# Oracle® Application Integration Architecture

Agile Product Lifecycle Management Integration Pack for SAP: Design to Release Implementation Guide

Release 11.1

**E27417-03**

April 2012

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

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

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

ORACLE

Oracle Application Integration Architecture Agile Product Lifecycle Management Integration Pack for SAP: Design to Release Implementation Guide, Release 11.1

E27417-03



Primary Author: Oracle Corporation

# Contents

# Preface

Welcome to the Agile Product Lifecycle Management Integration Pack for SAP: Design to Release Implementation guide.

## What's New in this Guide

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

## Common Oracle AIA Pre-Built Integration Guides

- Oracle Application Integration Architecture Pre-Built Integrations 11.1 includes the following guides shared by all products delivered with this release:
- Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations Release 11.1
- This guide provides an overview of the installation process, including how to install, configure, and deploy your pre-built integrations. The steps required to upgrade your pre-built integrations to the latest release are also provided.
- Oracle Application Integration Architecture Pre-Built Integrations 11.1:

Utilities Guide

- This guide describes:
- How to work with and configure Session Pool Manager (SPM), which is a service in the Oracle SOA Suite web server whose primary function is to manage a pool of web server session tokens that can be reused by BPEL flows.
- How to deploy and configure the AIACompositeScheduler. This is a utility component that is used by pre-built integrations to schedule a service-oriented architecture (SOA) composite to be invoked at the specified time interval.
- Oracle Application Integration Architecture Pre-Built Integrations 11.1: Product-to-Guide Index

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

# Additional Resources

The following resources are also available:

- Oracle Application Integration Architecture Foundation Pack:

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

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

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

- Known Issues and Workarounds:

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

- Release Notes:

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

- Documentation updates:

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

# Chapter 1: Overview of Agile PLM Integration for SAP

This chapter discusses:

- Architecture of Agile PLM Integration
- Agile PLM to SAP process flows
- SAP to Agile PLM process flows
- Solution Design Assumptions and Constraints
- Components of Agile SAP Design To Release PIP

This integration between Agile PLM and SAP is designed to address the primary requirements for product information synchronization between Agile Product Collaboration and SAP.

The Agile PLM pre-built integration is aimed at enabling product development process between Agile PLM and SAP. This integration allows rapid implementation of Agile's next generation integrated enterprise PLM processes and helps customers reduce cost and mitigate risks associated with third party and custom integrations.

This integration aims to meet the following objectives:

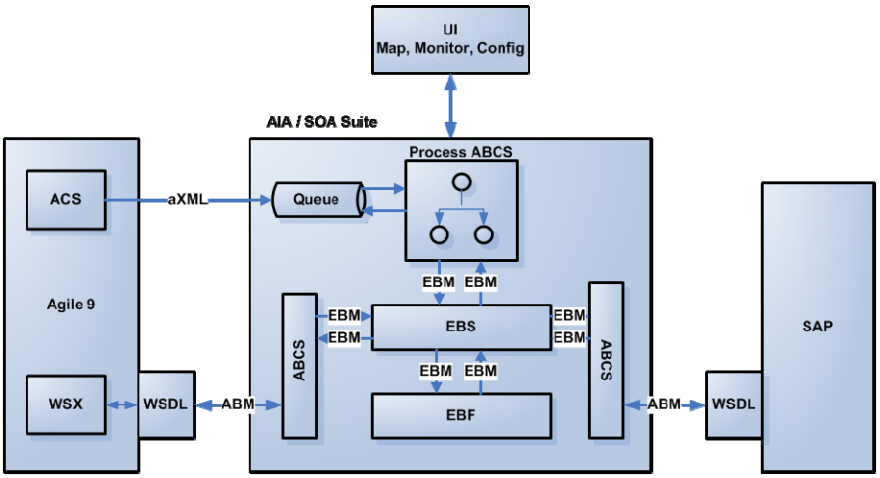
- Follows a business process flow
- Maintains data accuracy and integrity at all times in both the systems
- Allows for easy configuration of business process flow, data field mappings, and transformations tools, such as Rules, Lookups, and User Exits
- Provides for robust exception handling including ease of understanding for end user, strong notification configurability, and easy troubleshooting

These are the functionalities of the Agile to SAP integration:

- Manufacturing release of new product definition and product launch
- Change Management of previously launched products
- Bi-directional synchronization of Engineering Change status and Item Cost information, from SAP to Agile PLM
- Tracking and monitoring of the change processing queue

# Architecture of Agile PLM to SAP Integration

The Agile PLM to SAP Integration is requester-provider type integration as shown in this diagram:




## Agile to SAP Processes

These are the Agile to SAP Processes:

- Change Order Release
- New Part Request

Change Order Release

During a product design phase, new products and parts are introduced and existing parts go through design changes. Change Orders (CO) are used to release information about the attributes and design information of a part to the manufacturing system.

The Change Order Release Process (CORP) constitutes of New Part/Product Release (PREL) and Product Design Modification flows of Agile PLM.

While Agile is the system of record for item description, design, specs and many other pieces of information above, the ERP system typically has many more attributes and placeholders for information than the PLM system. Hence, the change order release needs to be updated in ERP system.

The release of a change order in Agile PLM system acts as a trigger for the synchronization of product design information with ERP system. Agile system being a system of records for product design data, the synchronization process typically involves transfer of the released revision of product design from Agile PLM to the manufacturing system. The Agile system can be configured to trigger the synchronization process on any Change Order status as required.

Change Order Release Process as well as New Product Release processes use the same integration sequence. Chapter 2 - Process Integration for Change Order Release describes the integration details for both.

New Part Request

A New Part Request (NPR) process can be triggered On-demand by user. This is a synchronous process, which reserves and fetches the part number from the SAP system.

See Chapter 3 - Process Integration for New Part Request for complete information.

# SAP to Agile Processes

Though Agile is considered as system of records, there might be several part attributes that can be maintained within ERP. The Product Data Management (PDM) group might require information on these attributes within PLM item data. To provide the PDM group with this information within Agile item object, SAP to Agile process enable synchronous or batch update of the configured field values from ERP in to designated item fields of PLM.

For example update in cost of an item in ERP would get updated in Agile PLM in batch mode, based on the back ground schedule of job in SAP.

## Solution Design Assumptions and Constraints

These are the design assumptions and constraints:

### Design Assumptions

- Agile Content Server will be the used for Events to trigger the payload from Agile to Integration.
- This design assumes that the following statements are true:
    - There will be pre-defined blank templates made available for Custom fields
    - Transformation logic for classification elements will be pre-coded in the Out-of-the-Box XSL, but you may need to modify it to suit your PLM implementation

- This design leverages AIA error handling framework.

### Design Constraints

- In few cases configuration driven XSLT may not reflect the changes immediately and may require a restart as the main XSL sheet would get cached after a successful compilation
- ACS limits the Events to be triggered from workflow only for the Change Status which can be used for this Integration.
- Error handling capabilities of this integration are constrained by the capabilities of AIA framework.

# Components of Agile PLM Integration for SAP

This integration has the following components:

- Agile PLM Content Service (ACS)
- Software Development Kit (SDK)
- Agile Process extensions (PX)
- Web Service Extensions (WSX)
- Oracle Application Integration Architecture (AIA) Foundation Pack
- SAP

## 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 and BOM changes during any phase of the product lifecycle 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.

## Software Development Kit (SDK)

SDK contains a collection of tools; application programming interfaces (APIs), sample applications, and documentation. You use SDK to build custom applications that access Agile application server functionality. By using Agile SDK, you can create programs that perform tasks automatically in Agile PLM.

Agile SDK enables the following operations:

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

Agile SDK has the following modules:

**Agile API**: A Java API with interfaces that expose Agile PLM business objects. Use Agile API to create additional Agile PLM clients. You can also use it as part of an extension developed using web service extensions (WSX) or process extensions (PX).

**Process Extension(PX)**: A framework that allows Agile PLM customers to extend the functionality of Agile PLM clients by adding external reports, user-driven and workflow-driven customized actions, customized tools, and customized automatic number sources. PX helps in binding the data in Agile PLM with other applications.

The new part request is the only process integrations that use PX to extract Agile PLM's data and send it to ABCS.

**Web service Extension (WSX)**: A framework that allows Agile PLM customers to extend the functionality of the Agile PLM server and expose customer-specific solutions using a web service. WSX is a Web service engine enabling communication between Agile PLM and disparate systems both internal and external, including Enterprise Resource Planning (ERP) systems. WSX can be used to provide content to exchanges, reports, and custom applications and import product content data from ERP and other supply chain applications. WSX can simplify the process for aggregating raw product content and making critical product content available in realtime to other core systems.

**For more information** about Agile PLM components, see "*Agile Product Lifecycle Management Administrator Guide" and "SDK Developer Guide*."

## Oracle Application Integration Architecture Foundation Pack

Oracle Application Integration Architecture (AIA) Foundation Pack enables customers to simplify cross-application business process integrations using a standards-based, pre-built integration solution. Designed to promote rapid deployment, re-usability, and configurability of business processes, AIA Foundation Pack saves significant time, effort and cost, compared to building integrations from the ground up and helps you realize the value of a Service Oriented Architecture (SOA) at an accelerated pace.

AIA pre-built integrations are specific to business processes and applications. Built using AIA Foundation Pack, these specific business process integrations provides an open, standards-based approach for organizations to integrate end-to-end business processes across a broad range of custom, Oracle or third-party applications.

**For more information** about the AIA Foundation Pack development methodology, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation***,** Pack 11g Release 1**.**

## SAP

SAP is an acronym for System Application and Products which creates a common centralized database for all the applications running in an organization. SAP products are generally focused on Enterprise Resource Planning (ERP).

SAP applications are built to manage product operations, cost accounting, assets, materials and personnel. SAP ERP runs on majority of platforms including Microsoft Windows and it uses client/server model. SAP provides majority of new dimensional enterprise products namely SEM, SRM, CRM, APO, etc.

# Chapter 2: Process Integration for Change Order Release

This chapter discusses:

Posting CO Data to SAP

Change Order Release Process

- CO Release Process Integration Solution Assumptions
- CO Release Integration Sequence
- AIA Services for CO Release
- CO Release Integration Customization Points

Integration of Change Order Release, namely the Engineering Change Order (ECO) and New Part/Product Release (PREL) from Agile to SAP is about introducing a new product, components & structure into SAP upon release of design from Agile or updating existing design metadata or structure.

This process is triggered from Agile.

**Note**: The integration flow is similar for both ECO and PREL use cases.

## Change Order Release Process

The process of change order release, which comprises of new part introduction and manufacturing update, is the flow of Item and BOM information from Agile to SAP. This information is pushed from Agile, triggered by an event tied to the change in status of a Change Order object. The information is then parsed in an integration object format and sent to SAP for implementation. A confirmation of the implementation status is sent back to Agile.

This integration process supports the following flow:

1. Release of Change Order in Agile
2. aXML generation by Agile Content Server
3. Parsing and transformation of aXML Data
4. Posting Changer Order Data to SAP
5. Communicating Order Processing Status to Agile

## Release of CO in Agile

When a change analyst approves the change order in Agile, it is marked as *Released*. This makes all the changes specified in CO take effect in Agile.

Before a CO is released, it is pre-validated at the approval stage. This involves validation of certain business rules to ascertain that the flow of CO from Agile to SAP would meet all the conditions set in the destination system.

### aXML generation by Agile Content Server

The Agile content server generates an aXML (Agile XML) file from CO data. This file contains information of items, BOM, manufacturers and the CO itself.

For the purpose of Agile to SAP integration process, the ACS is configured to ensure the following:

- The aXML file is configured to carry the following elements from a CO:
    - **Change Order Data**: Cover Page, Page Two, Page Three, Affected Items tab attributes
    - **Revised Item Data**: Title Block, Page Two, Page Three
- BOM Data: BOM tab of Items (including reference designators) with delta BOMs only for the revision on CO.
- Upon release of a CO, the aXML file goes to a JMS Queue.

## Parsing and Transformation of aXML Data

The data contained in aXML file generated by Agile is not in the format that is understood by enterprise business objects. Hence, this data has to be parsed and transformed.

The parsing and transformation of aXML data entails the following:

1. Segregation of Business Objects
2. Sequencing and Queuing of Change Orders
3. Translating Agile ‘Site’ specific objects into SAP’s ‘Plant’ specific objects
4. Translating Agile CO Types into SAP’s CO Type. All the change types from Agile are mapped to Engineering Change Master record in SAP.
5. Ascertaining business objects existence from SAP to prepare the data in create or update mode.
6. Mapping Agile CO Attributes to corresponding attributes in SAP
7. Defining User Exit Points for custom transformations

## Segregation of Business Objects

The aXML file contains collective information about the business objects – CO, Item Attributes, Revised Item Lines, BOM Redlines, Reference designators, etc. This information is broken down into individual components and mapped, one to one, with corresponding EBOs, namely Item, Change and Structure.

In order to maintain referential integrity, the Change Number is associated with each individual business object.

## Sequencing and Queuing of COs

The CO release process begins with queuing of CO Number in Process Queue Controller, which sequences the COs for transfer of parsed data to SAP. Once the data is processed by SAP, and its implementation status received, the CO is removed from the Process Queue.

For complete details on sequencing and queuing of change orders, refer Queue Manager Section in Appendix D.

## Translating Site Objects

The data coming from Agile is split into individual SAP Organization specific business objects. This is because the data in Agile can be:

- Centralized – all design locations share the same product design information or
- De-centralized – the Item Attributes, change control, etc. are implemented to multiple sites.

The Data in SAP, however, is segregated by Plants.

## Translating CO Types

In Agile, a CO is categorized into the following change types:

- Engineering Change Order (ECO)
- Manufacturing Change Order (MCO)
- Site Change Order (SCO)

These categories are called *Classes* in Agile. A class may have one or more sub-classes.

SAP does not have separate categories for each of these change types, and hence, cannot be differentiated. Hence, while parsing and processing of these change types in Agile, they are translated in SAP with the following characteristics:

| Feature | ECO | MCO | SCO |
|---|---|---|---|
| Customer adoption | All installations | All installations | --- |
| New Revisions for revised items | Mandatory | Not supported | Not supported |
| Tables redlined | BOMs;<br>Global as well as site-specific | Global BOM as well as site-specific | Site-specific BOMs. |
| Effectivity Date | At line level;<br>Global when multi-site is not enabled;<br>Separate for each site when multi-site is enabled | Not specified on Change Order | At line level; site-specific only |
| Other line-level attributes | Global when multi-site is not enabled;<br>Separate for each site when multi-site is enabled | Global when multi-site is not enabled;<br>Separate for each site when multi-site is enabled | Site-specific only |
| New Item Release | Supported | Supported | Only item updates are supported |

## Mapping CO Attributes

The source system attributes are mapped to destination system attributes. This mapping also defines the direction of data flow.

This is accomplished as follows:

1. Two different sets of mappings are defined.
   - One for information flowing from Agile to EBOs.
   - Another, for information flowing from EBOs to Agile.
2. These are accessed in the UI as different ‘mapping profiles’.
3. A parameter is specified with each mapping done from the UI. This parameter determines whether the mapping applies to one or both of the directions of data flow.

See Appendices for complete list of Agile to SAP Mappings.

## Ascertaining Item Existence in SAP

An Item is created in the SAP Plant in two ways:

1. Agile releases New Part Introduction information to SAP through a CO, as a First Time release.
2. The Item information is loaded in SAP by an external system, other than SAP and Agile.

If the Item already exists in the SAP Plant, and Agile releases CO to create the same Item, the system would update the item. Since, Agile does not explicitly pass information about first time or subsequent release of Item; a cross reference table is employed to ascertain the existence of the Item in SAP.

This cross reference table maintains unique identifiers for the Items received from Agile and corresponding Items created in SAP. It also maintains the unique identifiers for the Items created in SAP by an external application. These unique identifiers help in ascertaining the existence of an Item in SAP, thus eliminating any duplication errors.

## Defining User Exit Points

User Exits have been provided in the integration to allow custom transformations or filtration routines that a customer may want to add in the process without affecting the main integration flow. The user exit points for each process are listed in their respective chapters.

## Posting CO Data to SAP

PLM records the design specifications of a part. This information is updated either on an engineering change order (ECO) or a manufacturing change order (MCO). A new part is created in the ERP using reference material in SAP. Revisions to a part, effective date, bills of material, item, and reference designator information are controlled by ECO which transfers the information from Agile to ERP system.

The release of change order in the PLM system acts as a trigger for the synchronization of product design information with ERP system. Synchronization process is the transfer of the released revision of the product design from Agile PLM to the manufacturing system.

Process queue controller drives the integration of change order release process between agile PLM and SAP system. To maintain the integrity of data in the ERP system, it is important that change orders are transferred to that system in the same sequence they were released by the source system.

The Agile PLM to SAP process flow consists of two phases: *Pre-Process ECO* and *Post-Process ECO*:

**Pre-Process ECO**

ABM to EBM transformations

- Invoke Provider

- Receive Response
- Send Response to the Queue

**Post-Process ECO**

- Update transfer status in Agile

## CO Processing in SAP

The processing of Change Order data into the ERP system is the backbone of this integration. As part of this step, the following activities need to be performed:

- **Item Master synchronization**: For all items pushed to the ERP system, verify whether or not the items already exist. If the item didn't exist in ERP then create it. If the Item already exists in ERP, then update the Item. It is assumed that the source and target systems are in synch before the integration come in to play and hence the integration doesn’t check if both the systems are in synch before posting the data.
- **Create** Change Order: The actual Change Order is created as change master object in the ERP system. When the Change master record is posted in to SAP, the status would be set to 'Inactive'. Once all the line items on this Change are posted successfully in to SAP then the integration should change the Change master status in SAP to 'Active'. That means this Change Master is complete and all the changes effective on this Change will be in-force from scheduled effectivity/valid from date. This 'Active' Change Order status of SAP is transferred to the corresponding mapped Page Two field in Agile as "Transferred".
- **Item Plant assignment**: Items are assigned to plants based on criteria specified in the section on supporting distributed manufacturing above.
- **BOM Update** (including reference designator): The aXML data from Agile contains only the changes made to Bills of Material and not the complete Bill of Material. As a result, BOM data needs to be in sync between Agile and ERP for the older revision in order for the new revision of BOM data to be posted successfully.

