundation Pack: Development Guide*, "How to Set Up AIA Workstation", Updating SOA MDS with AIA MetaData.

Properties	(default) Value/Setting	Description
moduleName	Agile PLM	
MULTISITE_ENABLED	TRUE	When set to True, the sites specified in Sites Tab of Items are used to determine the Orgs in JDE E1 to which they are mapped.

Properties	(default) Value/Setting	Description
		When set to False, Page2 Multilist01attribute is used to determine the Orgs in JDE E1 to which the Item will be extended to.
Item.UnitCostAttribute	Site.Numeric01	Determines the attribute to which the unit cost from JDE E1 would be updated in Agile PLM.
Item.AvailableQuantityAttribute	Site.Numeric02	Determines the attribute to which the available quantity from JDE E1 would be updated in Agile PLM.
Item.OnHandQuantityAttribute	Site.Numeric03	Determines the attribute to which the on-hand quantity from JDE E1 would be updated in Agile PLM.
Item.ReservedQuantityAttribute	Site.Numeric04	Determines the attribute to which the reserved quantity from JDE E1 would be updated in Agile PLM.
REPLICATE_BOM_ENABLED	FALSE	Used for sample replicate BOM customization. Refer Readme in Samples folder
COMMON_BOM_ENABLED	FALSE	Used for sample common BOM customization. Refer Readme in Samples folder
serviceName	CreateQueueService	
TRACE.LOG.ENABLED	false	Use tracelog for the flow
serviceName	QueueProcessorServiceImpl	
TRACE.LOG.ENABLED	false	Use tracelog for the flow
serviceName	UpdateEngineeringChangeOrderListAgileProvABCSImpl	
Default.SystemID	AGILE_01	System ID of Agile PLM application instance
Routing.MergeABSService.RouteToCAVS	FALSE	If set to true route to CAVS else route to Agile PLM application. This invocation would be with respect to Change merge ABS service
Routing.MergeABSService.AGILE_01.EndpointURI	http://${http.hostname}:${http.port}/AIAComponents/ApplicationObjectLibrary/Agile/V1/wsdls/ItemABS.wsdl	Agile PLM change merges ABS service URL. When RouteToCAVS property is set to false use the URL mentioned to connect to Agile PLM application This invocation would be with respect to Change merge ABS

Properties	(default) Value/Setting	Description
		service
Routing.MergeABSService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS SOAP URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to Change merge ABS service
Routing.ChangeABSService.RouteToCAVS	FALSE	If set to true route to CAVS else route to Agile PLM application. This invocation would be with respect to Change ABS service
Routing.ChangeABSService.AGILE_01.EndpointURI	http://${http.hostname}:${http.port}/AIAComponents/ApplicationObjectLibrary/Agile/V1/wsdls/ItemABS.wsdl	Agile PLM change ABS service URL. When RouteToCAVS property is set to false use the URL mentioned to connect to Agile PLM application This invocation would be with respect to Change ABS service
Routing.ChangeABSService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS SOAP URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to Change ABS service
Routing.ChangeStatusService.RouteToCAVS	FALSE	If set to true route to CAVS else route to Agile PLM application. This invocation would be with respect to Change Status ABS service
Routing.ChangeStatusService.AGILE_01.EndpointURI	http://${http.hostname}:${http.port}/AIAComponents/ApplicationObjectLibrary/Agile/V1/wsdls/ItemABS.wsdl	Agile PLM change status ABS service URL. When RouteToCAVS property is set to false use the URL mentioned to connect to Agile PLM application This invocation would be with respect to Change Status ABS service
Routing.ChangeStatusService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to Change Status ABS service
Routing.EngineeringChangeOrderResponseEBS.RouteToCAVS	FALSE	If set to true route to CAVS else route to Agile PLM application. This invocation would be with respect to ECO response EBS
Routing.EngineeringChangeOrderResponseEBS.UpdateEngineeringChang	http://${http.hostname}:${http.port}/event/AIASystem/EBS/ItemBalanceRes	ECO response EBS SOAP URL. When RouteToCAVS property is set

Properties	(default) Value/Setting	Description
eOrderListResponse.EndpointURI	ponseEBS	to false use the URL mentioned to connect to ECO response EBS. This invocation would be with respect to ECO response EBS
Routing.EngineeringChangeOrderResponseEBS.UpdateEngineeringChangeOrderListResponse.CAVS.EndpointURI	http://${http.hostname}:${http.port}/event/AIASystem/EBS/ItemBalanceResponseEBS	CAVS SOAP URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to ECO response EBS
Routing.EngineeringChangeOrderResponseEBS.UpdateEngineeringChangeOrderListResponse.MessageProcessingInstruction.EnvironmentCode		Environment code like 'PRODUCTION'/ 'CAVS' etc. Identifies the installation environment
ABCSExtension.PreProcessABM	FALSE	User exit for the pre-process ABM should be called or not
ABCSExtension.PreProcessEBM	FALSE	User exit for the pre-process EBM should be called or not
ABCSExtension.PostProcessEBM	FALSE	User exit for the post-process EBM should be called or not
ABCSExtension.PostProcessABM	FALSE	User exit for the post-process ABM should be called or not
TRACE.LOG.ENABLED	false	Use tracelog for the flow
ROUTE_TO_CAVS	false	Route to CAVS, if set as True
DEFAULT_TARGET_ENDPOINT_URI	http://<host>:<port>/event/CoreAgile/EngineeringChangeOrderResponseEBS	EngineeringChangeOrderResponseEBS runtime target endpoint URI
serviceName	UpdateItemBalanceListAgileProvABCSImpl	
Default.SystemID	AGILE_01	System ID of Agile PLM application instance
Routing.ItemABSService.RouteToCAVS	FALSE	If set to true route to CAVS else route to Agile PLM application. This invocation would be with respect to Item ABS
Routing.ItemABSService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator Routing.ItemBalanceResponseEBS.UpdateItemBalanceListResponse.RouteToCAVS	CAVS URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to Item ABS
Routing.ItemBalanceResponseEBS.UpdateItemBalanceListResponse.Endp	http://${http.hostname}:${http.port}/event/AIASystem/EBS/ItemBalanceRes	Item balance response EBS SOAP URL. When RouteToCAVS property is

Properties	(default) Value/Setting	Description
ointURI	ponseEBS	set to false use the URL mentioned to connect to Item balance response EBS. This invocation would be with respect to item balance response EBS
Routing.ItemBalanceResponseEBS.UpdateItemBalanceListResponse.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncresponsesimulator	CAVS URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to Itembalance response EBS
Routing.ItemBalanceResponseEBS.UpdateItemBalanceListResponse.MessageProcessingInstruction.EnvironmentCode		Environment code like 'PRODUCTION'/ 'CAVS' etc. Identifies the installation environment
ABCSExtension.PreProcessABM	FALSE	User exit for the pre-process ABM should be called or not
ABCSExtension.PreProcessEBM	FALSE	User exit for the pre-process EBM should be called or not
ABCSExtension.PostProcessEBM	FALSE	User exit for the post-process EBM should be called or not
ABCSExtension.PostProcessABM	FALSE	User exit for the post-process ABM should be called or not
TRACE.LOG.ENABLED	FALSE	Use tracelog for the flow
ROUTE_TO_CAVS	FALSE	Route to CAVS, if set as True
DEFAULT_TARGET_ENDPOINT_URI	http://<host>:<port>/event/CoreAgile/ItemBalanceResponseEBS	ItemBalanceResponseEBS runtime target endpoint URI
serviceName	UpdateItemListAgileProvABCSImpl	
Default.SystemID	AGILE_01	System ID of Agile PLM application instance
Routing.ItemResponseEBS.UpdateItemListResponse.MessageProcessingInstruction.EnvironmentCode		Environment code like 'PRODUCTION'/ 'CAVS' etc. Identifies the installation environment
Routing.ItemABSService.RouteToCAVS	FALSE	If set to true route to CAVS else route to Agile PLM application. This invocation would be with respect to Agile Item ABS service
Routing.ItemABSService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	Agile Item ABS service SOAP URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation

Properties	(default) Value/Setting	Description
		would be with respect to Agile Item ABS service
Routing.ItemResponseEBS.UpdateItemListResponse.RouteToCAVS	FALSE	If set to true route to CAVS else route to Item EBS. This invocation would be with respect to Item EBS
Routing.ItemResponseEBS.UpdateItemListResponse.EndpointURI	http://${http.hostname}:${http.port}/event/AIASystem/EBS/ItemResponseEBS	Item response EBS SOAP URL. When RouteToCAVS property is set to false use the URL mentioned to connect to Item response EBS. This invocation would be with respect to item response EBS
Routing.ItemResponseEBS.UpdateItemListResponse.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncresponsesimulator	CAVS URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to Item response EBS
ABCSExtension.PreProcessABM	FALSE	User exit for the pre-process ABM should be called or not
ABCSExtension.PreProcessEBM	FALSE	User exit for the pre-process EBM should be called or not
ABCSExtension.PostProcessEBM	FALSE	User exit for the post-process EBM should be called or not
ABCSExtension.PostProcessABM	FALSE	User exit for the post-process ABM should be called or not
TRACE.LOG.ENABLED	FALSE	Use tracelog for the flow
ROUTE_TO_CAVS	FALSE	Route to CAVS, if set as True
DEFAULT_TARGET_ENDPOINT_URI	http://localhost:8888/event/CoreAgile/ItemResponseEBS	ItemResponseEBS runtime target endpoint URI
serviceName	ProcessEngineeringChangeOrderAgileReqABCS	
Default.SystemID	AGILE_01	System ID of Agile PLM application instance
Routing.EngineeringChangeOrderEBS.CreateEngineeringChangeOrderList.RouteToCAVS	FALSE	If set to true route to CAVS else route to EngineeringChangeOrder EBS (Enterprise Business Service)
Routing.EngineeringChangeOrderEBS.CreateEngineeringChangeOrderList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncresponsesimulator	CAVS URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS
Routing.EngineeringChangeOrderEBS.CreateEngineeringChangeOrderList	PRODUCTION	Environment code like 'PRODUCTION'/ 'CAVS' etc. Identifies

Properties	(default) Value/Setting	Description
.MessageProcessingInstruction.EnvironmentCode		the installation environment
ABCSExtension.PreProcessABM	FALSE	User exit for the pre-process ABM should be called or not
ABCSExtension.PreProcessEBM	FALSE	User exit for the pre-process EBM should be called or not
ABCSExtension.PostProcessEBM	FALSE	User exit for the post-process EBM should be called or not
ABCSExtension.PostProcessABM	FALSE	User exit for the post-process ABM should be called or not
TRACE.LOG.ENABLED	FALSE	Use tracelog for the flow
ROUTE_TO_CAVS	FALSE	Route to CAVS, if set as True
DEFAULT_TARGET_ENDPOINT_URI	http://<host>:<port>/event/CoreAgile/EngineeringChangeOrderEBS	EngineeringChangeOrderEBS runtime target endpoint URI
Default.SystemID	AGILE_01	System ID of Agile PLM application instance
Routing.ItemEBS.CreateItem.RouteToCAVS	FALSE	If set to true route to CAVS else route to Item EBS (Enterprise Business Service)
Routing.ItemEBS.CreateItem.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncresponsesimulator	CAVS SOAP URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS
Routing.ItemEBS.CreateItem.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Environment code like 'PRODUCTION'/ 'CAVS' etc. Identifies the installation environment

Properties	(default) Value/Setting	Description
moduleName	JDE E1	
DEFAULT_BOM_TYPE	M	
INITIALLOAD.DEFAULT.SOURCE	JDEE1_01	System ID of JDE E1 application instance
INITIALLOAD.DEFAULT.TARGET	AGILE_01	System ID of Agile PLM application instance
AGILE_USERNAME	admin	Username used for invoking an Agile Integration Service (AIS) operation or creating a session for Agile PLM