## Communicating CO Implementation status to Agile

When the process of posting Change Order data into the SAP system completes (successfully or with errors), the following steps are taken

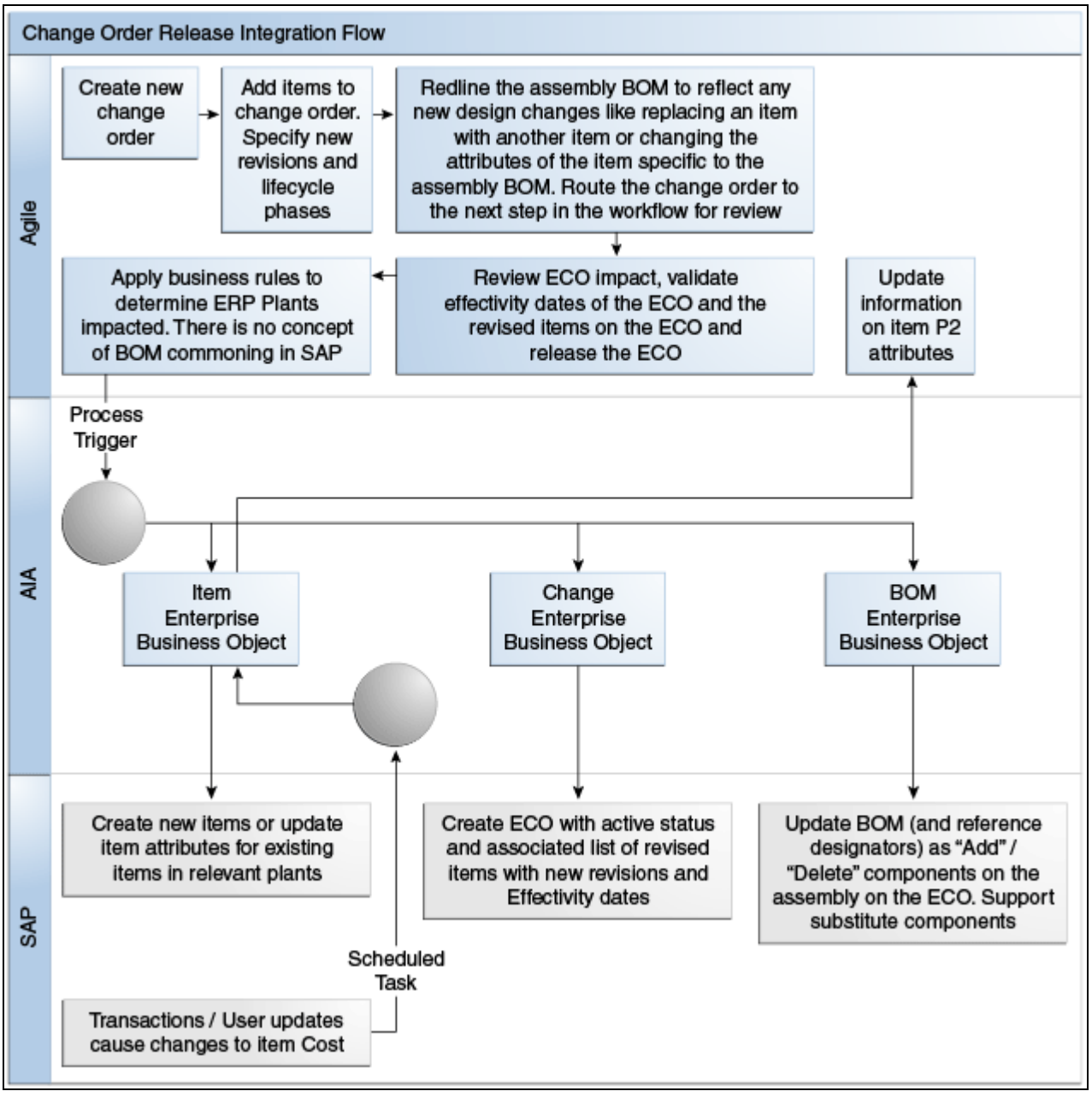
If the parameter **Change.TransferStatusAttribute** is assigned to Page Two field in Agile, then upon successful creation of Change Order in SAP with status *Active* would be propagated back to Agile as 'Transferred' in the specified Page Two field, else *Errored* status is propagated.

# CO Release Process Integration Solution Assumptions

- Change Orders should be transferred to the ERP in the order in which they were released by the source system.
- If a part does not exist in ERP system, then create the part.
- If a part exists in ERP system, just update the part.
- Design changes, Effectivity Date, Bill of Material, Item and Reference Designators are all controlled by ECO.
- Part information and design specifications are updated from Agile to ERP system using ECO.
- There is a business process that prioritizes and monitors the Change Order Process.
- At a given time, only one Change Order (i.e.) first one in the queue undergoes ERP Processing. The next process can start its execution only after the execution of the first process is completed.
- Integration administrator has the provision of error handling and reporting the same to the end user.

# CO Release Integration Sequence

The integration flow is same for both ECO and PREL:



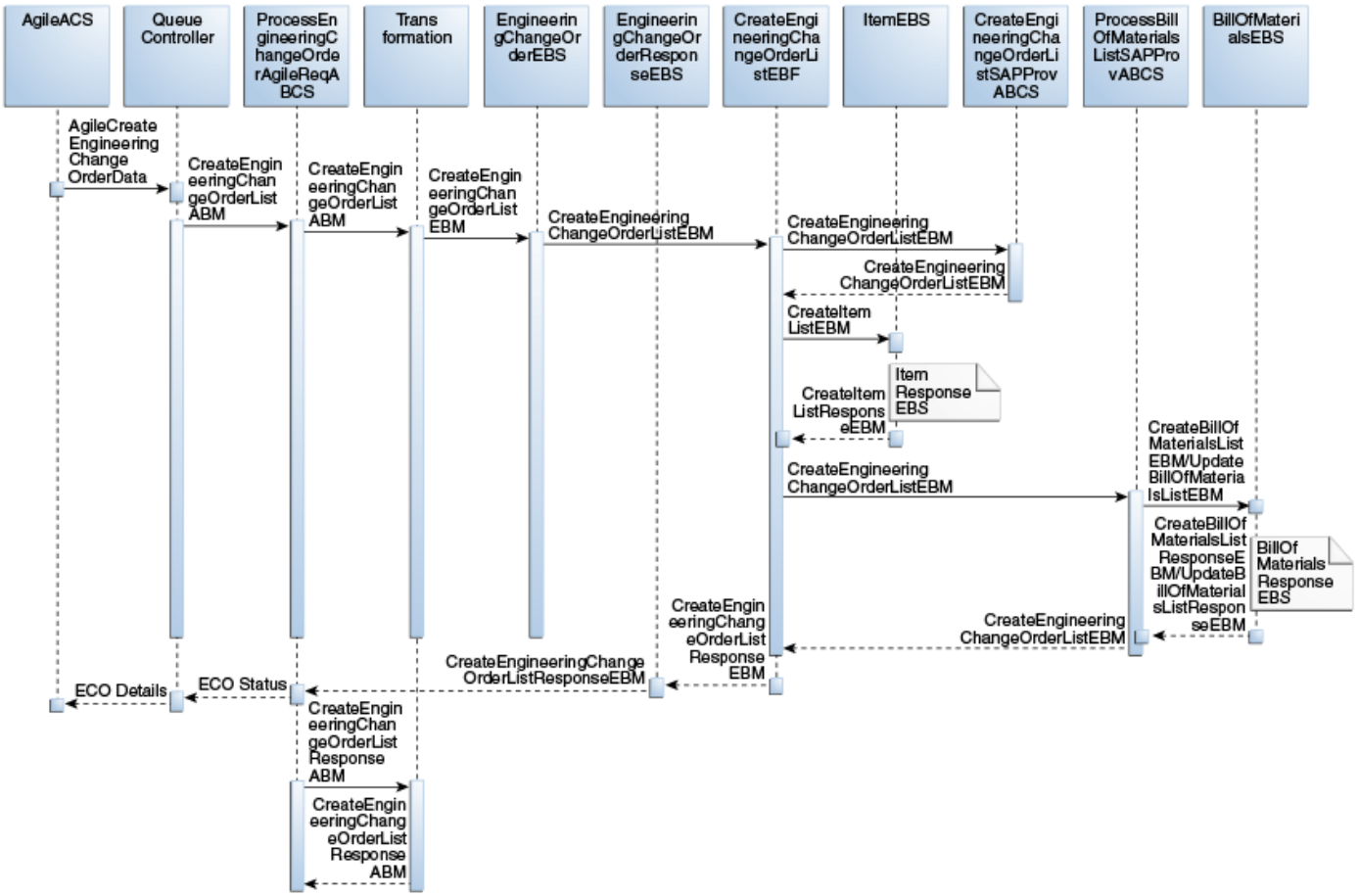

## CO Release Integration Flow

An Engineering Change Order is created with items in Affected Items (AI) tab with new revisions and lifecycle phases specified. The ECO is routed for Approval (workflow step).

1. An ACS Workflow Event is generated on Approval of the ECO in Agile to trigger the ECO process flow.
2. The Queue framework captures the ACS payload (AXML) generated for the event and adds it to the integration Queue.
3. The Queue framework identifies the highest priority Queue Message, processes it to create an ECO ABM and triggers the Requestor ABCS.
4. The Requestor ABCS transforms the ECO ABM to ECO EBM and triggers an operation on the EBS which routes the EBM to the ECO business flow with ECO EBM as the input.
5. The ECO Business Flow:
    - Creates new items in ERP
    - Creates an Engineering Change Order in the SAP.
    - Associates list of revised items with new revisions and affectivity dates, and Schedules the ECO for effectivity date.
    - Creates new BOM
    - Updates existing Item in SAP
    - Updates existing BOM in SAP
    - Updates the Transfer Status in Agile.
6. The status of Queue Message is updated in the Integration Queue, for monitoring.

# CO Release Services Orchestration

The below diagram illustrates the Change Order Services orchestration from Agile Application to SAP ERP.




This table list the activities involved in the Change Order Release Services orchestration:

| # | Activity | Remarks |
|---|---|---|
| 1 | **Agile ACS** acts as a trigger for ECO Use case. | 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. |
| 2 | **QueueController processes the payload** | The Queue Controller Framework reads the highest priority Queue Message and transforms the payload (aXML) to AgileCreateEngineeringChangeOrderListABM. |
| 3 | Invoke | QueueController invokes the |

| # | Activity | Remarks |
|---|---|---|
| | **ProcessEngineeringChangeOrderAgileReqABCS** | ProcessEngineeringChangeOrderAgileReqABCS with AgileCreateEngineeringChangeOrderListABM as input. |
| 4 | **ProcessEngineeringChangeOrderAgileReqABCS** makes call back to Agile Web services, if needed. | ProcessEngineeringChangeOrderAgileReqABCS calls the web services exposed in agile side to enrich AgileCreateEngineeringChangeOrderListABM.<br>AgileCreateEngineeringChangeOrderListABM is transformed into CreateEngineeringChangeOrderEBM. |
| 5 | **ProcessEngineeringChangeOrderAgileReqABCS** invokes **EngineeringChangeOrderE**BS | ProcessEngineeringChangeOrderAgileReqABCS invokes CreateEngineeringChangeOrder operation on EngineeringChangeOrderEBS with CreateEngineeringChangeOrderEBM as input |
| 6 | **EngineeringChangeOrderEBS** routes CreateEngineeringChangeOrderListEBM to **CreateEngineeringChangeOrderListEBF** | CreateEngineeringChangeOrderListEBM is passed to CreateEngineeringChangeOrderListEBF as input. |
| 7 | **CreateEngineeringChangeOrderEBF** invokes ItemEBS | CreateItemListEBM is passed to ItemEBS as input |
| 8 | **ItemEBS** routes the **CreateItemListEBM** to **SyncItemListSAPProvABCS** | CreateItemListEBM is passed to SyncItemListSAPProvABCS as input |
| 9 | **SyncItemListSAPProvABCS** transforms the input and calls the SAP service | SyncItemListSAPProvABCS transforms CreateItemListEBM to the input of SAP service and calls that service.Creates items in ERP |
| 10 | **SyncItemListSAPProvABCS** calls **ItemResponseEBS (Response Port)** with **CreateItemListResponseEBM** | SyncItemListSAPProvABCS invokes the CreateItemListResponse operation on ItemResponseEBS with CreateItemListResponseEBM as input |
| 11 | **CreateEngineeringChangeOrderListEBF** calls **ProcessBillOfMaterialsListSAPProvABCS** | CreateEngineeringChangeOrderListEBM will be passed to ProcessBillOfMaterialsListSAPProvABCS as the input |
| 12 | **ProcessBillOfMaterialsListSAPProvABCS** invokes **BillOfMaterialsEBS** | |
| 13 | Based on the existence of the BOM **BillOfMaterialsEBS** routes to **CreateBillOfmaterialsListSAPProvABCSImpl** (or) **UpdateBillOfMaterialsListSAPProvABCSImpl** | |
| a) | **BillOfMaterialsEBS** calls **CreateBillOfMaterialsListSAPProvABCSI** | CreateBillOfMaterialsListEBM is passed to CreateBillOfmaterialsListSAPProvABCSImpl |

| # | Activity | Remarks |
|---|---|---|
| | **mpl** | |
| b) | **ProcessBillOfMaterialsListSAPProvABCSImpl** is invoked before **BillOfMaterialsEBS** calls **UpdateBillOfMaterialsListSAPProvABCSImpl** | CreateEngineeringChangeOrderListEBM is passed to ProcessBillOfMaterialsListSAPProvABCSImpl & after this UpdateBillOfMaterialsListEBM is passed to UpdateBillOfMaterialsListSAPProvABCSImpl |
| 14<br>a) | **CreateBillOfMaterialsListSAPProvABCSImpl** transforms the input and calls the SAP service | CreateBillOfMaterialsListSAPProvABCSImpl will transform CreateBillOfMaterialsListEBM to the input of SAP service and calls that service.<br>Creates the BillOfMaterialsList in ERP |
| b-i) | **ProcessBillOfMaterialsListSAPProvABCSImpl transforms the input and calls the SAP service** | ProcessBillOfMaterialsListSAPProvABCSImpl transforms CreateEngineeringChangeOrderListEBM to the input of SAP service and calls that service |
| b.ii) | **UpdateBillOfMaterialsListSAPProvABCSImpl transforms the input and calls the SAP service** | UpdateBillOfMaterialsListSAPProvABCSImpl transforms UpdateBillOfMaterialsListEBM to the input of SAP service and calls that service.<br>Updates the *BillOfMaterialsList* in ERP |
| 15<br>a) | **CreateBillOfMaterialsListSAPProvABCSImpl** calls **BillOfMaterialsResponseEBS (**Response Port) with **CreateBillOfMaterialsListResponseEBM** | CreateBillOfMaterialsListSAPProvABCSImpl invokes CreateBillOfMaterialsListResponse operation on BillOfMaterialsResponseEBS with CreateBillOfMaterialsrListResponseEBM as input |
| b) | **UpdateBillOfMaterialsListSAPProvABCSImpl** calls **BillOfMaterialsResponseEBS (Response Port)** with **UpdateBillOfMaterialsListResponseEBM** | UpdateBillOfMaterialsListSAPProvABCSImpl invokes UpdateBillOfMaterialsListResponse operation on BillOfMaterialsEBS with UpdateBillOfMaterialsListResponseEBM as input |
| 16 | **CreateEngineeringChangeOrderEB**F **calls CreateEngineeringChangeOrderSAPProvABCS** | CreateEngineeringChangeOrdeListEBM is passed to CreateEngineeringChangeOrderListSAPProvABCS as input |
| 17 | **CreateEngineeringChangeOrderListSAPProvABCS** transforms the input and calls the SAP service | CreateEngineeringChangeOrderListSAPProvABCS transforms CreateEngineeringChangeOrderListEBM into the input of SAP Service and calls that service.<br>Creates items in ERP, creates an ECO, associates revised items to it and creates BOM |
| 18 | **CreateEngineeringChangeOrderListEBF** calls **EngineeringChangeOrderResponseEBS** (Response Port) with the **CreateEngineeringChangeOrderResponseEBM** | CreateEngineeringChangeOrderListEBF invokes CreateEngineeringChangeOrderListResponse operation on EngineeringChangeOrderResponseEBS with CreateEngineeringChangeOrderListResponseEBM as input. |
| 19 | **EngineeringChangeOrderResponseEBS** | EngineeringChangeOrderResponseEBS *routes* |

| # | Activity | Remarks |
|---|---|---|
| | routes the response message to **ProcessEngineeringChangeOrderAgileReqABCS.** | CreateEngineeringChangeOrderListResponseEBM *to* ProcessEngineeringChangeOrderAgileReqABCS. |
| 20 | **ProcessEngineeringChangeOrderAgileReqABCS** sends the status back to the **QueueController** to update the queue. | ProcessEngineeringChangeOrderAgileReqABCS replies back to the QueueController with the status of the ECO business process.<br>This status is updated against the Queue message in the database by the QueueController |
| 21 | **ProcessEngineeringChangeOrderAgileReqABCS** invokes the Agile Web Services. | ProcessEngineeringChangeOrderAgileReqABCS transforms CreateEngineeringChangeOrderListResponseEBM into AgileUpdateEngineeringChangeOrderListABM.<br>AgileUpdateEngineeringChangeOrderListABM is sent as an input to the Agile Web Service.<br>The web services would update transfer status on the Change Order in Agile which will be predefined P2 or P3 attributes on ECO object in Agile.<br>AgileUpdateEngineeringChangeOrderListResponse ABM is sent back to *ProcessEngineeringChangeOrderAgileReqABCS*. |

# AIA Services for CO Release

The process integration for ECO/PREL uses the following industry components:

| | |
|---|---|
| EBOs | 1. EngineeringChangeOrderEBO<br>2. BillOfMaterialsEBO<br>3. ItemEBO |
| EBMs | 4. CreateEngineeringChangeOrderListEBM<br>5. CreateEngineeringChangeOrderListResponseEBM<br>6. CreateBillOfMaterialsListEBM<br>7. CreateBillOfMaterialsListResponseEBM<br>8. UpdateBillOfMaterialsListEBM<br>9. UpdateBillOfMaterialsListResponseEBM<br>10. CreateItemListEBM<br>11. CreateItemListResponseEBM |

| | |
|---|---|
| EBSs | 12. EngineeringChangeOrderListEBS (CreateEngineeringChangeOrderListResponseEBM)<br>13. BillOfMaterialsEBS (CreateBillOfMaterialsListEBM/ CreateBillOfMaterialsListResponseEBM)<br>14. ItemEBS (CreateItemListEBM/ CreateItemListResponseEBM) |

### Core Components Locations

| | |
|---|---|
| EBO & EBM XSD files | $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/EngineeringChangeOrder/<br>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/BillOfMaterials/<br>$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/Item/ |
| WSDL files | $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/ |

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

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 AIA Artifacts."

## Agile & SAP Components for CO Release

| Services | Agile (Requester) | SAP (Provider) |
|---|---|---|
| ABMs | AgileCreateEngineeringChangeOrderListABM<br>AgileCreateEngineeringChangeOrderListResponseABM<br>AgileUpdateEngineeringChangeOrderListABM<br>AgileUpdateEngineeringChangeOrderListResponseABM | BAPI_MATERIAL_GET_ALL<br>BAPI_MATERIAL_GET_ALLResponse<br>BAPI_MATERIAL_GET_DETAIL<br>BAPI_MATERIAL_GET_DETAILResponse<br>BAPI_MATERIAL_SAVEDATA<br>BAPI_MATERIAL_SAVEDATAResponse<br>CSAP_MAT_BOM_READ |

| Services | Agile (Requester) | SAP (Provider) |
|---|---|---|
| | | CSAP_MAT_BOM_READResponse<br>CCAP_REV_LEVEL_MAINTAIN<br>CCAP_REV_LEVEL_MAINTAINResponse<br>BAPI_MAT_BOM_EXISTENCE_CHECK<br>BAPI_MAT_BOM_EXISTENCE_CHECKResponse<br>CSAP_MAT_BOM_ALLOC_CREATE<br>CSAP_MAT_BOM_ALLOC_CREATEResponse<br>CSAP_MAT_BOM_CREATE<br>CSAP_MAT_BOM_CREATEResponse<br>CSAP_MAT_BOM_MAINTAIN<br>CSAP_MAT_BOM_MAINTAINResponse<br>CCAP_ECN_MAINTAIN<br>CCAP_ECN_MAINTAINResponse |
| ABCS | ProcessEngineeringChangeOrderAgileReqABCS | CreateEngineeringChangeOrderListSAPProvABCSImpl<br>CreateEngineeringChangeOrderListEBF<br>ProcessBillOfMaterialsListSAPProvABCS<br>ProcessBillOfMaterialsListSAPProvABCSImpl<br>CreateBillOfMaterialsListSAPProvABCSImpl<br>UpdateBillOfMaterialsListSAPProvABCSImpl<br>SyncItemListSAPProvABCS |
| EBS | EngineeringChangeOrderEBS (CreateEngineeringChangeOrderList Operation) | EngineeringChangeOrderResponseEBS (CreateEngineeringChangeOrderListResponse Operation)<br>BillOfMaterialsEBS<br>BillOfMaterialsResponseEBS<br>ItemEBS<br>ItemResponseEBS |
| BPEL | CreateQueueService<br>QueueProcessorServiceImpl | --- |
| Mediator Service | ACSAXMLJMSConsumer<br>ACSAXMLFileConsumer<br>CreateQueueControlService | ItemService<br>BOMServiceESB<br>ReserveItemservice |

| Services | Agile (Requester) | SAP (Provider) |
|---|---|---|
| | QueueProcessorService<br>EngineeringChangeOrderService | ECORouterService |

### Component Locations

| | |
|---|---|
| ABO XSD files | $AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas |
| ABM XSD files | $AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas<br>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/SAP/Release1/Core/ABM/EngineeringChangeOrder |

# Integration Services for CO Release

These are the integration services for Change Order release:

## 1. EngineeringChangeOrderEBS

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

**EngineeringChangeOrderEBS Mediator service**

- ProcessEngineeringChangeOrderAgileReqABCS

  Route CreateEngineeringChangeOrderListEBM to
  CreateEngineeringChangeOrderListSAPProvABCSImpl

**EngineeringChangeOrderResponseEBS Mediator Service**

- CreateEngineeringChangeOrderListSAPProvABCSImpl

  Route CreateEngineeringChangeOrderListResponseEBM to
  ProcessEngineeringChangeOrderAgileReqABCS

## 2. ItemEBS

**ItemEBS Mediator Service**

- CreateEngineeringChangeOrderListEBF

Route CreateItemListEBM to SyncItemListSAPProvABCS

**ItemResponseEBS Mediator service**

- SyncItemListSAPProvABCS

  Route CreateItemListResponseEBM to
  CreateEngineeringChangeOrderListEBF

### 3. BillOfMaterialsEBS

**BillOfMaterialsEBS Mediator service**

- ProcessBillOfMaterialsListSAPProvABCS
  - Route CreateBillOfMaterialsListEBM to CreateBillOfMaterialsListSAPProvABCSImpl
  - Route UpdateBillOfMaterialsListEBM to UpdateBillOfMaterialsListSAPProvABCSImpl

**BillOfMaterialsResponseEBS Mediator service**

- CreateBillOfMaterialsListSAPProvABCSImpl

  Route CreateBillOfMaterialsListResponseEBM to
  ProcessBillOfMaterialsListSAPProvABCS

- UpdateBillOfMaterialsListSAPProvABCSImpl

  Route UpdateBillOfMaterialsListResponseEBM to
  ProcessBillOfMaterialsListSAPProvABCS

### 4. ProcessEngineeringChangeOrderAgileReqABCS

ProcessEngineeringChangeOrderAgileReqABCS is used for transforming AgileCreateEngineeringChangeOrderListABM into CreateEngineeringChangeOrderListEBM. This service invokes the CreateEngineeringChangeOrder operation on EngineeringChangeOrderEBS for creation of ECO in SAP.

Based on the status of ECO creation in SAP this service updates the queue status. Also the Transfer status attribute in Change Order is updated by this service. This service is implemented as *Asynchronous* BPEL composite.

The QueueController creates the AgileCreateEngineeringChangeOrderListABM and invokes the ProcessEngineeringChangeOrderAgileReqABCS.

1. Transforms the AgileCreateEngineeringChangeOrderListABM to CreateEngineeringChangeOrderListEBM and invokes the CreateEngineeringChangeOrder operation on EngineeringChangeOrderEBS with CreateEngineeringChangeOrderListEBM as input. This is routed for creation of ECO in SAP.
2. CreateEngineeringChangeOrderListResponseEBM is received from EngineeringChangeOrderEBS and based on the status of ECO creation in SAP, The QueueController is invoked to update the status of the Queue Message.
3. ProcessEngineeringChangeOrderAgileReqABCS transforms CreateEngineeringChangeOrderListResponseEBM into AgileUpdateEngineeringChangeOrderListABM and sent as an input to the Agile Web Service.
4. The web services update the transfer status of the Change Order in Agile which will be predefined P2 or P3 attributes on ECO object in Agile. AgileUpdateEngineeringChangeOrderListResponseABM is sent back to ProcessEngineeringChangeOrderAgileReqABCS.

**Transformations**

AgileCreateEngineeringChangeOrderListABM_to_CreateEngineeringChangeOrderEBM.xsl

1. Transforms AgileCreateEngineeringChangeOrderListABM to CreateEngineeringChangeOrderListEBM
2. CreateEngineeringChangeOrderResponseEBM_to_AgileUpdateEngineeringChangeOrderListABM.xsl
3. Transforms CreateEngineeringChangeOrderResponseEBM to AgileUpdateEngineeringChangeOrderListABM

## 5. CreateEngineeringChangeOrderListSAPProvABCSImpl

1. This is a single operation service. This accepts an Engineering Change Order containing Item and BOMs information message as a request and returns a response.
2. In Agile to SAP flow, CreateEngineeringChangeOrderListSAPProvABCSImpl is used for transforming CreateEngineeringChangeOrderListEBM into CCAP_ECN_MAINTAINABM, which invokes the CreateEngineeringChangeOrder operation in SAP.
3. In return flow, CCAP_ECN_MAINTAIN BAPI sends CCAP_ECN_MAINTAINResponse which is transformed by CreateEngineeringChangeOrderListSAPProvABCSImpl into CreateEngineeringChangeOrderListResponseEBM.
4. This service is implemented as Asynchronous BPEL Composite

# CO Release Integration Customization Points

This table lists the Change Order integration customization information:

### Agile

| Name | Property | Description |
|---|---|---|
| ProcessEngineeringChangeOrderAgileReqABCS<br>(Agile Process ECO requestor flow) | AgileCreateEngineeringChangeOrderListABM_to_CreateEngineeringChangeOrderListEBM_ Impl .xsl | ReqABM to ReqEBM (main) |
| | AgileCreateEngineeringChangeOrderListABM_to_CreateEngineeringChangeOrderListEBM_ Custom.xsl | ReqABM to ReqEBM (custom elements) |
| | CreateEngineeringChangeOrderListEBM_EBMHeader_Custom.xsl | EBM to EBMHeader (custom elements) |
| | CreateEngineeringChangeOrderListEBM_EBMHeader_Impl.xsl | EBM to EBMHeader (main) |
| | CreateEngineeringChangeOrderListResponseEBM_to_UpdateEngineeringChangeOrderListEBM_Impl.xsl | RespEBM to ReqEBM (main) |
| | UpdateEngineeringChangeOrderListEBM_to_AgileUpdateEngineeringChangeOrderListABM_Impl.xsl | ReqEBM to ReqABM (main) |

### SAP

| Name | Property | Description |
|---|---|---|
| CreateEngineeringChangeOrderListSAPProvABCSImpl<br>CreateEngineeringChangeOrderListEBF | CreateEngineeringChangeOrderListEBMtoABM_Custom.xsl | Custom transformations for Engineering Change Order Request EBM to Request ABM |
| | XformReqEBMToResEBM_Custom.xsl | Custom transformations for Engineering Change Order Request EBM to Engineering Change Order Response EBM |
| | CreateECOEBMtoItemListEBM_Custom.xsl | Custom transformations for Engineering Change Order Request EBM to Item List EBM |

| | | |
|---|---|---|
| | UpdateXformEngineeringChangeOrderListEBMtoEBM_Custom.xsl | Custom transformations for update Engineering Change Order list EBM to Engineering Change Order list EBM |
| | XformEBMtoBillOfMaterialProcess_Custom.xsl | Custom transformations for Engineering Change Order list EBM to Engineering Change Order list EBM for BOM |
| | XformECOListEBMtoECOListResponseEBM_Custom.xsl | Custom transformations for Engineering Change Order list request EBM to Engineering Change Order list Response EBM |
| | XformEngineeringChangeOrderListEBMtoEBM_Custom.xsl | Custom transformations for create Engineering Change Order list EBM to Engineering Change Order list EBM |
| | ECOEBMToExistenceCheck_Custom.xsl | Custom transformations for Engineering Change Order list EBM to BOM_Existence_Check ABM |
| ProcessBillOfMaterialsListSAPProvABCS | ECOEBMToGlobalBOMExistenceCheck_Custom.xsl | Custom transformations for Engineering Change Order list EBM to global BOM_Existence_Check ABM |
| | ECOEBMToGlobalUpdateBOMEBM_Custom.xsl | Custom transformations for Engineering Change Order list EBM to global Update BillOfMaterials List EBM |
| | ECOListEBMToBOMListEBM_Custom.xsl | Custom transformations for Engineering Change Order list EBM to Create BillOfmaterials List EBM |
| | ECOListEBMToUpdateBOMListEBM_Custom.xsl | Custom transformations for Engineering Change Order list EBM to Update BillOfMaterials List EBM |
| | XformEBMHeadertoECOListResponseEBM_Custom.xsl | Custom transformations for EBM Header to Create EngineeringChangeOrder List EBM |

| | | |
|---|---|---|
| | XformECOListEBMtoEBMHeader_Custom.xsl | Custom transformations for Create EngineeringChangeOrder List EBM to EBM Header |
| CreateBillOfMaterialsListSAPProvABCSImpl | BOMEBMToCreateGlobalBOMABM_Custom.xsl | Custom transformations for Create BillOfMaterials List EBM to Create Global BOM ABM |
| | BOMEBMToExistenceCheck_Custom.xsl | Custom transformations for Create BillOfMaterials List EBM to global BOM Existence Check ABM |
| | BOMEBMToPlantExistenceCheck_Custom.xsl | Custom transformations for Create BillOfMaterials List EBM to BOM Existence Check ABM |
| | BOMEBMToPlantExtensionABM_Custom.xsl | Custom transformations for Create BillOfMaterials List EBM to BOM Plant Extension ABM |
| | BillOfMaterialsListEBMToCreateBOM_Custom.xsl | Custom transformations for Create BillOfMaterials List EBM to Create BOM ABM |
| | XformBOMReqEBMToResEBM_Custom.xsl | Custom transformations for Create BillOfMaterials List EBM to Create BillOfMaterials List Response EBM |
| UpdateBillOfMaterialsListSAPProvABCSImpl | BOMEBMToGlobalBOMABM_Custom.xsl | Custom transformations for Update BillOfMaterials List EBM to Update global BOM ABM |
| | BOMEBMToPlantExistenceCheck_Custom.xsl | Custom transformations for Update BillOfMaterials List EBM to BOM Existence Check ABM |
| | BOMEBMToPlantExtensionABM_Custom.xsl | Custom transformations for Update BillOfMaterials List EBM to BOM Plant Extension ABM |
| | BOMEBMToUpdateBOMABM_Custom.xsl | Custom transformations for Update BillOfMaterials List EBM to Update BOM ABM |
| | XformBOMReqEBMToResEBM_Custom.xsl | Custom transformations for Update BillOfMaterials List EBM to Update BillOfMaterials List Response EBM |

| | | |
|---|---|---|
| ProcessBillOfMaterialsListSAPProvABCSImpl | XformECOEBMtoABM.xsl | Custom transformations for CreateEngineeringChangeOrderListEBM to Read BOM ABM |
| | XformABMtoECOEBM.xsl | Custom transformations for Read BOM ResponseABM to CreateEngineeringChangeOrderList EBM |
| SyncItemListSAPProvABCS | XFormItemEBMToGetDetailRefMaterial_Custom.xsl | Custom transformations for CreateItem List EBM to Get detail Reference Material ABM |
| | XFormItemEBMToReferenceMaterial_Custom.xsl | Custom transformations for CreateItem List EBM to Get Reference Material ABM |
| | XFormItemEBMToRevisionABM_Custom.xsl | Custom transformations for CreateItem List EBM to Create Revision ABM |
| | XFormItemListEBMToSavedData_Custom.xsl | Custom transformations for CreateItem List EBM to Create Item List ABM |
| | XFormItemListEBMToUpdateItem_Custom.xsl | Custom transformations for CreateItem List EBM to Update Item list ABM(for Get All ReferenceMaterial-4.6) |
| | XFormItemListEBMToUpdate_Item_Custom.xsl | Custom transformations for CreateItem List EBM to Update Item list ABM(for Get Detail ReferenceMaterial) |
| | XformItemListEBMtoItemListResponseEBM_Custom.xsl | Custom transformations for CreateItem List EBM to Create ItemList Response EBM |

# Properties and DVMs for CO Release

The following mandatory DVMs and Properties (in AIAConfigProperties.xml) must be set for the CO Release process:

- AGILE_SITE_TARGET_MAPPING
- AGILE_TARGET_SITE_MAPPING
- LANGUAGE_CODE

- ECO_ENGINEERINGCHANGEORDERLINE_REVISEDBILLOFMATERIALS_BILLOFMATERIALSCOMPONENTITEM_CHANGETYPECODE
- ITEM_PRIMARYCLASSIFICATIONCODE
- ITEM_STATUS_CODE
- ITEM_UOM_CODE
- ECO_STATUS_CODE
- REFERENCE MATERIALS

# Chapter 3: Process Integration for New Part Request

This chapter discusses:

- NPR Process in Agile
- NPR Process in SAP
- NPR Process Integration Solution Assumptions
- NPR Integration Sequence
- AIA Services for NPR
- NPR Integration Customization Points

Agile PLM system record for product design and part information, new part numbers themselves may originate in a system outside of Agile. This is easily understood when looking at the interactions of various roles that can be involved in the process of generating new part numbers.

An engineer designing an electronic assembly, for example, a circuit board may need a part that cannot be found in the system, for instance, a new resistor. The engineer raises a new part request to doc control to route the request to materials managers for review. A buyer looks into catalogs offered by approved manufacturers and finds a couple of parts that meet the criteria. The buyer then contacts the manufacturers, confirms availability, procures specifications, and then approves the new part request with the new manufacturer part numbers attached to it. Doc control compiles all the information and assigns an internal part number corresponding to the manufacturer parts and informs the engineer.

A New Part Request (NPR) process can be triggered on demand by user. This is a synchronous process, which reserves and fetches the part number from the SAP system.

## NPR Process in Agile

A New Part Request process in Agile is executed using an object of the Engineering Change Requests (ECR) class. Since the part number(s) being requested do not exist yet in Agile, there are no affected items on this ECR.

A number of variants of this process are possible:

- A single item number is reserved in SAP per ECR
- Multiple item numbers are reserved in SAP per ECR

## On Demand New Part Request

The synchronous on-demand NPR process allows you to achieve the following:

User in Agile requires a new part number to be used on a design. In Agile, user clicks on New Object icon to create a new item for a given subclass such as Part or Document. The on-demand NPR process compliments or replaces the new item creation process in Agile by providing the user a Process Extension that a user can use to trigger the creation of a new item.

The overall process flow is as follows:

- User calls a process extension (PX) from Agile's auto number menu on the Create New Item popup. This option is available whenever the part number generation process is generic (mainly using a sequence).
- Since a SAP material type is required to get the next available sequence number from SAP, the item type (subclass) is still passed as a parameter as part of this process. In DVM (Domain Value Mapping) the Agile's subclass is mapped to corresponding SAP material type.
- The PX requests a new part number from SAP. This process reserves the next available part number in SAP.
- New reserved part number is sent back to Agile.
- The new part number is returned to the calling PX and the Agile New Part Creation process is executed as usual from there onwards. If an error is encountered during PX execution, then the error message is displayed as a popup window in Agile.