Properties	(default) Value/Setting	Description
moduleName	JDE E1	
		Server Instance while using Agile PLM SDK.
AGILE_PASSWORD_ENCRYPTED	TCnFqWTinW4=	Encrypted password used for invoking an Agile Integration Service (AIS) operation or creating a session for Agile PLM Server Instance while using Agile SDK. Note: Since this is an encrypted password, it will be automatically populated during PIP Installation. If required, this needs to be populated with the ‘agile.password.encrypted’ value from ‘deploy.properties’ file in AIA_HOME.
AGILE_CHANGE_TYPE	ECO	The *ChangeType* option used by AIS when importing items/BOM in the Redline mode. This is the subclass name of the change order for the ECO.
AGILE_CHANGE_WORKFLOW	JDE E1 Initial Load Change Orders	The name of the change order workflow used to advance the change order directly from Pending to Released status. Note: This workflow name should match with the workflow created in Chapter 7 - Setting Up Agile PLM Applications, 7.Create New Workflow for Initial Load Change Orders section.
Properties	**(default) Value/Setting**	**Description**
serviceName	CreateEngineeringChangeOrderListJDEE1ProvABCSImpl	
Default.SystemID	JDEE1_01	System ID of JDE application instance
Routing.EngineeringChangeOrderManager.RouteToCAVS	false	If set to true route to CAVS else route to JDE application. This invocation would be with respect to EngineeringChangeOrderManager ABS service
Routing.EngineeringChangeOrderResponseEBS.RouteToCAVS	false	If set to true route to CAVS else route to ECO Response EBS. This invocation would be with respect to ECO response EBS
Routing.EngineeringChangeOrderMan	http://${http.hostname}:${http.port}// AIAValidationSystemServlet/syncres	CAVS SOAP URL. When RouteToCAVS property is set to true,

Properties	(default) Value/Setting	Description
moduleName	JDE E1	
ager.CAVS.EndpointURI	ponsesimulator	use the URL mentioned to connect to CAVS. This invocation would be with respect to EngineeringChangeOrderManager ABS service
Routing.EngineeringChangeOrderResponseEBS.CAVS.EndpointURI	http://${http.hostname}:${http.port}//AIAValidationSystemServlet/asyncresponsesimulator	CAVS SOAP URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to ECO response EBS
Routing.EngineeringChangeOrderManager.JDEE1_01.EndpointURI	http://${http.hostname}:${http.port}//PY900/EngineeringChangeOrderManager	JDE EngineeringChangeOrderManager ABS service URL. When RouteToCAVS property is set to false use the URL mentioned to connect to JDE application This invocation would be with respect to EngineeringChangeOrderManager ABS service
ABCSExtension.PreProcessABM	false	User exit for the pre-process ABM should be called or not
ABCSExtension.PostProcessABM	false	User exit for the pre-process EBM should be called or not
ABCSExtension.PreProcessEBM	false	User exit for the post-process EBM should be called or not
ABCSExtension.PostProcessEBM	false	User exit for the post-process ABM should be called or not
Trace.Log.Enabled	false	Use tracelog for the flow
Routing.EngineeringChangeOrderResponseEBS.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Environment code like 'PRODUCTION'/ 'CAVS' etc. Identifies the installation environment
ECO_Implemented_Status	E5	
Default_ECO_Branch_Plant		
serviceName	UpdateEngineeringChangeOrderListJDEE1ReqABCSImpl	
Default.SystemID	JDEE1_01	System ID of JDE application instance
ABCSExtension.PreXformABMtoEBM	false	User exit for the pre-process ABM should be called or not
ABCSExtension.PostXformABMtoEBM	false	User exit for the pre-process EBM

Properties	(default) Value/Setting	Description
moduleName	JDE E1	
		should be called or not
ABCSExtension.PreInvokeEBS	false	User exit for the post-process EBM should be called or not
ABCSExtension.PostInvokeEBS	false	User exit for the post-process ABM should be called or not
Routing.TimeStampManager.RouteToCAVS	false	If set to true route to CAVS else route to JDE application. This invocation would be with respect to IntegrationTimeStampManager ABS service
Routing.TimeStampManager.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS SOAP URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to IntegrationTimeStampManager ABS service
Routing.EngineeringChangeOrderEBS.UpdateEngineeringChangeOrderList.RouteToCAVS	false	If set to true route to CAVS else route to ECO EBS. This invocation would be with respect to ECO EBS
Routing.EngineeringChangeOrderEBS.UpdateEngineeringChangeOrderList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncresponsesimulator	CAVS SOAP URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to ECO EBS
Routing.EngineeringChangeOrderEBS.UpdateEngineeringChangeOrderList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Environment code like 'PRODUCTION'/ 'CAVS' etc. Identifies the installation environment
Routing.TimeStampManager.JDEE1_01.EndpointURI	http://${http.hostname}:${http.port}/<Env>/IntegrationTimeStampManager	JDE IntegrationTimeStampManager ABS service URL. When RouteToCAVS property is set to false use the URL mentioned to connect to JDE application This invocation would be with respect to IntegrationTimeStampManager ABS service
TRACE.LOG.ENABLED	false	Use tracelog for the flow
serviceName	UpdateItemBalanceListJDEE1ReqABCSImpl	
Default.SystemID	JDEE1_01	System ID of JDE application instance
ABCSExtension.PreXformABMtoEBM	false	User exit for the pre-process ABM

Properties	(default) Value/Setting	Description
moduleName	JDE E1	
		should be called or not
ABCSExtension.PostXformABMtoEBM	false	User exit for the pre-process EBM should be called or not
ABCSExtension.PreInvokeEBS	false	User exit for the post-process EBM should be called or not
ABCSExtension.PostInvokeEBS	false	User exit for the post-process ABM should be called or not
Routing.TimeStampManager.RouteToCAVS	false	If set to true route to CAVS else route to JDE application. This invocation would be with respect to IntegrationTimeStampManager ABS service
Routing.TimeStampManager.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS SOAP URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to IntegrationTimeStampManager ABS service
Routing.ItemBalanceEBS.UpdateItemBalanceList.RouteToCAVS	FALSE	If set to true route to CAVS else route to ItemBalance EBS. This invocation would be with respect to ItemBalance EBS
Routing.ItemBalanceEBS.UpdateItemBalanceList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncresponsesimulator	CAVS SOAP URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to ItemBalance EBS
Routing.ItemBalanceEBS.UpdateItemBalanceList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Environment code like 'PRODUCTION'/ 'CAVS' etc. Identifies the installation environment
Routing.TimeStampManager.JDEE1_01.EndpointURI	http://${http.hostname}:${http.port}/<Env>/IntegrationTimeStampManager	JDE IntegrationTimeStampManager ABS service URL. When RouteToCAVS property is set to false use the URL mentioned to connect to JDE application This invocation would be with respect to IntegrationTimeStampManager ABS service