For detail error information, log on to BPEL console.

## NPR Process in SAP

It is assumed that the internal number range assignments are configured in SAP for those material types which are mapped to corresponding Part or Document subclass of Agile that requires auto number generation.

When the synchronous auto number PX is executed from Agile, it calls a standard SAP API with input as material type.

1. SAP checks if there is a valid internal number range defined for the provided material type. If yes, the program returns the next available number to the calling program.
2. If no valid internal number range is defined, it returns an error to the calling program.

Since this complete process is synchronous, the user will not experience the SAP processes at all as they run in the background to return the SAP assigned number. For the user, the process will appear similar to the auto number assignment within Agile.

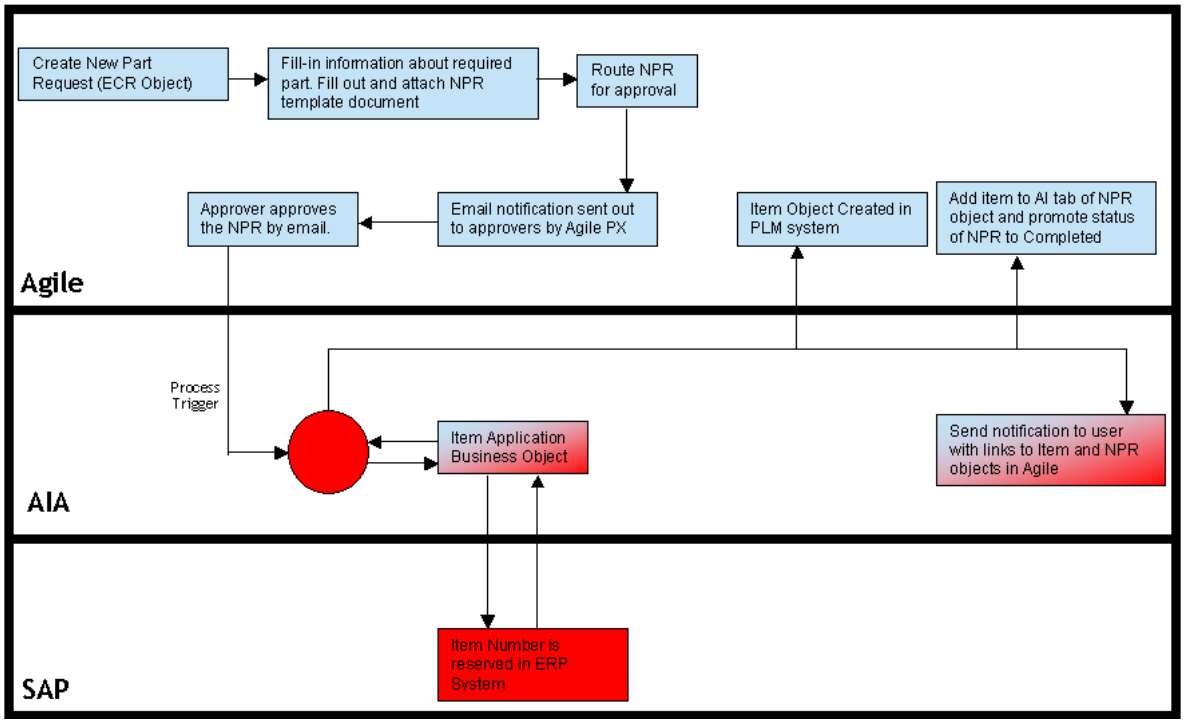
## NPR Process Integration Solution Assumptions

Here are the assumptions for the NPR process:

- The Item type/Agile subclass type is mapped to material type (MARA-MTART) in SAP.
- For a given valid material type, internal number range assignment is already configured in SAP.

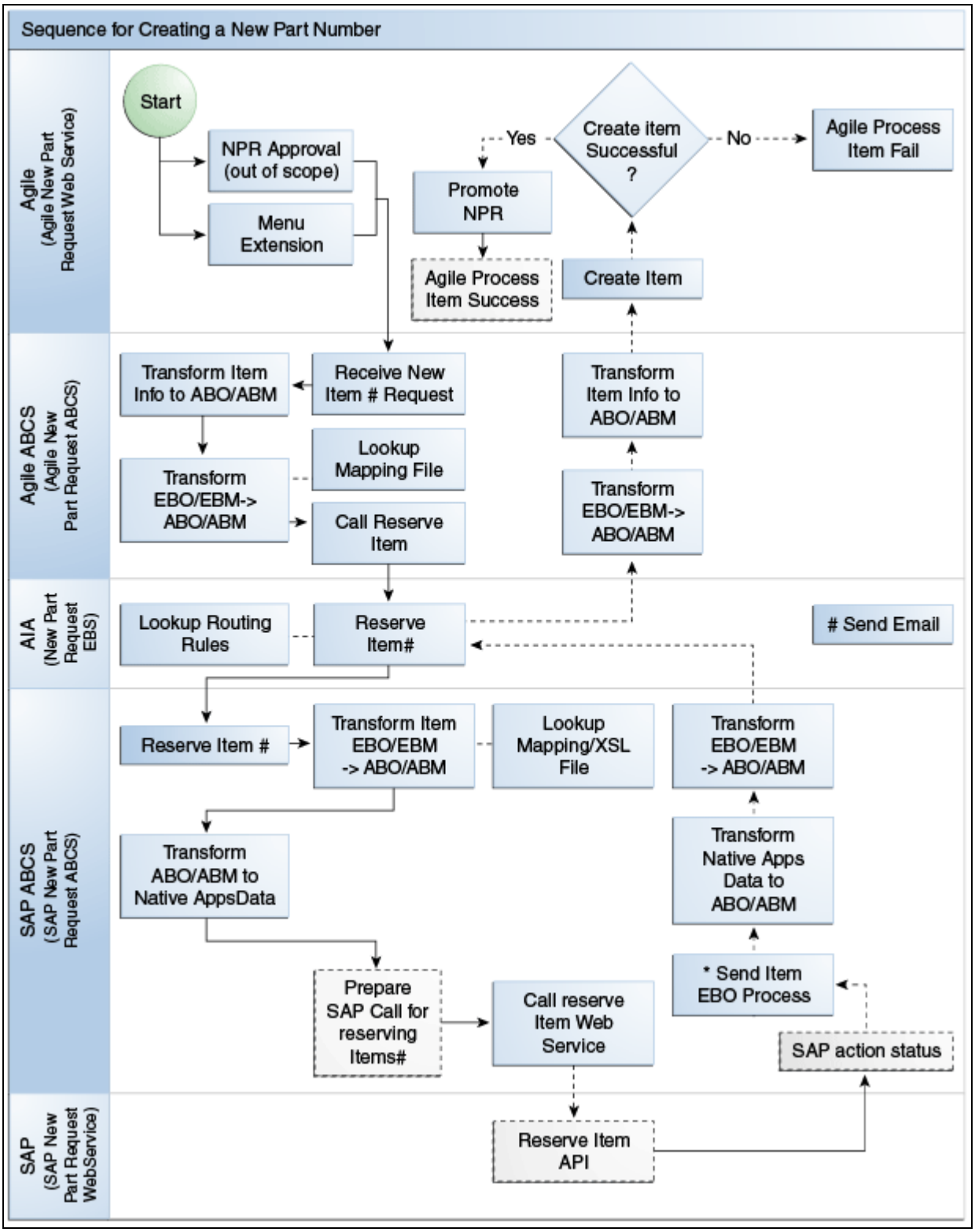
## NPR Integration Sequence

This figure shows the process flow from Agile to SAP for process a new part number request:




### NPR process flow

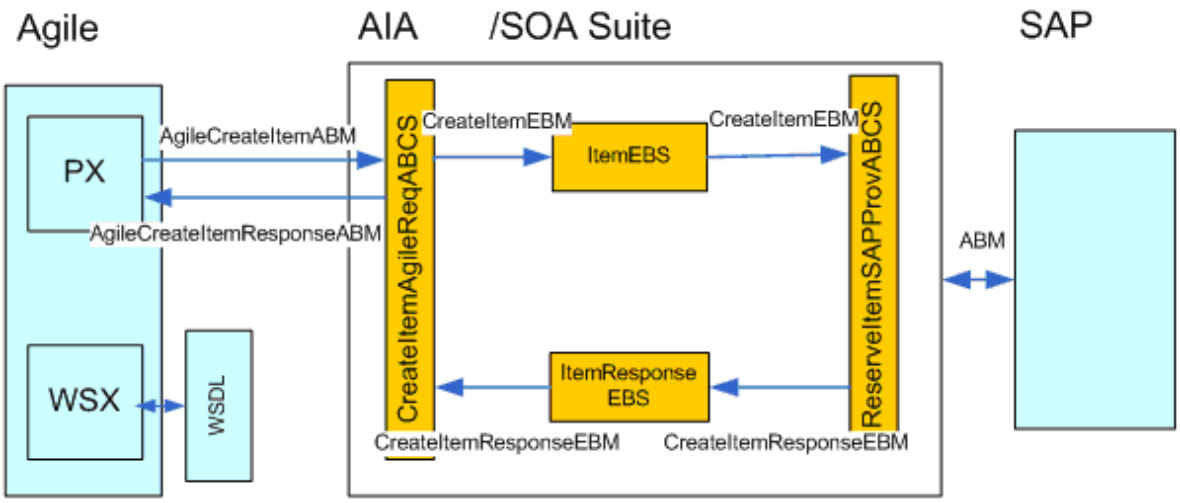
This figure shows the sequence of events that occur when creating a new part number request:




Sequence for creating a new part number

## NPR Integration Flow

The New Part Request process is triggered from a PX. It is a synchronous process and therefore the process is designed based on the Requestor - Provider pattern as shown in this figure:




### Process flow from Agile to SAP for new part number

The PX creates the Agile request ABM (*AgileCreateItemABM*) for the *CreateItemAgileReqABCS* and invokes it:

1. The request *AgileCreateItemABM* is transformed to the CreateItemEBM for NPR process.
2. Using the Asynchronous Message Pattern NPR Process is invoked on the *ItemEBS* by the *CreateItemAgileReqABCSImpl* with the *CreateItemEBM* as input.
3. The *CreateItemEBM* is routed to the *ReserveItemSAPProvABCS* with the *CreateItemEBM* as input.
4. The *ReserveItemSAPProvABCS* will implement the business logic for generating the New Part number in SAP.
5. A response *CreateItemResponseEBM* with New Part Number is returned to the *ItemResponseEBS* and routed back to the *CreateItemAgileReqABCSImpl*.
6. The *CreateItemAgileReqABCSImpl* transforms the response *CreateItemResponseEBM* to Agile response *AgileCreateItemResponseABM*.
7. *CreateItemAgileReqABCS* receives *AgileCreateItemResponseABM* and returns back to the PX.
8. The Part Number from the response ABM (*AgileCreateItemResponseABM*) is returned to the web client and displayed in the Number field in the Create Item screen.

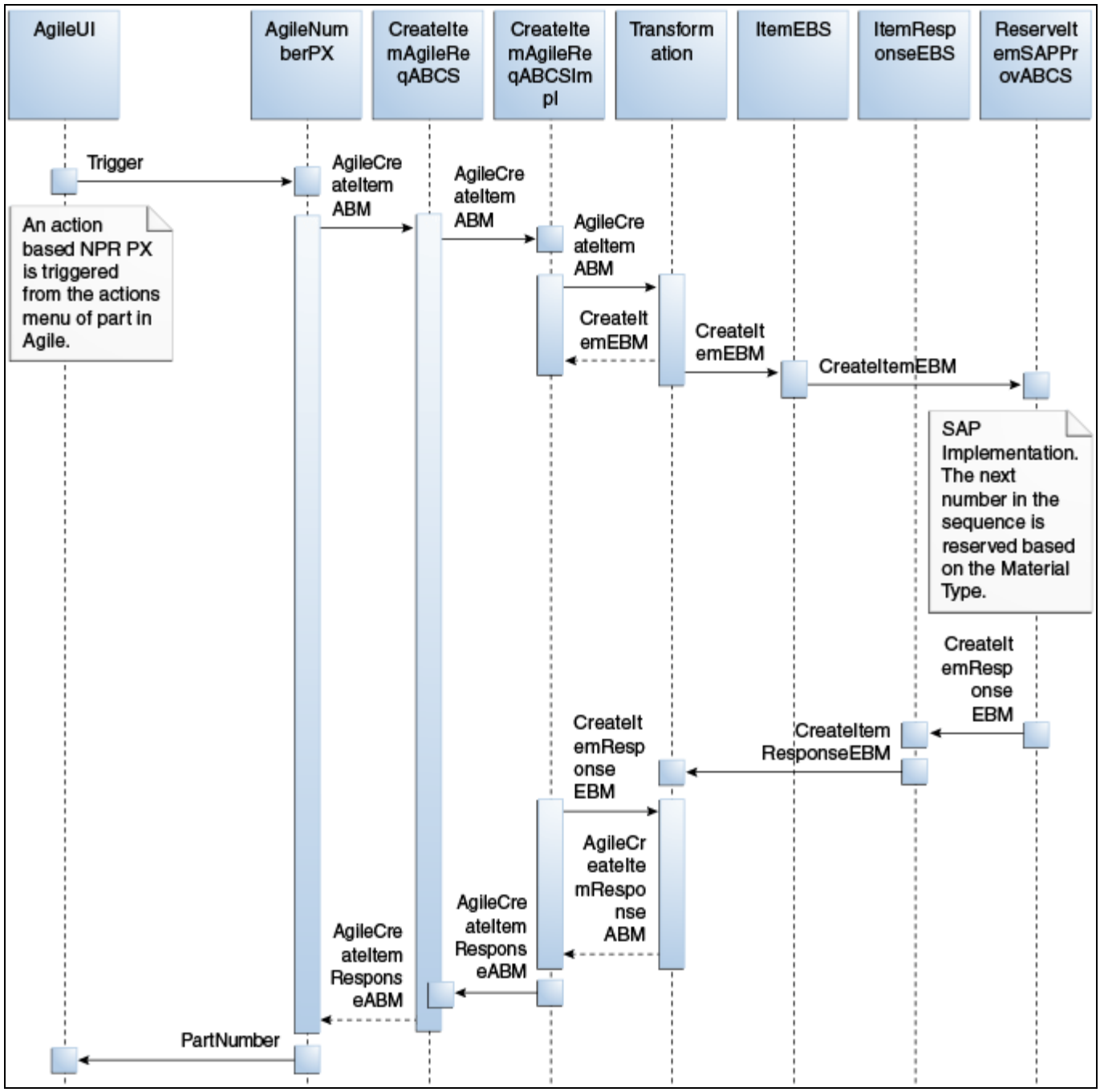
# NPR Services Orchestration

NPR Process itself is an asynchronous process, whereas PX expects the response synchronously. There could be a delayed response from the provider as the *CreateItemAgileReqABCSImpl* is implemented as an *Asynchronous BPEL Composite*. To facilitate this, *Synchronous BPEL Composite CreateItemAgileReqABCS* is used to invoke the *CreateItemAgileReqABCSImpl* and receive the response back. *CreateItemAgileReqABCS* is invoked by the PX synchronously.

The asynchronous message pattern Request-Delayed response pattern with one way calls in Enterprise Business Services (EBS) is used here with EBMID as the correlation ID.

Between the *CreateItemAgileReqABCS* and *CreateItemAgileReqABCSImpl* ABMHeader Id is used as the correlation ID.

This figure shows the sequence of events during NPR services orchestration:




## AutoNumber PX

This table describes the activities involved when creating a new part number request:

| # | Activity | Remarks |
|---|---|---|
| 1 | Agile **NPRAutonumberPX** is triggered | The Agile NPRAutonumberPX is triggered by an Agile user from the Agile Web client as |

| # | Activity | Remarks |
|---|---|---|
| | | part of New Part creation in Agile. |
| 2 | Invoke **CreateItemAgileReqABCS** with **AgileCreateItemABM** as input | NPRAutonumberPX process invokes the CreateItemAgileReqABCS with AgileCreateItemABM ABM as input. |
| 3 | Invoke **CreateItemAgileReqABCSImpl** with **AgileCreateItemABM** as input | NPRAutonumberPX process invokes the CreateItemAgileReqABCSImpl with AgileCreateItemABM ABM as input. |
| 4 | **CreateItemAgileReqABCSImpl** invokes the **ItemEBS** with CreateItem operation | An invoke activity in CreateItemAgileReqABCSImpl transforms the AgileCreateItemABM ABM to CreateItemReqMsg EBM and invokes the CreateItem operation on ItemEBS with CreateItemReqMsg as the input.<br>CreateItemReqMsg will be routed to SAP ReserveItemSAPProvABCS. |
| 5 | **ReserveItemSAPProvABCS** on the SAP first transforms & then calls the SAP service | ReserveItemSAPProvABCS will first transform CreateItemReqMsg into the input of SAP Service to reserve the item in SAP and then calls that service. This would create an Item in using a generic part number generation process (mainly using a sequence).<br>ReserveItemSAPProvABCS invokes ItemResponseEBS with CreateItemResponseMsg as input which is routed back to the CreateItemAgileReqABCSImpl. |
| 6 | **CreateItemAgileReqABCSImpl** first transforms & then returns the response to **CreateItemAgileReqABCS** | CreateItemAgileReqABCSImpl will first transform CreateItemResponseMsg to AgileCreateItemResponseABM and returns the same to CreateItemAgileReqABCS. |
| 7 | **CreateItemAgileReqABCS** returns **AgileCreateItemResponseABM** to NPRAutonumberPX. | |
| **8** | **NPRAutonumberPX** returns the partNumber to New Part Creation UI in Agile | NPRAutonumberPX gets the partNumber from the *AgileCreateItemResponseABM* and returns it to the New Part Creation UI in Agile web client which gets displayed in the Number field. |

# AIA Services for NPR

The process integration for NPR uses the following industry components:

### Core components for NPR

| | |
|---|---|
| EBOs | ItemEBO |
| EBMs | CreateItemEBM<br>CreateItemResponseEBM |
| EBSs | ItemEBS |

### Core Components Locations

| | |
|---|---|
| EBO & EBM XSD files | $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/Item |
| WSDL files | $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/Item |

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

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 AIA Artifacts."

## Agile & SAP Components for NPR

This table contains the Agile and SAP components for NPR:

| Services | Agile (Requester) | SAP (Provider) |
|---|---|---|
| ABMs | AgileCreateItemABM<br>AgileCreateItemResponseABM | BAPI_STDMATERIAL_GETINTNUMBER<br>BAPI_STDMATERIAL_GETINTNUMBERResponse |
| ABCS | CreateItemAgileReqABCS<br>CreateItemAgileReqABCSImpl | ReserveItemSAPProvABCS |
| EBS | ItemEBS | ItemResponseEBS |

### Component Locations

| | |
|---|---|
| ABO XSD files | $AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas |
| ABM XSD files | $AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas<br>$AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/SAP/Release1/Core/ABM/Item |

## Integration Services for NPR

These are the integration services for NPR:

**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 Mediator service**

- CreateItemAgileReqABCSImpl

  Route CreateItemEBM to ReserveItemSAPProvABCS

**ItemResponseEBS Mediator service**

- ReserveItemSAPProvABCS

  Route CreateItemResponseEBM to CreateItemAgileReqABCSImpl

**CreateItemAgileReqABCSImpl**

*CreateItemAgileReqABCSImpl* will transform the Agile message (AgileCreateItemABM) into *CreateItemEBM* & call the routing service to Create Item and get the *CreateItemResponseEBM* response from the SAP system. It then transforms the *CreateItemResponseEBM* response from the routing EBS back to an Agile message (AgileCreateItemResponseABM) & send it to *CreateItemAgileReqABCS* which returns it to the calling Agile PX.

**Flow**

Receives *AgileCreateItemABM* from *CreateItemAgileReqABCS*, which gets the same from NPRAutonumberPX process extensions.

1. Transforms this message into the CreateItemEBM -

Populates EBM header

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

Validates Required fields

Maps to EBM

2. Call Target 'ItemEBS' Service with operation CreateItem
3. Transform CreateItemResMsg Response EBM(CreateItemResponseEBM) to AgileCreateItemResponseABM Agile message.
4. Send 'AgileCreateItemResponseABM back to CreateItemAgileReqABCS, which in turn sends it back to the respective PX.
5. CreateItemAgileReqABCSImpl has the following transformation:
    a. AgileCreateItemABM_to_CreateItemEBM2. CreateItemResponseEBM_to_AgileCreateItemResponseABM

**ReserveItemSAPProvABCS**

This is a single operation service. This accepts a Material Type of an Item information message as a request and returns a response.

In Agile to SAP flow, ReserveItemSAPProvABCS is used for transforming CreateItemEBM into BAPI_STDMATERIAL_GETINTNUMBER ABM, which invokes the Reserve Item operation in SAP.

In return flow, BAPI_STDMATERIAL_GETINTNUMBER sends BAPI_STDMATERIAL_GETINTNUMBERResponseABM which is transformed by ReserveItemSAPProvABCS into CreateItemResponseEBM.

This service is implemented as asynchronous BPEL Composite.

## NPR Integration Customization Points

These are the NPR integration customization points:

### Agile

| Property | Name | Description |
|---|---|---|
| CreateItemAgileReqABCSImpl<br><br>(Agile Process Item requestor flow) | AgileCreateItemABM_to_CreateItemEBM_Custom.xsl | Request ABM to Request EBM |
| | CreateItemEBM_EBMHeader_Custom.xsl | Request EBM to EBM Header |

### SAP

| Property | Name | Description |
|---|---|---|
| ReserveItemSAPProvABCS | GenerateItemEBMtoABM_Custom.xsl | Request EBM to Request ABM |
| | GenerateItemABMtoEBM_Custom.xsl | ResponseABM to ResponseEBM |

## Properties and DVMs for NPR

The ITEM_PRIMARYCLASSIFICATIONCODE property must be set for the New Part Request in the AIAConfigProperties.xml to work.

# Chapter 4: Process Integration for Item Attribute Update

This chapter discusses:

- Item Attribute Update Process
- IA Update Process Integration Solution Assumptions
- IA Update Integration Sequence
- AIA Services for IA Update
- IA Update Integration Customization Points

Required as part of the Manufacturing Update process for product information synchronization, the Item Attributes information from ERP system is required to be updated in Agile. The term Item applies to both Parts and Documents in Agile considering the documents are integrated in to SAP as part object only and not as Document Info Record (DIR). DIR is not in scope for this version of integration.

Since item attribute information can exist in multiple plants in the ERP system, it is supported only for cases where the system can accurately determine which plant in ERP to pick up item attribute information from.

## Item Attribute Update Process

The update of Item Attribute information from ERP to PLM is done event based and in near real time.

The following steps are executed:

The process kicks off when configured field/s are modified and saved on an item in SAP.

1. Retrieve the attributes that have changed on the item in SAP. This data is retrieved from one or more organizations (plants) as per site-org mappings or from the configured single organization.
2. Update the information into Agile.

The input consists of the item attribute values, which need to be updated in Agile. Unique identifiers for the item, along with values for all the attributes mapped for transfer from ERP to Agile are provided as input.

As of Agile 9.3.1, an item number is the only attribute that is required for uniquely identifying the item to be updated. A planned enhancement in the next major release of Agile PLM would require that the item revision be specified as part of the unique key as well, since all the item attributes could potentially be under revision control with that enhancement. Even in Agile 9.2.2.2, Item Description and Mass can have different values for different revisions of the item.

A unique item revision in Agile is composed of a Rev Number as well as a Change Order number (since it's possible for certain types of Change Orders in Agile to revise an item without changing its revision number). In the context of integration, however, updates to item attributes applies to the latest released revision in most cases.

The following guidelines are followed when querying an item from Agile for update:

- In the first release of integration, developed on Agile 9.2.2.5, only the item number is used to query the item to be updated. The item number will pull up the latest released revision of the item by default, and all updates are made to the latest released revision of the item.
- A future release of the integration should plan on enhancing this capability to accept a revision number (and optionally, a Change number) along with the item number to pull up a unique item revision for update (if a Change number is not specified, the latest released Change corresponding to that revision is assumed).

## Processing

Item attribute update is done near real time, or batch. In order to achieve this implementation needs one time configuration of change pointers and *Application Link Enabling* (ALE) in SAP. For more details please refer to Chapter 5: Implementing the Process Integration Pack Section: Setup for SAP.

As part of this configuration, all the fields that need to be updated back in to Agile will be configured such that any changes saved to these fields on an item would trigger the update process immediately. For the batch process, it is required to schedule the **IDOC** as a background job.

The moment user saves the changes on item in SAP the change pointers configuration would trigger an event. The event will push the data to the configured partner (ALE configurations) and in this case it is Agile system. All the changes made on the item are captured and will be updated on the specified item in Agile. This process triggers once for each item change save event.

### Exceptions

In all cases, the exception message displayed to the user must contain the item number that was being updated along with other supporting information as indicated.

Insufficient privilege to discover, read, or update the item.

- Invalid field value
    - Invalid list value

    - Invalid data format for text, money or date fields
- Item does not exist

# IA Update Process Integration Solution Assumptions

If item information needs to be retrieved from multiple plants in SAP (ERP) system, the following constraints must be met in order to support this process:

Agile multi-sites is configured

1. There is a one-to-one mapping between Agile sites and ERP plants
2. The attribute(s) to be updated with ERP data are on the Sites tab of the item

This release of the integration intends to solve a use case where any given Change is interfaced to only one destination system.

The following exception conditions are tested to make sure that they return user-friendly error messages. In all cases, the exception message displayed to the user must contain the Item number that was being updated and the step at which the error occurred, along with other supporting information as needed.

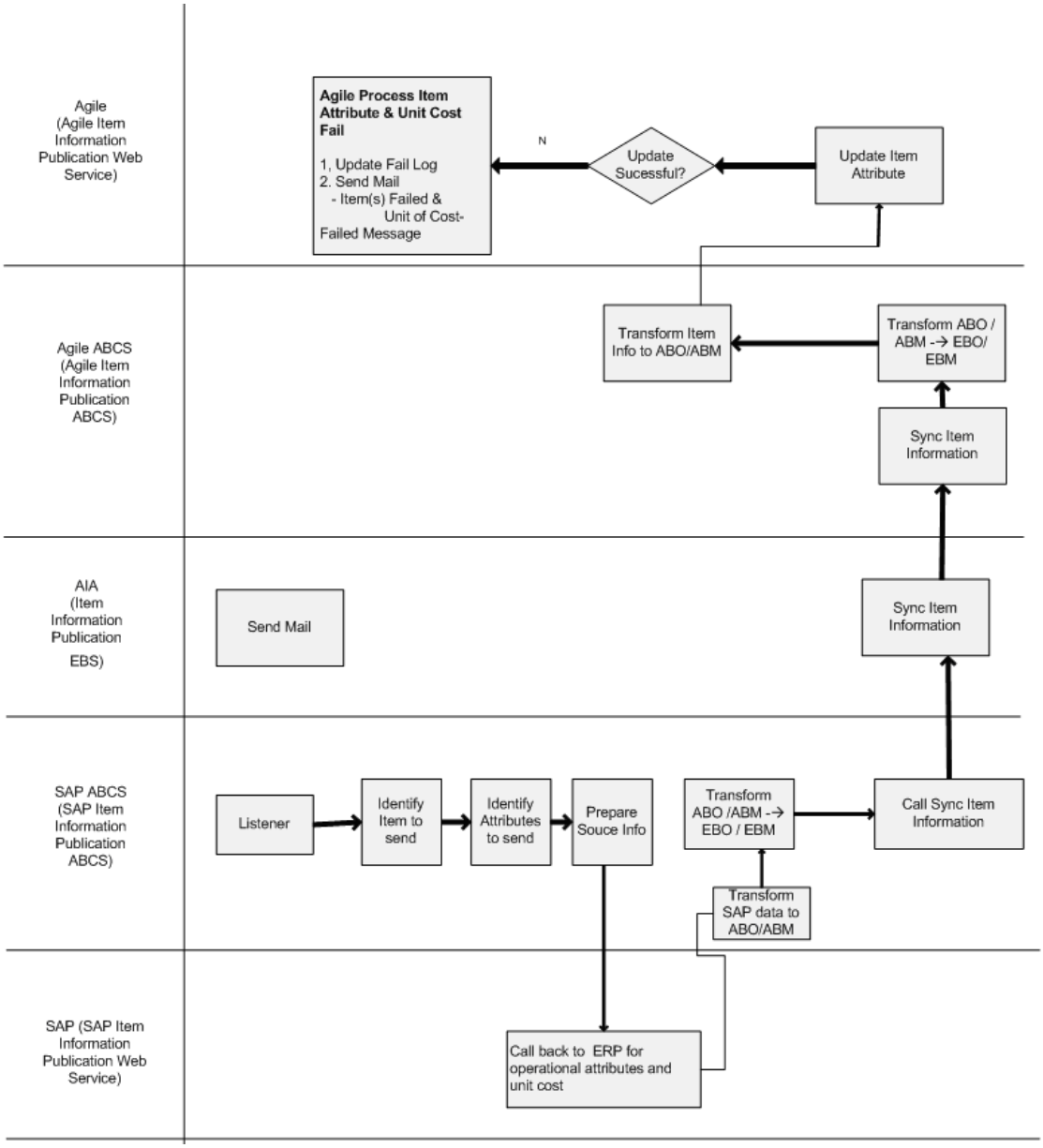
Insufficient privilege to discover, read, or update attributes on the Item object

Invalid field value (indicate the field for which the value is incorrect, and the value that is being passed on to the field)

- Item object does not exist

**Note**: New Item Creation is not part of the scope for this activity. It is assumed that the item exists in both Agile and SAP, by means of a prior New Part Introduction or independent offline load processeIA Update Integration Sequence

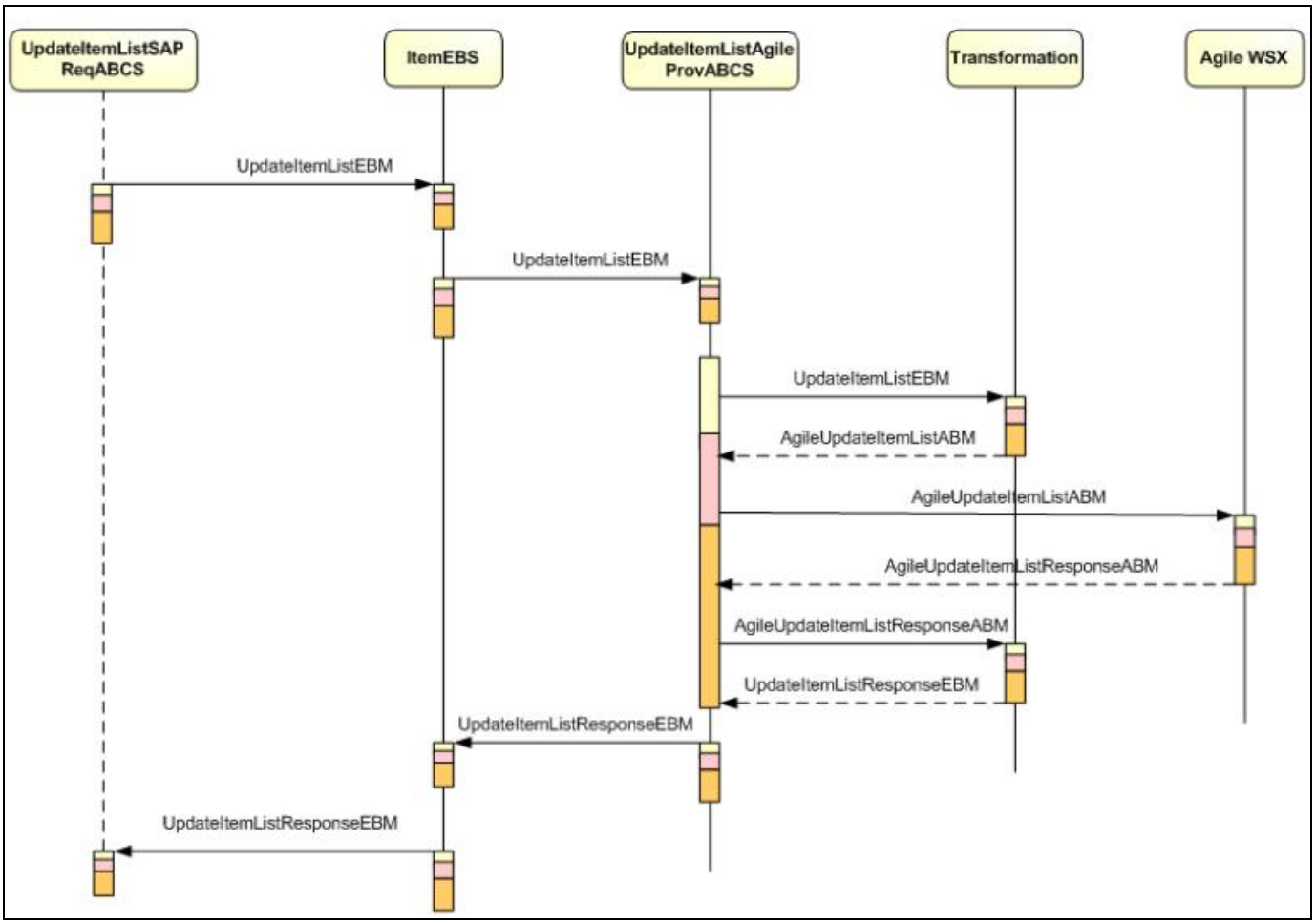
This figure shows the sequence of events that occur during IA update:




## IA update sequence

## IA Update Services Orchestration

This sequence diagram shows the process flow for IA update process flow:




### Sequence diagram for IA update process flow

This table describes the activities involved in the IA update process:

| # | Activity | Description |
|---|---|---|
| 1 | SAP Item Operational Attribute Update Action Trigger | SAP Item Operational Attribute Update Action trigger Invokes the UpdateItemListSAPReqABCS |
| 2 | **UpdateItemListSAPReqABCS** calls **UpdateItemListSAPReqABCSImpl** | UpdateItemListABM will be passed to UpdateItemListSAPReqABCSImpl as the input |
| 3 | **UpdateItemListSAPReqABCSImpl** invokes **ItemEBS** | An invoke activity in UpdateItemListSAPReqABCSImpl invokes the |

| # | Activity | Description |
|---|---|---|
| | | UpdateItemList operation on ItemEBS with UpdateItemListEBM as input |
| 4 | **ItemEBS** routes **UpdateItemListEBM** to **UpdateItemListAgileProvABCSImpl** | ItemEBS routes UpdateItemListEBM as input to UpdateItemListAgileProvABCSImpl |
| 5 | **UpdateItemListAgileProvABCSImpl** invokes Agile's Item Operational Attribute Update Web Service | UpdateItemListAgileProvABCSImpl transforms the UpdateItemListEBM to AgileUpdateItemListABM and invokes UpdateItem service operation on Agile Web Service to update Item cost related attribute information from SAP to Agile.<br>The AgileUpdateItemListResponseABM is returned back to UpdateItemListAgileProvABCSImpl |

# AIA Services for IA Update

The process integration for IA Update uses the following industry components:

### Core Components for AIA Update

| | |
|---|---|
| EBOs | ItemEBO |
| EBMs | UpdateItemListEBM<br>UpdateItemListResponseEBM |
| EBSs | ItemEBS<br>ItemResponseEBS |

### Core Components Locations

| | |
|---|---|
| EBO & EBM XSD files | $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/Item |
| WSDL files | $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/Item |

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

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 AIA Artifacts."

## SAP & Agile Components for IA Update

This table lists the SAP & Agile components for IA updates:

| Services | SAP (Requester) | Agile (Provider) |
|---|---|---|
| ABMs | UpdateItemListABM | AgileUpdateListABM<br>AgileUpdateListResponseABM |
| ABCS | UpdateItemListSAPReqABCSImpl<br>UpdateItemListSAPReqABCS | UpdateItemListAgileProvABCSImpl |
| EBS | ItemEBS | ItemResponseEBS |

### Component Locations

| | |
|---|---|
| ABO XSD files | $AIA_HOME/AIAMetaData/AIAComponents/ApplicationObjectLibrary/Agile/V1/schemas |

## Integration Services for IA Update

These are the integration services for IA Update:

1. ItemEBS

ItemEBS is the Enterprise Business Services which exposes the following operations related to the Item Attribute Update Integration on the ItemEBO.