Properties	(default) Value/Setting	Description
moduleName	JDE E1	
TRACE.LOG.ENABLED	false	Use tracelog for the flow
serviceName	UpdateItemListJDEE1ReqABCSImpl	
Default.SystemID	JDEE1_01	System ID of JDE application instance
ABCSExtension.PreXformABMtoEBM	false	User exit for the pre-process ABM should be called or not
ABCSExtension.PostXformABMtoEBM	false	User exit for the pre-process EBM should be called or not
ABCSExtension.PreInvokeEBS	false	User exit for the post-process EBM should be called or not
ABCSExtension.PostInvokeEBS	false	User exit for the post-process ABM should be called or not
Routing.TimeStampManager.RouteToCAVS	false	If set to true route to CAVS else route to JDE application. This invocation would be with respect to IntegrationTimeStampManager ABS service
Routing.TimeStampManager.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS SOAP URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to IntegrationTimeStampManager ABS service
Routing.ItemEBS.UpdateItemList.RouteToCAVS	FALSE	If set to true route to CAVS else route to ItemBalance EBS. This invocation would be with respect to ItemBalance EBS
Routing.ItemEBS.UpdateItemList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncresponsesimulator	CAVS SOAP URL. When RouteToCAVS property is set to true, use the URL mentioned to connect to CAVS. This invocation would be with respect to ItemBalance EBS
Routing.ItemEBS.UpdateItemList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Environment code like 'PRODUCTION'/ 'CAVS' etc. Identifies the installation environment
Routing.TimeStampManager.JDEE1_01.EndpointURI	http://${http.hostname}:${http.port}/<Env>/IntegrationTimeStampManager	JDE IntegrationTimeStampManager ABS service URL. When RouteToCAVS property is set to false use the URL mentioned to connect to JDE application This invocation would be with respect to

Properties	(default) Value/Setting	Description
moduleName	JDE E1	
		IntegrationTimeStampManager ABS service
TRACE.LOG.ENABLED	false	Use tracelog for the flow
serviceName	InitialLoadBillOfMaterialsListJDEE1toAgileImpl	
Default.SystemID	JDEE1_01	System ID of JDE application instance
TRACE.LOG.ENABLED	false	Use tracelog for the flow
ABCSExtension.PreXformABMtoAXML	false	User exit for the pre-process ABM should be called or not
ABCSExtension.PreInvokeAIS	false	User exit for the pre-process Agile aXML should be called or not
PARENT_ITEM_REVISION_LEVEL	B01	The item revision that will be used in Agile PLM for the parent items updated **Note:** 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.
RELEASE_CHANGE_ORDER_REASON	JDE E1 to Agile PLM BOM Initial Load	The Change Order Reason to add to the Change Order created in Agile PLM
RELEASE_CHANGE_ORDER_DESCRIPTION	JDEE1 to Agile PLM BOM Initial Load	The Change Order Description to add to the Change Order created in Agile PLM
serviceName	InitialLoadItemListJDEE1toAgileImpl	
Default.SystemID	JDEE1_01	System ID of JDE application instance
TRACE.LOG.ENABLED	false	Use tracelog for the flow
ABCSExtension.PreXformABMtoAXML	false	User exit for the pre-process ABM should be called or not
ABCSExtension.PreInvokeAIS	false	User exit for the pre-process Agile aXML should be called or not
PRIMARY_ITEM_ID	2ndItemNumber	Property that determines which JDE E1 Item Number to use in Agile PLM
ITEM_REVISION_LEVEL	A01	The item revision that will be used in Agile PLM for all items imported. **Note:** This needs to be populated with the non-blank value; otherwise the

Properties	(default) Value/Setting	Description
moduleName	JDE E1	
		Item Initial Load will not be processed.
RELEASE_CHANGE_ORDER_REASON	JDE E1 to Agile PLM Item Initial Load	The Change Order Reason to add to the Change Order created in Agile PLM
RELEASE_CHANGE_ORDER_DESCRIPTION	JDE E1 to Agile PLM Item Initial Load	The Change Order Description to add to the Change Order created in Agile PLM

Note. When Multisite_Enabled property 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 is denoted by the first element, Site. For example, Site.Numeric01.

When it is set to FALSE, all these attributes will be set to Page2 or Page3 Flex Fields. Hence, the settings will need to be changed to PageTwo.Numeric01 or PageThree.Numeric01 accordingly.

The names of the attributes can be derived from the ItemABM Schema, which can be found in Agile PLM Interfaces.

Queue Management

The Queue Management feature in the PIP caters to the following requirements:

- An Event to produce filtered payload to a File Destination to a JMS Destination.
- The Payload is defined using a standard XSD.
- The files or JMS Messages produced by Events are sequenced in the order in which the objects are released.

Note. These requirements are leveraged using the Agile Content Service (ACS). ACS has the ability to produce payload to a File or JMS destination. The payload is based on filtered configured for the ACS Event defined by Agile PLM provided AXML schema definition. Also the ACS transmits the messages in the order in which the ATOs are released.

- A Queue to manage the order of Messages.
- A Queue Monitoring UI to enable reordering and resubmitting unprocessed messages.
- The Queue manages the payloads based on the Business Process for which the message is produced by the Event.
- A Queue controlling mechanism
  - Triggers the Business Flow based on the business process of message.
  - Processes the messages sequentially depending upon 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.
- The order of the messages, which have not been picked for processing, can be re-ordered.

For more information on features and functionality of Queue Manager, and how to use it, refer to the latest "*Agile PLM Integration Pack for JDE E1 – User Guide*" located at http://www.oracle.com/technology/documentation/agile.html.

Queue Management Solution

The Queue Management Solution comprises of the following components:

- **Queue DB**: The database persist the data related to a Queue messages.
- **Queue Controller**: Polls for new Event payloads and add 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**: UI which monitors the Queue message status supports Reordering of priorities of the Queue Messages. Also, it provides the facility to resubmit the unprocessed messages.