The following are the routing rules:

**ItemEBS Mediator service**

- UpdateItemListSAPReqABCSImpl

  Route UpdateItemListEBM to UpdateItemListAgileProvABCSImpl

2. UpdateItemListSAPReqABCSImpl

This is called when there is a need to update multiple items in Agile with all the operational attributes as well as attributes such as unit cost from SAP. This is a push from SAP to the Agile.

The requester ABCS, defined as a "synchronous" process, receives a list of Item Ids from the SAP Change Pointer program. The lists of Ids are those which have last_update_date greater than the last run date of the Change Pointer.

- The requester BPEL Composite filters the list of Ids that are present in the xref tables in the FMW layer. This provides a list of Items that were actually from the Agile system alone. The original list of Item Ids may be those which are from non-Agile sources as well.
- The BPEL Composite then makes an Mediator call out for end point virtualization. This Mediator layer has an SAP Adapter which calls the BAPI that provides the Item details that is needed to be sent out. This is the Item ABM.
- A transformation converts the ABM to an EBM.
- An Asynchronous request-delayed response call is made to the ItemEBS with the ItemListEBM. This is routed to the appropriate provider.
- The BPEL instance gets invoked when the asynchronous call gets back from the provider and provides the status of the transaction back to the caller concurrent program (since that is a synchronous call).
- The concurrent program logs the status of this call.

### 3. UpdateItemListAgileProvABCSImpl

UpdateItemListAgileProvABCSImpl is used to facilitate the communication between ItemEBS and Agile web service used for updating the Items' operational attributes, item cost information in batch mode in Agile.

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 side) to update Items in Agile.
- AgileUpdateItemListResponseABM is received on successful execution of Coarse Grained API.
- Transform operation is called to convert the AgileUpdateItemListABM to UpdateItemListResponseEBM, which is returned as output of this BPEL Composite.
- If the UpdateItems service operation fails on the Agile side, a new Fault is generated and is sent across with appropriate error message.

# IA Update Integration Customization Points

These are the IA Update integration customization points:

### SAP

| Property | Name | Description |
|---|---|---|
| UpdateItemListSAPReqABCSImpl | XFormABMToEBM_Custom.xsl | Custom transformations for Item Request ABM to Request EBM |

### Agile

| Property | Name | Description |
|---|---|---|
| UpdateItemListAgileProvABCSImpl<br>(Agile update item attributes provider flow) | AgileUpdateItemListResponseABM_to_UpdateItemListResponseEBM_Custom.xsl | RespABM to RespEBM (custom element) |
| | AgileUpdateItemListResponseABM_to_UpdateItemListResponseEBM_Impl.xsl | RespABM to RespEBM (main) |
| | UpdateItemListEBM_to_AgileUpdateItemListABM_Impl.xsl | ReqEBM to ReqABM (main) |

# Essential Properties and DVMs for IA Update

The following mandatory DVMs and Properties (in AIAConfigProperties.xml) should be set for the Item Attribute Update process to flow to work:

- AGILE_SITE_TARGET_MAPPING
- AGILE_TARGET_SITE_MAPPING
- ITEM_PRIMARYCLASSIFICATIONCODE

# Chapter 5: Implementing the Process Integration Pack

This chapter discusses:

- Setting up the Participating Applications
- Loading Cross Reference Data
- Configuring the integration
- Domain Value Maps
- Application Interfaces
- Handling Errors

## Prerequisites

The Agile PLM integration is deployed through the AIA Installer. For proper functioning, the integration requires certain settings and configurations in Partner Applications like Agile PLM and SAP ERP, and in AIA configuration property files.

SOA 11.1.1.5 should be installed and relevant patches are to be applied.

AIA Foundation Pack Foundation Pack 3.1 should be installed with relevant patches applied if any.

- Agile PLM must be installed and configured accordingly.
- Oracle Application adapter for SAP need to be installed.
- All the required webservices for the corresponding BAPI's and Idoc's of SAP need to be generated from Oracle Application adapter.
- SAP environment must be installed and configured accordingly.
- SAP Version should be 4.7 and above.

## Setting up the Participating Applications

Before enable the Agile to SAP integration, you must set up the following participating applications:

- AgilePLM
- SAP
- Cross-reference data

## Agile PLM

After installation of Agile pre-built integration, the Agile Administrator is required to setup Agile Content Services. The following are the settings/configurations:

This section discusses the following post-installation configurations using Java client for the Agile PLM SAP pre-built integration:

Create new JMS Destination.

1. Create new events for ECO, MCO and SCO.
2. Create new subscribers for ECO, MCO and SCO.
3. Creating Auto Numbers
4. Updating Change Order Page Two Fields
5. Configuring Sites

6. Configuring Site Attributes
7. Configuring and Deploying PX

## Creating New JMS Destination

To create new JMS Destination:

1. Login to Agile Java Client using the login credentials provided.
2. Navigate to Admin, System Settings, Agile Content Service, Destinations.
3. Click  (New) button to create a new JMS Destination.

| Parameters | Variables |
|---|---|
| Name | SOA Application Server |
| Description | SOA Application Server |
| Protocol | JMS |
| Username | SOA Application Server Username |
| Password | SOA Application Server Password |
| Provider Context Factory | weblogic.jndi.WLInitialContextFactory |
| Connection Factory | jms/aia/AIAAgilePLMECOCF |
| Default Provider URL | t3://<soa_hostname>:<port> |
| Destination Name | jms/aia/AIAAgilePLMECOJMSQueue |

**Note:** If you are pointing Agile to a clustered FMW environment, the **Default Provider URL** needs to contain all possible URLs in the cluster separated by a comma. For example, *t3://<soa_hostname1>:<soa_port1>,<soa_hostname2>:<soa_port2>*.

4. Test the JMS Destination. The test needs to be successful.

## Create New Events for ECO, MCO and SCO

### To create new events for ECO, MCO and SCO

1. Navigate to Admin, System Settings, Agile Content Service, Events.
2. Click  button to create new Event(s).
   - **a.** Enter **Name** and **Description**.
   - **b.** Choose relevant workflows.
     - i. For SCO choose ***Default Site Change Orders***.
     - ii. For ECO choose ***Default Change Orders***.
     - iii. For MCO choose ***Default Manufacturer Orders***.
   - **c.** Choose the workflow status as ***Released***.

## Creating New Subscribers for ECO, MCO and SCO

### To create new subscribers for ECO, MCO and SCO:

1. Navigate to Admin, System Settings, Agile Content Service, Subscribers.
2. Click  button to create new Subscribers for ECO, MCO and SCO.
   - **a.** Enter a **Name** and a **Description** for subscribers.
   - **b.** Associate the workflow as Automated Transfer Orders(ATO).
   - **c.** Choose ***All Change Orders***, ***All Manufacturer Orders***, and ***All Site Change Orders*** as **Criteria** for ECO, MCO and SCO respectively.
   - **d.** Choose appropriate **Event** that you created for ECO, MCO and SCO.
   - **e.** Click **OK**.
3. Select the Subscriber Details tab to add the subscriber detail information.
4. Click  button to create a new row.

   This opens Subscriber Detail window.

   - **a.** Click **Destinations** field and select JMS Destination that was created.
   - **b.** Click **Filters** field and choose the following filters for **Filters** field using **> button** in the pop up that appears.
     - i. ECO: ***Default Change Order Filter***
     - ii. MCO: ***Default Manufacture Order Filter***.

iii. SCO: ***Default Site Changes Order Filter***.

**Note**: Also add the ***Default Item Filter*** as filter for all the three; ECO, MCO and SCO.

**c.** Click Role(s) field and select all the roles using **> button** in the pop up that appears.

**d.** Select ***aXML*** as **Format**.

**e.** Select ***English*** as **Language**.

**f.** Select ***All*** as **Site**.

5. Select each subscriber at a time and click  button to enable all the created subscriptions.
6. Navigate to Admin, System Settings, Agile Content Service, Filters.
7. Select Default Item Filter and double-click.
8. Navigate to the Filters tab and append Sites to the View Tabs field value.
   **a.** Choose ***Tab and Items*** in the **BOM Options** field.
   **b.** Select ***All Levels*** checkbox in the **BOM Level** field.
   **c.** Choose ***Tab and Manufacturer Parts*** in the AML Options field.
9. Select Default Change Order Filter and double-click.
10. Navigate to Filters tab and,
    **a.** Choose **Redline Changes Only** option as ***No***.
    **b.** Choose **Affected Items Options** as ***Tab and Items***.
11. Select Default Site Change Order Filter and double-click.
12. Navigate to Filters tab and,
    **a.** Choose Redline Changes Only option as ***No***.
    **b.** Choose Affected Items Options as ***Tab and Items***.
13. Navigate to Admin, User Settings, Privileges.
14. Click  button to create Modify privileges for Changes, MCO and SCO.
    **a.** Enter **Name** and **Description**.
    **b.** Select ***Yes*** for **Enabled**.
    **c.** Select ***Modify*** for **Privilege**.
    **d.** Click **Criteria** field and choose the relevant criteria for each of the Changes, MCO and SCO.
    **e.** Click **Applied to** field and select all attributes, including the invisible/disabled attributes.
    **f.** Click **OK**.

This opens the privilege screen of the entity for which you set privileges. You now need to assign roles to the privileges that you have created for Changes, MCO, and SCO.

15. Navigate to Where Used tab.
16. Click  button and select the roles in the Select Roles pop up.
17. Click OK to assign these roles to the privilege that you have created.
18. Navigate to Admin, User Settings, Privileges and double click Read.
19. Click  button in Privileges to Read screen to create Read privileges for MCO.
20. In the Create Privilege pop up that opens
    a. Enter **Name** and **Description** for the privilege in respective fields
    b. Choose ***Yes*** for **Enabled** field.
    c. Choose ***Read*** for **Privilege** field.
    d. Choose ***All Manufacturer Orders*** for **Criteria** field.
    e. Choose both visible and invisible/disabled attributes in the **Applied to** field.
    f. Click **OK**.

    This opens privilege information screen for MCO.
21. Navigate to Where Used tab.
22. Click  button to select roles and assign to the created privilege.
23. Modify the Read Changes and Read Items to include Admin user.
24. Click OK.
25. Navigate to Admin, Data Settings and double click Classes
26. Double click Parts in Classes window.
27. Navigate to User Interface Tabs tab.
28. Double click Page Two.
29. Navigate to Attributes Page Two tab.
30. Check for the attribute Numeric01 and double click it and change the Name to *ERP Manufacturer Cost*.

**Note**: If the user does not have Sites version on the Agile application then, perform the steps till the end of this section. Else move to the next section.

31. Navigate to Admin, Data Settings and double click Classes.

32. Double click Parts in Classes window.

**Note**: Repeat the process also for Document.

33. Navigate to User Interface Tabs tab.
34. Double click Page Two.
35. Navigate to Attributes Page Two tab.
36. Check for the MultiList01 and double click it and change the Name to *OrganizationCodes*.

    You also need to assign the values that are specific to sites or organization codes on the Agile side.

37. Click New List button to create the List of values.
38. Click OK.

    This opens Create a new list value screen.

39. Enter the values for Name and Description specifying the list of Organizations used.
40. Click Add.
41. Navigate to List tab.
42. Click  button to add the batch list.
43. Add batch list values in the pop up window.
44. Click OK.
45. Ensure that Page Two on these classes in enabled.

## Creating Auto Numbers

To create auto numbers:

1. Navigate to Admin, Data Settings, AutoNumbers.
2. Click  button to create auto numbers for NPR Document Number.
3. Enter the following fields in the Define the AutoNumber screen.
   a. Enter name as NPR Document Number
   b. Choose ***Yes*** for **Enabled**.
   c. Choose ***Custom*** for **Type**.

d. Choose ***Document*** for **Where Used**.

e. Choose ***com.oracle.aia.npr.NPRAutoNumber*** for **Custom AutoNumber**.

f. Click **OK**.

4. Click  button to create auto numbers for NPR Part Number.

5. Enter the following fields in the Define the AutoNumber screen:

a. Enter name as NPR Part Number

b. Choose ***Yes*** for **Enabled**.

c. Choose ***Custom*** for **Type**.

d. Choose ***Part*** for **Where Used**.

e. Choose ***com.oracle.aia.npr.NPRAutoNumber*** for **Custom AutoNumber**.

f. Click **OK**.

## Updating Change Order Page Two Fields

To update Change Order Page Two fields:

1. Navigate to Admin, DataSettings, Classes.
2. Double click Classes and navigate to Changes, Change Orders.
3. Double click Change Orders and navigate to User Interface Tabs in Change Order window.
4. Double click Page Two and Navigate to Attribute Page Two tab.
5. Select Text02 and double click.
6. Set Visible to *Yes* in Attributes Text02 window.
7. Click Save.
8. Navigate to Admin, User Settings, Privileges and double click Read.
9. Click  to create a new Read privilege to read all the ECO fields.
10. Select required fields in Applied To field in Read Changes Page Two window.
11. Click OK.
12. Navigate to Where Used tab.
13. Click  to select roles and select appropriate users in Select Roles window.

    Use the same procedure that you used for setting Read privileges and set Modify privilege for ECO fields.

## Configuring Sites

You need to configure sites only if the Agile application is enabled with Sites version.

To configure sites:

1. Login to web client using http://<AGILE_HOSTNAME>/Agile/PLMServlet.
2. Click  button.
3. Select Sites from the options.
4. Select *Site* for Type and enter Site Name.
5. Click Finish.

## Configuring Sites Attributes

To configure Site attributes:

1. Navigate to Admin, Data Settings, Classes.
2. Double click Classes and navigate to Items, Parts in Classes window.
3. Double click Parts and navigate to User Interface Tabs, Sites.
4. Double click Sites and navigate to Attributes: Sites tab.
5. Select Numeric01 and double click and in the window that opens
   a. Change **Name** to ***UnitCostAttribute***.
   b. Set **Visibility** to ***Yes***.

# Setting up SAP

SAP ALE (Application Link Enabling) interface is a part of integration layer within SAP's Business Framework Architecture (BFA). It is a component based architecture that enables business process integration and asynchronous data communication either between two or more SAP systems or between SAP and external systems.

Application systems are loosely coupled in an ALE integrated system and the data is exchanged asynchronously. Consistency is maintained as data is exchanged between application systems in a controlled manner.

The following are the details that have to be maintained for ALE Configuration. Each step is dealt in detail with the help of a screen shots.

## Defining Logical System

1. Assigning Logical Systems
2. Modeling Distribution Scenario
3. Defining RFC Destination
4. Defining Ports
5. Defining Partner Profiles
6. Activation of Change Pointers Globally

## Defining Logical System

1. Enter transaction code SALE in the transaction space. Go to *Sending and Receiving Systems* > *Logical Systems*. Select Define Logical System option.
2. In *Change View "Logical System" Overview* screen, select **New Entries** option.
3. In *New Entries: Overview of Added Entries* screen, type in the **logical system names** which have to be assigned to the systems that are in communication and **save** the entries.

## Assigning Logical Systems

1. Enter the transaction code SALE. Go to *Sending and Receiving Systems* > *Logical Systems*. Select **Assign Client to Logical Systems** option.
2. In *Change View "Clients": Overview* screen, select **New Entries** option.
3. As per the selection, user would be redirected to the *New Entries: Details of Added Entries* screen. Enter the Client & Logical System defined earlier. Also enter the other appropriate values and **save** the data.

## Modeling the Distribution Scenario

1. Enter transaction code **BD64**, switch to **edit** mode and select **Create Model View**.
2. In *Create Model View* screen, enter **Short text, Technical name, Start date** and **End Date**.
3. Select the Model view and Add Message Type 'MATMAS. Save the entries. Select the Model view and then choose the EDIT option in the menu bar. Choose model view & then Distribute. The Model view is Distributed now.

## Defining RFC Destination

1. Enter transaction **SM59**, choose **TCP/IP connections** and click on **Create** button in the menu..
2. In *RFC Destination* screen, enter data in **RFC destination, Connection type** as **T** and **Description**, respectively.

a. Click **Enter** to enable TCP/IP connection related fields.
b. Choose **Registration** and provide **Program ID** and **save** the entries.

## Defining Ports

1. Enter transaction code **WE21** in the transaction space.
   c. Select **Transactional RFC** and choose **Create** option.
2. In *Ports in IDoc processing* pop-up, select **Generate port name** option.
   d. Provide **port name**, if **Own port name** option is selected.
3. In *creating a tRFC port* screen, enter **Description, RFC Destination** and select an appropriate **Version** and **save** the entries.

## Defining Partner Profiles

1. Enter transaction code **WE20** in the transaction space.
2. From the drop down list, choose **Partner Type LS** and select **Create** option.
3. In *Partner profiles* screen, add message type **SYNCH** in **out bound parameters** and **MATMAS** in **inbound parameters**. Provide any other relevant information, like ports, idoc type and **save** the inputs.

   *Partner profiles: Outbound parameters* screen displays the parameters and the setup of *ALE Configuration* is completed.

## Activating Change Pointers Globally

This section explains about the configuration of **Change Pointers** for triggering **MATMAS Idoc** to send **Material Attributes** from SAP to Agile.

Enter transaction code **BD61** in the transaction area. In the *Activate Change Pointers Generally* screen, select **Change Pointers** checkbox and save the transaction.

**Activation of Change Pointers for MATMAS Message Type**

Enter transaction code BD50. In *Change View "Activate Change Pointers for Message Type": Overview* screen, select MATMAS message type checkbox and Save the transaction.

**Activation of Change pointers for Individual fields in Material Master**

Enter transaction code **BD52**.In the work area gives the Message Type as MATMAS and click continue. Delete those fields for which Change Pointer activation are not required. Save the transaction.

## Reference Material Setup in SAP

To create an Item in SAP, reference material need to be configured based on the Material type. In case if the Item need to be created in more than one Plant then reference material need to be maintained in all the Plants .The purpose of this reference material is to default all the mandatory values which are needed for creation of material in SAP. This reference material varies from customer to customer based on their implementation setup.

# Loading Cross Reference Data

This pre-built integration uses the xref_data table present in Metadata database (MDS) to maintain a cross-reference between Agile and SAP. This cross-reference information helps map Agile Parts/Documents/Change Orders to SAP Items/Change Orders.

There are two main virtual tables in the AIA_XREF schemas XREF_DATA table that maintain this cross-reference information.

CHANGE_CHANGEID - maintains all the Change Order information

ITEM_ITEMID - maintains Item information.

Example:

- A Change Order ECO001 contains two revised items
  - P0001 in site 'Detroit'
  - P0002 in site 'Michigan'
- Both these revised items have component items.
  - P0001 has component items C0001 and C0002
  - P0002 has component items C0003 and C0004
- When this Change Order is released from Agile to the SAP, the following entries are made in the CHANGE_CHANGEID virtual table.

| XREF_TABLE | XREF_COLUMN | ROW_NUMBER | VALUE |
|---|---|---|---|
| CHANGE_CHANGEID | AGILE_01 | E45E015046AF11DD9F2E436FB39961A8 | ECO001::DETROIT |
| CHANGE_CHANGEID | COMMON | E45E015046AF11DD9F2E436FB39961A8 | 2d3738333032373031323838373536 31 |
| CHANGE_CHANGEID | SAP_01 | E45E015046AF11DD9F2E436FB39961A8 | 11075 |
| CHANGE_CHANGEID | AGILE_01 | E45E015046AF11DD9F2E436FB39961A9 | ECO001::MICHIGAN |

| XREF_TABLE | XREF_COLUMN | ROW_NUMBER | VALUE |
|---|---|---|---|
| CHANGE_CHANGEID | COMMON | E45E015046AF11DD9F2E436FB39961A9 | 2d3738333032373031 32383837353632 |
| CHANGE_CHANGEID | SAP_01 | E45E015046AF11DD9F2E436FB39961A9 | 11076 |

- The first row entry is made by the Agile BPEL flow for Change Order number ECO001 that is created on an item, which belongs to a site mapped to the 0001 Plant in SAP.
- The second entry (Common) is also created by the Agile BPEL flow. It indicates the common "business component id" for this particular in integration entity and is used for linking change orders to SAP Change Orders.
- The third row entry represents the SAP change id corresponding to the Agile change order ECO001. Another set of entries will be made for the change order ECO0001 for site PAV0.

- For each revised item and component items in the Change Order, entries will be made into the ITEM_ITEMID virtual table.

For example: For ECO001, the following entries are made in the ITEM_ITEMID table:

| XREF_TABLE | XREF_COLUMN | ROW_NUMBER | VALUE |
|---|---|---|---|
| ITEM_ITEMID | AGILE_01 | 0078BE703EC711DDBF9CA7AA7FE3BDFB | P0001::DETROIT |
| ITEM_ITEMID | COMMON | 0078BE703EC711DDBF9CA7AA7FE3BDFB | 353138353737393537 32383638303435 |
| ITEM_ITEMID | SAP_01 | 0078BE703EC711DDBF9CA7AA7FE3BDFB | 66247::204 |
| ITEM_ITEMID | AGILE_01 | 0078BE703EC711DDBF9CA7AA7FE3BDFC | P0002::MICHIGAN |
| ITEM_ITEMID | COMMON | 0078BE703EC711DDBF9CA7AA7FE3BDFC | 353138353737393537 32383638303436 |
| ITEM_ITEMID | SAP_01 | 0078BE703EC711DDBF9CA7AA7FE3BDFC | 66248::207 |
| ITEM_ITEMID | AGILE_01 | 0078BE703EC711DDBF9CA7AA7FE3BDFD | C0001::DETROIT |
| ITEM_ITEMID | COMMON | 0078BE703EC711DDBF9CA7AA7FE3BDFD | 353138353737393537 32383638303437 |

| XREF_TABLE | XREF_COLUMN | ROW_NUMBER | VALUE |
|---|---|---|---|
| ITEM_ITEMID | SAP_01 | 0078BE703EC711DDBF9CA7AA7FE3BDFD | 66249::204 |
| ITEM_ITEMID | AGILE_01 | 0078BE703EC711DDBF9CA7AA7FE3BDFE | C0002::DETROIT |
| ITEM_ITEMID | COMMON | 0078BE703EC711DDBF9CA7AA7FE3BDFE | 353138353737393537 32383638303438 |
| ITEM_ITEMID | SAP_01 | 0078BE703EC711DDBF9CA7AA7FE3BDFE | 66250::204 |
| ITEM_ITEMID | AGILE_01 | 0078BE703EC711DDBF9CA7AA7FE3BDFF | C0003::MICHIGAN |
| ITEM_ITEMID | COMMON | 0078BE703EC711DDBF9CA7AA7FE3BDFF | 353138353737393537 32383638303439 |
| ITEM_ITEMID | SAP_01 | 0078BE703EC711DDBF9CA7AA7FE3BDFF | 66251::207 |
| ITEM_ITEMID | AGILE_01 | 0078BE703EC711DDBF9CA7AA7FE3BDFG | C0004::MICHIGAN |
| ITEM_ITEMID | COMMON | 0078BE703EC711DDBF9CA7AA7FE3BDFG | 353138353737393537 32383638303440 |
| ITEM_ITEMID | SAP_01 | 0078BE703EC711DDBF9CA7AA7FE3BDFG | 66252::207 |

- The first row entry is made by the Agile PLM BPEL flow for part P0001. This part belongs to an Agile site that is mapped to site DETROIT.
- The second entry (Common) is also created by the Agile BPEL flow. It indicates the common "business component id" for this particular in integration entity and is used to link Agile Parts/Documents/Change orders to SAP Items/Change Orders.
- The third row entry represents the SAP inventory item id corresponding to the item P1B and the organization id for the item. The rest of the entries represent the revised and component items for the ECO001.
- The ITEM_ITEMID virtual table contains the similar xref entries for each item/part created through the new part request process.

## Creating Cross Reference Data

If data needs to be ported from other existing integrations to the Agile PLM pre-built integration, first the relationship between the Agile entities and the SAP entities should be established and each Agile site should be mapped to corresponding SAP plant by inserting the relevant data in the **xref_data** table.

Even for existing Agile PLM pre-built integrations, if an item or change order needs to be created in SAP and then it is created in Agile, for the pre-built integration to process the item xref entry for that item needs to be made. All the update (reverse) flows for the pre-built integration will update the item attributes in Agile only if there is an entry in the **xref_data** table for that particular item.

Cross Reference file is stored in Metadata database repository in the below path

oramds:/apps/AIAMetaData/xref/<xref-name>.xref where xref–name is ITEM_ITEMID or CHANGE_CHANGE_ID

The data that has to be inserted in the xref_data table should be put in an XML file with the following format.

This is a sample change order data file:

```
<xref xmlns="http://xmlns.oracle.com/xref">
    <table name="CHANGE_CHANGEID">
      <columns>
        <column name="SAP_01"/>
        <column name="COMMON"/>
        <column name="AGILE_01"/>
      </columns>
    <rows>
      <row>
        <cell colName="SAP_01">11075</cell>
        <cell colName="COMMON">2d37383330323730313238383735363l</cell>
        <cell colName="AGILE_01">ECO001::DETROIT</cell>
      </row>
      <row>
        <cell colName="SAP_01">11076</cell>
        <cell colName="COMMON">2d37383330323730313238383735363 2</cell>
        <cell colName="AGILE_01"> ECO001::MICHIGAN</cell>
      </row>
    </rows>
```

```
</table>
</xref>
```

The change id number that is inserted in the SAP_01 column for a particular change order can be obtained using the following query:-

Select change_id from eng_engineering_changes

```
where
change_notice=<AgileChangeOrder>;
```

- The business component id that is inserted in the COMMON column can be any unique number.
- The Agile change order number and the SAP organization corresponding to the Agile site are inserted into the AGILE_01 column separated by "**:**".

This is a sample Item Data file:

```
<xref xmlns="http://xmlns.oracle.com/xref">
  <table name="ITEM_ITEMID">
    <columns>
      <column name="SAP_01"/>
      <column name="COMMON"/>
      <column name="AGILE_01"/>
    </columns>
  <rows>
    <row>
      <cell colName="SAP_01">66247::0001</cell>
      <cell colName="COMMON">353138353737393537323836383303435</cell>
      <cell colName="AGILE_01"> P0001::DETROIT</cell>
    </row>
    <row>
      <cell colName="SAP_01">66248::PAV0</cell>
      <cell colName="COMMON">353138353737393537323836383303436</cell>
      <cell colName="AGILE_01"> P0002::MICHIGAN</cell>
    </row>
       <row>
      <cell colName="SAP_01">66249::0001</cell>
```

```
            <cell colName="COMMON">35313835373739353732383638303437</cell>
            <cell colName="AGILE_01"> C0001::DETROIT</cell>
        </row>
        <row>
            <cell colName="SAP_01">66250::0001</cell>
            <cell colName="COMMON">35313835373739353732383638303438</cell>
            <cell colName="AGILE_01"> C0002::DETROIT</cell>
        </row>
    <row>
            <cell colName="SAP_01">66251::PAV0</cell>
            <cell colName="COMMON">35313835373739353732383638303439</cell>
            <cell colName="AGILE_01"> C0003::MICHIGAN</cell>
        </row>
        <row>
            <cell colName="SAP_01">66252::PAV0</cell>
            <cell colName="COMMON">35313835373739353732383638303440</cell>
            <cell colName="AGILE_01"> C0004::MICHIGAN</cell>
        </row>
      </rows>
    </table>
    </xref>
```

The Item inventory number and the organization id is inserted in the SAP_01 column separated by ":"

The business component id that is inserted in the COMMON column can be any unique number.

- The Agile item number and the SAP plant corresponding to the Agile site are inserted into the AGILE_01 column separated by "::".

# Configuring the Integration

This pre-built integration 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 pre-built integrations running on the SOA suite. Service level parameters can be configured at the individual service levels, such as ABCS.

The configuration properties from Agile Module and SAP 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 pre-built integration 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 above.

### Steps to upload to MDS

a) Update the required property in AIAConfigurationProperties.xml file.

b) Go to <AIA_HOME>\aia_instances\AIA1115\config' path and open the UpdateMetaDataDP.xml file and add <include name="config/AIAConfigurationProperties.xml"/>.

- c) Go to <AIA_HOME>\Infrastructure\Install\config and run the below command.
- ant –f UpdateMetaData.xml

**Note**: Whenever the AIAConfigurationProperties.xml file is updated, the file must be updated in the MDS

## Agile Configurations

| Properties | (default) Value/Setting | Description |
|---|---|---|
| **Module Name** | **Agile** | |
| **MULTISITE_ENABLED** | TRUE | When set to True, the sites specified in Sites Tab of Items are used to determine the Plants in SAP to which they are mapped. |

| Properties | (default) Value/Setting | Description |
|---|---|---|
| | | When set to False, Page2 Multilist01attribute is used to determine the Plants in SAP to which the Item will be extended to. |
| **Item.UnitCostAttribute** | Site.Numeric01 | Determines the attribute to which the unit cost from SAP would be updated in Agile. |
| **Change.TransferStatusAttribute** | PageTwo.Text02 | Determines the attribute to which the transfer status of a Change should be written.<br>When the Change flow is from Agile to SAP, the possible values are "Transferred" or "Errored".<br>When the Change flow is from SAP to Agile, the value would be the same as that of the Status of the Change in all the Plants of SAP. |
| **Service Name** | **CreateQueueService** | |
| TRACE.LOG.ENABLED | false | Use tracelog for the flow |
| **Service Name** | **QueueProcessorServiceImpl** | |
| TRACE.LOG.ENABLED | false | Use tracelog for the flow |
| **Service Name** | **UpdateItemListAgileProvABCSImpl** | |
| 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 |
| Routing.ItemABSService.AGILE_01.EndpointURI | http://${agile.host}:${agile.port}/Agile/integration/services/ItemABS | Route to ItemABS |

| Properties | (default) Value/Setting | Description |
|---|---|---|
| ItemResponseEBS.UpdateItemListResponse.RouteToCAVS | FALSE | Route to CAVS is set to TRUE |
| Routing.ItemResponseEBS.UpdateItemListResponse.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | If Route to CAVS set as true this endpoint URI will trigger |
| **Service Name** | **ProcessEngineeringChangeOrderAgileReqABCSImpl** | |
| 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.SystemID | AGILE_01 | |
| Routing.ChangeABSService.RouteToCAVS | false | |
| Routing.ChangeABSService.AGILE_01.EndpointURI | http://${agile.host}:${agile.port}/Agile/integration/services/ChangeABS | |
| Routing.EngineeringChangeOrderEBS.CreateEngineeringChangeOrderList.RouteToCAVS | false | CAVS property set to false |
| Routing.EngineeringChangeOrderEBS.CreateEngineeringChangeOrderList.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS end point URI |

**Note**: Multisite_Enabled property is governed by Distributed Processing aspects covered in Chapter 2: Process Integration for ECO/PREL. When it is set to TRUE (default), the Item.UnitCostAttribute is set to Site Tab Flex Attributes. You can use Numeric, Text or Money flex fields in the Site tab for these settings and is denoted by the first element, Site. For example, if Site.Numeric01 is set to FALSE, all these attributes are set to Page2 or Page3 flex fields. Hence, the settings 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 Application Interfaces.

## SAP Configurations

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| **ModuleName** | SAP | |
| LANG_LOCALE | en-US | Used to specify the language code and locale of the SAP system |
| **GLOBAL_LEVEL_BOM** | FALSE | When this property is set to FALSE, in SAP the BOM are created at plant level and not at the global level. If set to TRUE, the BOM will be created at the global level. |
| **DEFAULT_SYSTEM_ID** | SAP_01 | Used to get the default XREF Target column name when TargetId is empty in incoming EBM. It should be kept SAP_01 as default for this release |
| **MATERIAL_TYPE** | FERT | Used to specify the material type in SAP to be “Finished Product” |
| **SAP46_N_ABOVE** | TRUE | Specifies that the version of SAP used is greater than 4.6. |
| **BOM_USAGE** | 1 | BOM_USAGE denotes the usage of the BOM. 1indicates production level BOM. This value is dependent on the business process implemented at the client site. Some other values are:<br>2 indicates engineering/design BOM<br>3 indicates universal BOM<br>4 indicates plant maintenance BOM |
| **ITEM_CATEGORY** | L | ITEM_CATEGORY denotes the category of the item that is added to the BOM. L indicates stock item. This value is dependent on the business process implemented at the client site. Some other values |

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| | | are:<br>N indicates non-stock item<br>T denotes text item<br>D denotes document item |
| **ServiceName** | **ReserveItemSAPProvABCS** | |
| 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.POSTPROCESSITEMEBM | false | User exit for the post-process Item EBM should be called or not |
| ABCSEXTENSION.PREPROCESSITEMEBM | false | User exit for the pre-process Item EBM should be called or not |
| Default.SystemID | SAP_01 | Used to get the default XREF Target column name when TargetId is empty in incoming EBM. It should be kept SAP_01 as default for this release |
| Routing.ItemResponseEBS.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.ItemResponseEBS.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/asyncresponserecipient | CAVS Endpoint URI |
| Routing.GenerateItemNumberService_Reserve.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.GenerateItemNumberService_Reserve.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/ReserveItemService/ReserveItemService_ep | ReserveItemSAP runtime target endpoint URI |
| Routing.GenerateItemNumberService_Reserve.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet | CAVS Enpoint URI |

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| | /syncresponsesimulator | |
| TRACE.LOG.ENABLED | false | Use tracelog for the flow |
| **Service Name** | **CreateEngineeringChangeOrderListEBF** | |
| 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.POSTPROCESSITEMEBM | false | User exit for the post-process Item EBM should be called or not |
| ABCSEXTENSION.PREPROCESSITEMEBM | false | User exit for the pre-process Item EBM should be called or not |
| Default.SystemID | SAP_01 | Used to get the default XREF Target column name when TargetId is empty in incoming EBM. It should be kept SAP_01 as default for this release |
| Routing.ItemEBSV2.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.ItemEBSV2.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/ItemEBSV2/ItemEBSV2_ep | ItemEBS runtime target endpoint URI |
| Routing.ItemEBSV2.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| Routing.EngineeringChangeOrderResponseEBS.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.EngineeringChangeOrderResponseEBS.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/Engineeri | EngineeringChangeOrderEBS runtime target endpoint URI |

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| | ngChangeOrderResponseEBS!1.0/EngineeringChangeOrderResponseEBS_ep | |
| Routing.EngineeringChangeOrderResponseEBS.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/asyncresponserecipient A | CAVS Endpoint URI |
| TRACE.LOG.ENABLED | false | Use tracelog for the flow |
| **Service Name** | **CreateEngineeringChangeOrderListSAPProvABCS** | |
| ABCSEXTENSION.PREPROCESSABM | false | User exit for the pre-process ABM should be called or not |
| ABCSEXTENSION.POSTPROCESSABM | false | User exit for the post-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 |
| Default.SystemID | SAP_01 | Used to get the default XREF Target column name when TargetId is empty in incoming EBM. It should be kept SAP_01 as default for this release |
| Routing.CreateEngineeringChangeOrderService.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.CreateEngineeringChangeOrderService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-Infra/services/default/ECOService/ECORouterService_ep | CreateEngineeringChangeOrderService runtime target endpoint URI |
| Routing.CreateEngineeringChangeOrderService.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| TRACE.LOG.ENABLED | false | Use tracelog for the flow |
| **Service Name** | **SyncItemListSAPProvABCS** | |
| ABCSEXTENSION.PREPROCESSREVLEVELABM | false | User exit for the pre-process ABM should be called or not |

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| ABCSEXTENSION.POSTPROCESSREVLEVELABM | false | User exit for the post-process ABM should be called or not |
| ABCSEXTENSION.PREPROCESSSAVEDDATAABM | false | User exit for the pre-process SAVED DATA ABM should be called or not |
| ABCSEXTENSION.POSTPROCESSSAVEDDATAABM | false | User exit for the post-process SAVED DATA ABM should be called or not |
| ABCSEXTENSION.PREPROCESSREFMATERIALABM | false | User exit for the pre-process REF MATERIAL ABM should be called or not |
| ABCSEXTENSION.POSTPROCESSREFMATERIALABM | false | User exit for the post-process REF MATERIAL 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 |
| Default.SystemID | SAP_01 | Used to get the default XREF Target column name when TargetId is empty in incoming EBM. It should be kept SAP_01 as default for this release |
| Routing.Call_ItemListResponseEBS.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.Call_ItemListResponseEBS.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/ asyncresponserecipient | CAVS Endpoint URI |
| Routing.CreateReferencematerial.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.CreateReferencematerial.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/ItemServiceESB/ItemServiceESB_ep | CreateReferenceMaterial runtime target endpoint URI |
| Routing.CreateReferencematerial.CAVS.EndpointURI | http:// ${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/sync | CAVS Endpoint URI |