Queue Controller

A polling strategy on the Queue DB is used for addressing the Queue Management business requirements. The Queue Controller provides an ECO system to ensure that this polling strategy works in tandem to ensure the following:

- All Event transmitted File/JMS Messages are added to the Queue.
- At any given point of time there is only one pending message in the control table
- Once the processing of a message in control table is complete, insert the highest priority queue message from the queue table to the control table.
- In case the Integration flow errors out, the queue manager will wait until the message is resubmitted or removed.

Queue Schema

In order to support the above solution flow a polling strategy similar to "PollingControlTableStrategy" is used. Two tables are used to manage the sequential processing and reordering of messages.

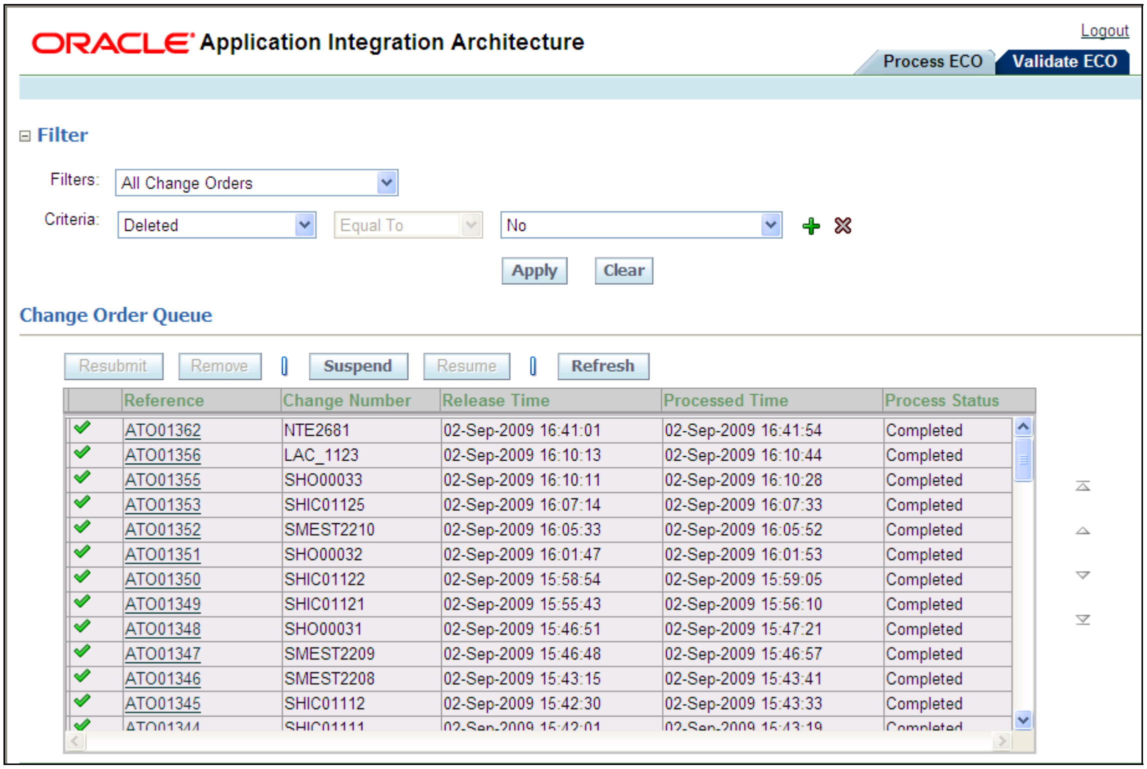
The first table QUEUE_TABLE will have all the queue messages that are being provided by the Event trigger. The QUEUE_CONTROL_TABLE table will store the relevant information of the message from the QUEUE_TABLE which has not been processed yet.

The Queue Manager needs to ensure that there is only one message in the control table which is not yet processed. When the processing of a message is complete, a Pending message from the Queue table is inserted into this table. This would facilitate the Sequential processing of message. Also since all the pending messages are stored in the Queue table, they could be reordered.

Queue Monitor

When a Change Order is released by Agile Content Service (ACS), it is picked up by the Queue Controller. The Queue Monitor displays a list of all the Change Orders that are waiting to get processed. It also facilitates you to reorder their sequence of processing.

For more information, see the Agile PLM to JDE E1 Integration User Guide located at http://www.oracle.com/technetwork/documentation/agile-085940.html#aia. Scroll down to the Agile AIA Documentation section, click **Download** and extract **E14766_01.pdf** from the zip file.



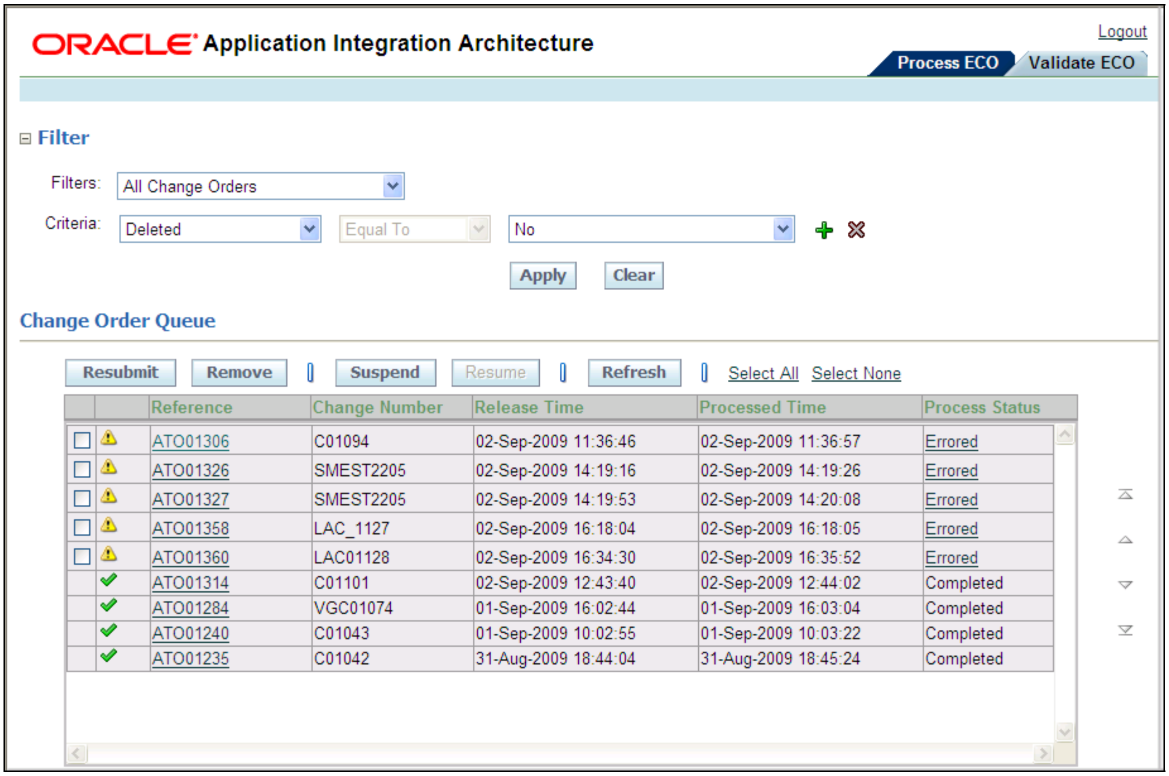
Filters: All Change Orders

Criteria: Deleted | Equal To | No

Change Order Queue

	Reference	Change Number	Release Time	Processed Time	Process Status
✓	ATO01362	NTE2681	02-Sep-2009 16:41:01	02-Sep-2009 16:41:54	Completed
✓	ATO01356	LAC_1123	02-Sep-2009 16:10:13	02-Sep-2009 16:10:44	Completed
✓	ATO01355	SHO00033	02-Sep-2009 16:10:11	02-Sep-2009 16:10:28	Completed
✓	ATO01353	SHIC01125	02-Sep-2009 16:07:14	02-Sep-2009 16:07:33	Completed
✓	ATO01352	SMEST2210	02-Sep-2009 16:05:33	02-Sep-2009 16:05:52	Completed
✓	ATO01351	SHO00032	02-Sep-2009 16:01:47	02-Sep-2009 16:01:53	Completed
✓	ATO01350	SHIC01122	02-Sep-2009 15:58:54	02-Sep-2009 15:59:05	Completed
✓	ATO01349	SHIC01121	02-Sep-2009 15:55:43	02-Sep-2009 15:56:10	Completed
✓	ATO01348	SHO00031	02-Sep-2009 15:46:51	02-Sep-2009 15:47:21	Completed
✓	ATO01347	SMEST2209	02-Sep-2009 15:46:48	02-Sep-2009 15:46:57	Completed
✓	ATO01346	SMEST2208	02-Sep-2009 15:43:15	02-Sep-2009 15:43:41	Completed
✓	ATO01345	SHIC01112	02-Sep-2009 15:42:30	02-Sep-2009 15:43:33	Completed
✓	ATO01344	SHIC01111	02-Sep-2009 15:42:01	02-Sep-2009 15:43:19	Completed

Change Order Queue Showing Completed Process Status



Filters: All Change Orders

Criteria: Deleted | Equal To | No

Change Order Queue

		Reference	Change Number	Release Time	Processed Time	Process Status
□	⚠	ATO01306	C01094	02-Sep-2009 11:36:46	02-Sep-2009 11:36:57	Errored
□	⚠	ATO01326	SMEST2205	02-Sep-2009 14:19:16	02-Sep-2009 14:19:26	Errored
□	⚠	ATO01327	SMEST2205	02-Sep-2009 14:19:53	02-Sep-2009 14:20:08	Errored
□	⚠	ATO01358	LAC_1127	02-Sep-2009 16:18:04	02-Sep-2009 16:18:05	Errored
□	⚠	ATO01360	LAC01128	02-Sep-2009 16:34:30	02-Sep-2009 16:35:52	Errored
	✓	ATO01314	C01101	02-Sep-2009 12:43:40	02-Sep-2009 12:44:02	Completed
	✓	ATO01284	VGC01074	01-Sep-2009 16:02:44	01-Sep-2009 16:03:04	Completed
	✓	ATO01240	C01043	01-Sep-2009 10:02:55	01-Sep-2009 10:03:22	Completed
	✓	ATO01235	C01042	31-Aug-2009 18:44:04	31-Aug-2009 18:45:24	Completed

Change Order Queue Showing Completed Errored Process Status

Queue Manager Services

The following services are deployed as part of the Queue Manager:

1. CreateQueueService
2. CreateQueueControlService
3. QueueProcessorService
4. QueueProcessorServiceImpl

1. CreateQueueService

The CreateQueueService is implemented as an ESB Routing Service. An Adapter Service (File/JMS Adapter) polls on 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 into 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 ESB 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)

2. **CreateQueueControlService**

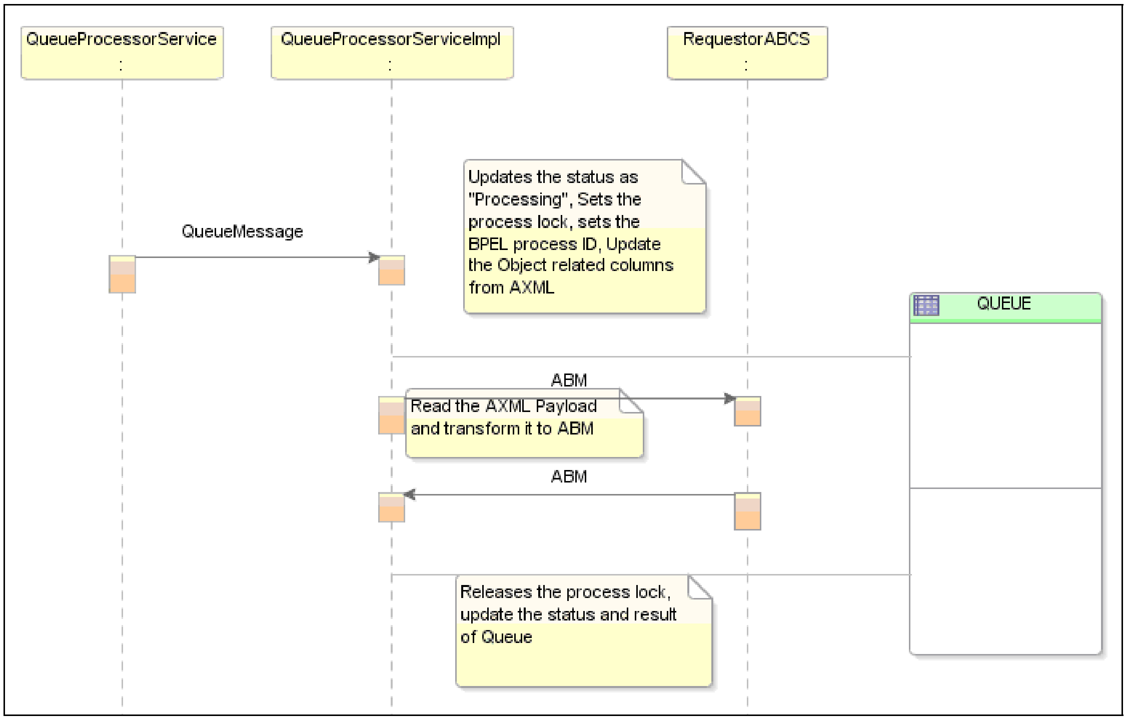
The CreateQueueControlService is implemented as an ESB Routing Service. A DB Adapter polls on the QUEUE_CONTROL_TABLE table. If there are no rows which are in Pending status, the CreateQueueControlService invokes a DB Adapter service which executes a custom SQL. This SQL identifies the highest priority pending Queue message from QUEUE_TABLE table and inserts the same in QUEUE_CONTROL_TABLE table.

This polling strategy ensures that at any point of time there is only one Pending message in the QUEUE_CONTROL_TABLE table. Once the Pending message is processed and status completed, a new Pending message is inserted from QUEUE_TABLE table to the QUEUE_CONTROL_TABLE table. When the status for a message is completed in the QUEUE_CONTROL_TABLE that row would be deleted from the table.

3. QueueProcessorService

The QueueProcessorService is implemented as an ESB service which acts like an Interface and provides a façade in front of 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 which processes the message. Based on the result from the implementation service the status of the message is updated in the control table.

4. **QueueProcessorServiceImpl**


QueueProcessorService :
QueueProcessorServiceImpl :
RequestorABCS :
Updates the status as "Processing", Sets the process lock, sets the BPEL process ID, Update the Object related columns from AXML
QueueMessage
QUEUE
ABM
Read the AXML Payload and transform it to ABM
ABM
Releases the process lock, update the status and result of Queue

Queue Processor Service Sequence diagram

The primary task of this service is to invoke the RequestorABCS. The Response from RequestorABCS is processed and the Queue is updated with processing status.

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

Output: QueueStatusMessage which contains the status and result of processed Queue message.

This table describes the queue services and implementation steps:

#	Name	Step Description
1	QueueProcessorService Invokes QueueProcessorServiceImpl process	The QueueProcessorService invokes QueueProcessorServiceImpl with QueueMessage (generated by Toplink solution for QUEUE table) as input.
2	Invoke UpdateQueueStatus DB Adapter service	The input QueueMessage in this process is assigned with following values to update the Queue message in the Queue DB PROCESS_STATUS: Processing PROCESS_ID: BPEL Process Id PROCESS_LOCK: 1
3	Transform AgileData (aXML) to ABM	The QueueMessage will have the AgileData payload which

#	Name	Step Description
		is transformed to ABM
4	Invoke RequestorABCS	QueueProcessorServiceImpl invokes the RequestorABCS with ABM as input.
5	Invoke Coarse Grained Web Service	RequestorABCS optionally invokes the coarse grained web services to get the ABM populated with any missing information required for the Integration flow.
6	RequestorABCS Transforms ABM to EBM	The response ABM from coarse grained WS is transformed to EBM and an operation on EBS is invoked with EBM as the input.
7	RequestorABCS orchestrates the business flow	The RequestorABCS routes the EBM to the EBS,
8	EBS routes the response to RequestorABCS	The response EBM from EBS is routed to the RequestorABCS which is transformed to ABM and returned to the QueueProcessorServiceImpl
9	QueueProcessorServiceImpl invokes UpdateQueueResult DB Adapter service	The result from the RequestorABCS is used to update the status of Queue in the Queue DB. Also the Process lock is released.

Transformations

The aXML payload is transformed to the ABM which is input for the RequestorABCS. Since the ABM schema is defined on the lines of aXML schema this transformation will be simpler to do in the Jdeveloper XSL Mapper.

Implementation Details

The QueueProcessorServiceImpl is implemented as an Asynchronous BPEL process. There are calls to the RequestorABCS, DB Adapters for updating Queue status and invoking the RequestorABCS. These involve some logic (parsing the aXML payload) which cannot be achieved using the ESB.

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

Error Management

All errors in the Integration flow are handled in the *RequestorABCS.* Any such errors leading to failure of the Queue processing will be handled in this process. As a result of such error the Queue Status and Result with failure status is updated in the Queue DB.

Setting Up for Change Order Creation in Released Status in EBS

OOTB ECO will be released from Agile in *Released* status and it must be created in *Scheduled* status ,so in ECO_STATUS_CODE DVM *Released* status of Agile is mapped to **SCHEDULED** value of **COMMON** Column .

To create a change order in EBS in *Released* status:

1. Apply EBS patch 12895719:R12.ENG.C on EBS host.
2. In Fusion Middleware, update the "COMMON" and "AGILE_01" columns of the ECO_STATUS_CODE DVM.

   This table shows the old settings.

COMMON	AGILE_01
SCHEDULED	Released
RELEASED	

   This table shows the new settings.

COMMON	AGILE_01
SCHEDULED	
RELEASED	Released

3. Save the DVM changes.

After completing these steps, you can create change orders in *Released* status in EBS using the Agile Product Lifecycle Management (PLM) integration for Oracle E-Business Suite.

Appendix A - Troubleshooting

1. Issue: Creating an ECO fails with the error “The SQL Exception is: "javax.resource.ResourceException: RollbackException: Transaction has been marked for rollback: Timed out".

   Solution:
   Increase the timeout values.
   For example, for 100 Affected Items [without any BOM data] being created in JDE E1 through a Change Order released from Agile PLM.

Location	Property to modify	Sample Value
<SOA_ORACLE_HOME>/bpel/domains/<domain_name>/config/domain .xml	syncMaxWaitTime	120
<SOA_ORACLE_HOME>/integration/esb/config/esb_config.ini	xa_timeout jms_receive_timeout	120 120
<SOA_ORACLE_HOME>/j2ee/<domain_name>/application-deployments/orabpel/ejb_ob_engine/orion-ejb-jar.xml	transaction-timeout	120
<SOA_ORACLE_HOME>/j2ee/<domain_name>/config/transaction-manager.xml	transaction-timeout	120

2. Issue: In ECO forward flow, after the ECO is processed successfully but the status attribute (flex) in the ECO in Agile PLM is not getting updated.

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

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

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

Queue Issues

1. Issue: Once an ECO/SCO has been released in Agile PLM, the Queue does not display any corresponding entry for the change order.

Solution:

a. Check Agile PLM for ATO which was created on the release of the particular change order. Check the status on the Where Sent tab of the ATO.

b. If it shows a 'Failure' message, this implies that an error occurred while ACS was processing the publishing of the data to JMS destination. The error message is specified in the Transmission Notes column.

c. After you make any necessary changes to the transfer order or the destination to correct the problem, 'Reset' the destination to attempt delivery again. Once a destination has failed, no other transfer orders can be sent to that destination until it has been reset.

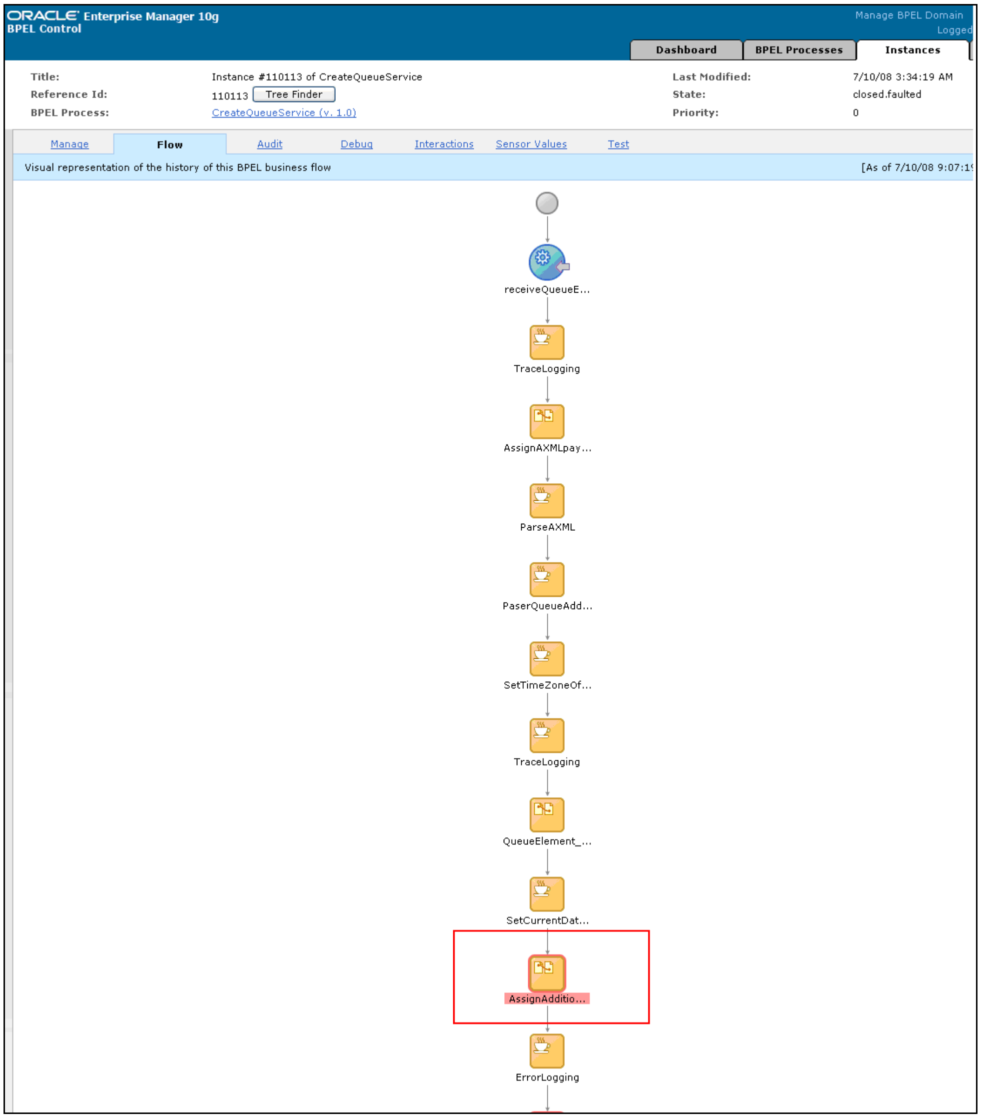
To reset the destination:

1. In Agile Java client, go to Admin > System Settings > Agile Content Service > Destinations.
2. Select the particular destination and click the Reset button on the top.

After resetting the Destination, test the Destination to ensure the Test is Successful for the Destination. In case it fails, this has to be resolved, mainly by ensuring all the ECO Queue settings are correct and OPMN Port specified in the URL is correct.

If the status of the ATO transfer is Success, it implies that the ACS publishing of data to JMS queue was successful. Then we need to troubleshoot in the BPEL console:

3. Navigate to the BPEL Console: http://<server *name>:<port number>/BPELConsole*
4. Click the Instances tab. Check for the instance of the *CreateQueueService* for which the error has occurred.
5. Click the Instance Name and go to the *Flow* link. In the BPEL flow shown, find the element at which the error has occurred.



Title: Instance #110113 of CreateQueueService

Reference Id: 110113 Tree Finder

BPEL Process: CreateQueueService (v. 1.0)

Last Modified: 7/10/08 3:34:19 AM

State: closed.faulted

Priority: 0

Visual representation of the history of this BPEL business flow

[As of 7/10/08 9:07:1

6. Click the element to view the Activity Audit trail which will have details of the error.

Index