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| | responsesimulator | |
| Routing.CreateDetailReferencematerial.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.CreateDetailReferencematerial.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/ItemServiceESB/ItemServiceESB_ep | CreateDetailReferenceMaterial runtime target endpoint URI |
| Routing.CreateDetailReferencematerial.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| Routing.CreateItemService.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.CreateItemService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-Infra/services/default/ItemServiceESB/ItemServiceESB_ep | CreateItemService runtime target endpoint URI |
| Routing.CreateItemService.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| Routing.CreateItemRevisionService.RouteToCAVS | False | Route to CAVS, if set as True |
| Routing.CreateItemRevisionService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}//soa-Infra/services/default/ItemServiceESB/ItemServiceESB_ep | CreateItemRevisiopnService runtime target endpoint URI |
| Routing.CreateItemRevisionService.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| TRACE.LOG.ENABLED | False | Use tracelog for the flow |
| **ServiceName** | **ProcessBillOfMaterialsListSAPProvABCS** | |

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| ABCSEXTENSION.PREPROCESSABM | false | User exit for the pre-process ABM should be called or not |
| ABCSEXTENSION.POSTPROCESSABM | false | User exit for the post-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 |
| Default.SystemID | SAP_01 | Used to get the default XREF Target column name when TargetId is empty in incoming EBM. It should be kept SAP_01 as default for this release |
| Routing.GlobalBillOfMaterialExistenceCheckService.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.GlobalBillOfMaterialExistenceCheckService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/BOMServiceESB/BOMServiceESB_ep | Global BillOfMaterialExistenceCheckService runtime target endpoint URI |
| Routing.GlobalBillOfMaterialExistenceCheckService.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulatorr | CAVS Endpoint URI |
| Routing.BillOfMaterialExistenceCheckService.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.BillOfMaterialExistenceCheckService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/BOMServiceESB/BOMServiceESB_ep | BillOfMaterialExistenceCheckService runtime target endpoint URI |
| Routing.BillOfMaterialExistenceCheckService.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| TRACE.LOG.ENABLED | False | Use tracelog for the flow |
| **ServiceName** | **CreateBillOfMaterialsListSAPProvABCSImpl** | |

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| ABCSEXTENSION.PREPROCESSBOMEXISTENCECHECKABM | false | User exit for the pre-process BOM Existence Check ABM should be called or not |
| ABCSEXTENSION.POSTPROCESSBOMEXISTENCECHECKABM | false | User exit for the post-process BOM Existence Check ABM should be called or not |
| ABCSEXTENSION.PREPROCESSBOMPLANTEXTENSIONABM | false | User exit for the pre-process BOM Plant Existence Check ABM should be called or not |
| ABCSEXTENSION.POSTPROCESSBOMPLANTEXTENSIONABM | false | User exit for the post-process BOM Plant Existence Check ABM should be called or not |
| ABCSEXTENSION.PREPROCESSBOMCREATEABM | false | User exit for the pre-process BOM Create ABM should be called or not |
| ABCSEXTENSION.POSTPROCESSBOMCREATEABM | false | User exit for the post-process BOM Create 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 |
| Default.SystemID | SAP_01 | Used to get the default XREF Target column name when TargetId is empty in incoming EBM. It should be kept SAP_01 as default for this release |
| Routing.BillOfMaterialsResponseEBS.RouteToCAVS | False | Route to CAVS, if set as True |
| Routing.BillOfMaterialsResponseEBS.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/ asyncresponserecipient | CAVS Endpoint URI |
| Routing.CreateGlobalBillOfMaterialService.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.CreateGlobalBillOfMaterialService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/BOMServiceESB/BOMServiceESB_ep | CreateBillOfMaterialService runtime target endpoint URI |

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| Routing.CreateGlobalBillOfMaterialService.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| Routing.CreateBillOfMaterialService.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.CreateBillOfMaterialService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/BOMServiceESB/BOMServiceESB_ep | CreateBillOfMaterialService runtime target endpoint URI |
| Routing.CreateBillOfMaterialService.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| Routing.BOMPlantExtensionService.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.BOMPlantExtensionService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/BOMServiceESB/BOMServiceESB_ep | BOMPlantExtensionService runtime target endpoint URI |
| Routing.BOMPlantExtensionService.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| Routing.BillOfMaterialExistenceCheckService.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.BillOfMaterialExistenceCheckService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/BOMServiceESB/BOMServiceESB_ep | BillOfMaterialExistenceCheckService runtime target endpoint URI |
| Routing.BillOfMaterialExistenceCheckService.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| TRACE.LOG.ENABLED | false | Use tracelog for the flow |
| **ServiceName** | **UpdateBillOfMaterialsListSAPProvABCSImpl** | |

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| ABCSEXTENSION.PREPROCESSBOMEXISTENCECHECKABM | false | User exit for the pre-process BOM Existence Check ABM should be called or not |
| ABCSEXTENSION.POSTPROCESSBOMEXISTENCECHECKABM | false | User exit for the post-process BOM Existence Check ABM should be called or not |
| ABCSEXTENSION.PREPROCESSBOMPLANTEXTENSIONABM | false | User exit for the pre-process BOM Plant Existence Check ABM should be called or not |
| ABCSEXTENSION.POSTPROCESSBOMPLANTEXTENSIONABM | false | User exit for the post-process BOM Plant Existence Check ABM should be called or not |
| ABCSEXTENSION.PREPROCESSBOMUPDATEABM | false | User exit for the pre-process BOM Update ABM should be called or not |
| ABCSEXTENSION.POSTPROCESSBOMUPDATEABM | false | User exit for the post-process BOM Update 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 |
| Default.SystemID | SAP_01 | Used to get the default XREF Target column name when TargetId is empty in incoming EBM. It should be kept SAP_01 as default for this release |
| Routing.BillOfMaterialsResponseEBS.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.BillOfMaterialsResponseEBS.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/ asyncresponserecipient | CAVS Endpoint URI |
| Routing.UpdateGlobalBillOfMaterialService.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.UpdateGlobalBillOfMaterialService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/BOMServ | UpdateBillOfMaterialService runtime target endpoint URI |

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| | iceESB/BOMServiceESB_ep | |
| Routing.UpdateGlobalBillOfMaterialService.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| Routing.BillOfMaterialExistenceCheckService.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.BillOfMaterialExistenceCheckService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/BOMServiceESB/BOMServiceESB_ep | BillOfMaterialExistenceCheckService runtime target endpoint URI |
| Routing.BillOfMaterialExistenceCheckService.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| Routing.BOMPlantExtensionService.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.BOMPlantExtensionService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/BOMServiceESB/BOMServiceESB_ep | BOMPlantExtensionService runtime target endpoint URI |
| Routing.BOMPlantExtensionService.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| Routing.UpdateBillOfMaterialService.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.UpdateBillOfMaterialService.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/BOMServiceESB/BOMServiceESB_ep | UpdateBillOfMaterialService runtime target endpoint URI |
| Routing.UpdateBillOfMaterialService.CAVS.EndpointURI | http://${fp.server.soaserverhostname | CAVS Endpoint URI |

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| | }:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | |
| TRACE.LOG.ENABLED | false | Use trace log for the flow |
| **ServiceName** | **UpdateItemListSAPReqABCSImpl** | |
| ABCSEXTENSION.PREPROCESSABM | false | User exit for the pre-process ABM should be called or not |
| ABCSEXTENSION.PREPROCESSEBM | false | User exit for the pre-process EBM should be called or not |
| ABCSEXTENSION.POSTPROCESSABM | false | User exit for the post-process ABM should be called or not |
| ABCSEXTENSION.POSTPROCESSEBM | false | User exit for the post-process EBM should be called or not |
| Default.SystemID | SAP_01 | Used to get the default XREF Target column name when TargetId is empty in incoming EBM. It should be kept SAP_01 as default for this release |
| Routing.ItemEBSV2.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.ItemEBSV2.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/ItemEBSV2/ItemEBSV2_ep | ItemEBS runtime target endpoint URI |
| Routing.ItemEBSV2.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| **ServiceName** | **UpdateItemListSAPReqABCS** | |
| ABCSEXTENSION.POSTPROCESSABM | false | User exit for the post-process ABM should be called or not |
| ABCSEXTENSION.PREPROCESSABM | false | User exit for the pre-process ABM should be called or not |
| Default.SystemID | SAP_01 | |
| TRACE.LOG.ENABLED | false | |
| ServiceName | ProcessBillOfMaterialsListSAPProvABCSImpl | |

| Properties | (Default) Value/Setting | Description |
|---|---|---|
| ABCSEXTENSION.PREPROCESSABM | false | User exit for the pre-process ABM should be called or not |
| ABCSEXTENSION.POSTPROCESSABM | false | User exit for the post-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 |
| Default.SystemID | SAP_01 | Used to get the default XREF Target column name when TargetId is empty in incoming EBM. It should be kept SAP_01 as default for this release |
| Routing.Retreive_BOMData.RouteToCAVS | false | Route to CAVS, if set as True |
| Routing.Retreive_BOMData.SAP_01.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/soa-infra/services/default/BOMServiceESB/BOMServiceESB_ep | ProcessBillOfMaterialsListImpl runtime target endpoint URI |
| Routing.Retreive_BOMData.CAVS.EndpointURI | http://${fp.server.soaserverhostname}:${fp.server.soaserverport}/AIAValidationSystemServlet/syncresponsesimulator | CAVS Endpoint URI |
| TRACE.LOG.ENABLED | false | Use tracelog for the flow |

## Setting up National Language Support

Here are the steps to set up National Language Support (NSL):

- Setting up NLS in Agile
- Setting up NLS in FMW for Agile
- Editing AIAConfigurations
- Setting Up DVMs
- Setting up the DVMs for NLS

## Setting up NLS in Agile:

To set up NLS in Agile:

1. In Agile Java Client, for each subscriber of MCO, SCO, ECO, go to the subscriber details page and set the language.

**Note**: For ease of understanding, we are using "Japanese" as language.

2. Change the language preference of all the users creating CO and Items, including integration user, to Japanese.

## Setting up NLS in FMW for Agile

To set up NLS in FMW for Agile:

1. Edit the following XSL file and replace the hard-coded string 'Preliminary' to Japanese equivalent coming from Agile List values

   \AIA_HOME\services\core\Agile\UtilityServices\QueueProcessorServiceImpl\AgileData_to_AgileCreateEngineeringChangeOrderListABM_Impl.xsl

2. Edit the following XSL file and replace the following values:

   \AIA_HOME\services\core\Agile\RequesterABCS\ProcessEngineeringChangeOrderAgileReqABCSImpl\AgileCreateEngineeringChangeOrderListABM_to_CreateEngineeringChangeOrderListEBM_Impl.xsl

   Replace the hard-coded string 'Preliminary' to the Japanese equivalent coming from Agile list values.

   Replace the hard-coded string 'SCO' to Japanese equivalent coming from Agile list values.

3. Edit the following XSL file and replace hard coded strings ‘Errored’ with Japanese equivalent

   \AIA_HOME\services\core\Agile\RequesterABCS\ProcessEngineeringChangeOrderAgileReqABCSImpl\AgileCreateEngineeringChangeOrderListABM_to_AgileUpdateEngineeringChangeOrderListABM_Impl.xsl

4. Edit the following XSL file and replace the following values:

   \AIA_HOME\services\core\Agile\RequesterABCS\ProcessEngineeringChangeOrderAgileReqABCSImpl\UpdateEngineeringChangeOrderListEBM_to_AgileUpdateEngineeringChangeOrderListABM_Impl.xsl

   Replace the hard coded string ‘Transferred’ with its Japanese equivalent.

   Replace the hard coded string ‘Errored’ with its Japanese equivalent..

**Note**: Use an UTF-8 based editor like JEdit to carry out these changes. Choose UTF-8 as the character set while loading the file to edit. Set FTP transfer mode to Binary while uploading these files to AIA components.

### Editing AIAConfigurations

Edit the property 'LANG_LOCALE' under module 'Agile' to 'Japanese'. The language value is available under column COMMON in the DVM - 'LANGUAGE_CODE'.

### Setting Up DVMs

To set up DVMs, enter the Japanese language values under Agile column, AGILE_01 in the following DVMs:

- ITEM_PRIMARYCLASSIFICATIONCODE
- ITEM_STATUS_CODE
- ITEM_UOM_CODE
- ECO_STATUS_CODE

After all the xsls are modified, the corresponding process flows need to be redeployed for the xsl changes to take effect.

### Setting up the DVMs for NLS

To set up DVMs for NLS, configure SAP_01 with the appropriate language code of SAP in the *LANGUAGE_CODE* DVM column.

## Domain Value Maps

DVMs are a standard feature of the Oracle SOA Suite. They enable you to equate lookup codes and other static values across applications, for example, FOOT and FT or US and USA.DVMs are static in nature, though administrators can add additional maps as needed. Transactional business processes never update DVMs; they only read from them. DVMs are stored in XML files and cached in memory at run time.

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

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

## Out-of-the-Box DVMs

During installation, the DVMs used for the pre-built integration 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 attributes being mapped are list values, the Agile data is not seeded. Hence, it should be changed accordingly and then updated in the MDS.

Some of the most used DVMs, with their Out-of-the-Box values, are listed below. You can modify their values as per your requirements. For complete list of available DVMs, including the most used ones, see the List of DVMs.

**Note**: You can only add more rows of value mappings but should not change the DVM name, the column names, or the number of columns.

## Sample DVMs

These are the sample domain value mapping (DVM):

### ECO_ENGINEERINGCHANGEORDERLINE_STATUS_CODE

Used for SAP attribute - ECO_ATTR/ECO_REVISED_ITEM_TYPE/ECO_REVISED_ITEM_TYPE_ITEM/STATUS_NAME

| SAP_01 | COMMON |
|---|---|
| Release | RELEASE |
| Open1 | OPEN_1 |
| EBS_Release | EBS_RELEASE |
| EBS Open | EBS_OPEN |
| Implemented | IMPLEMENTED |
| Create | CREATE |
| In Progress | IN_PROGRESS |
| Scheduled1 | SCHEDULED_1 |
| Cancelled | CANCELLED |

| SAP_01 | COMMON |
|---|---|
| Approval | APPROVAL |
| Pending Response | PENDING_RESPONSE |
| EBS_Create | EBS_CREATE |
| EBS_Review | EBS_REVIEW |
| EBS_Archive | EBS_ARCHIVE |
| Completed | COMPLETED |
| Hold | HOLD |
| Draft | DRAFT |
| Review | REVIEW |
| Released | RELEASED |
| Archive | ARCHIVE |

### ECO_STATUS_CODE

Used for SAP attribute - ECO_CHANGE_ORDER_TYPE/STATUS_NAME, ECO_REVISED_ITEM_TYPE/STATUS_NAME. Use for Agile's Status attribute of a Change.

| SAP_01 | COMMON | AGILE_01 |
|---|---|---|
| Release | RELEASE | |
| EBS_Approval | EBS_APPROVAL | |
| Open1 | OPEN_1 | |
| EBS_Release | EBS_RELEASE | |
| EBS Open | EBS_OPEN | |
| Implemented | IMPLEMENTED | Implemented |
| Create | CREATE | |
| In Progress | IN_PROGRESS | |
| Scheduled | SCHEDULED | Released |
| Cancelled | CANCELLED | |
| Approval | APPROVAL | |
| Pending Response | PENDING_RESPONSE | |
| EBS_Create | EBS_CREATE | |
| EBS_Review | EBS_REVIEW | |

| SAP_01 | COMMON | AGILE_01 |
|---|---|---|
| EBS_Archive | EBS_ARCHIVE | |
| Completed | COMPLETED | |
| Hold | HOLD | |
| Draft | DRAFT | |
| Review | REVIEW | |
| Released | RELEASED | |
| Archive | ARCHIVE | |

### ITEM_STATUS_CODE

This value is used for SAP Item attribute ITEM_OBJ/MAIN_OBJ_TYPE/INVENTORY_ITEM_STATUS_CODE. The Agile Item Lifecycle phase attribute is mapped.

| SAP_01 | COMMON | AGILE_01 |
|---|---|---|
| A | A | |
| Concept | CONCEPT | |
| Design | DESIGN | Preliminary |
| Engineer | ENGINEER | Pilot |
| Inactive | INACTIVE | Inactive, 02 |
| Lease | LEASE | |
| Non-Stock | NONSTOCK | |
| Nwe B | NWEB | |
| OPM | OPM | |
| Obsolete | OBSOLETE | Obsolete, 02 |
| Pending | PENDING | |
| Phase-Out | PHASEOUT | |
| Production | PRODUCTION | Production, 03 |
| Prototype | PROTOTYPE | Prototype |
| R&D | R&D | |

### AGILE_SITE_TARGET_MAPPING

DEFAULT_MASTER_ORG in SAP is specified here. This is used when the Multisite Enabled property is set to False and no Plant is specified for the Item where it extends to.

When the Multisite_Enabled property is true, the Sites in Agile are mapped to various Plants in SAP. A Site could be mapped to multiple Plants in the SAP column with "|" delimiter.

### AGILE_TARGET_SITE_MAPPING

The SAP Plants to Agile Sites are mapped. This is used for SAP to Agile flows. There is one to one mapping between the SAP Plant to Agile Site.

## List of DVMs

This table describes the domain value mappings and their description:

**Note**: Important DVMs are listed in bold typeface.

| Domain Value Map | Description |
|---|---|
| ITEM_UOM_CODE: | This DVM is used to maintain the Unit of Measure values of Agile corresponding to SAP |
| **REFERENCEMATERIALS:** | We are maintaining this DVM to populate the mandatory parameters of SAP BasicData1, Basic data2, MRP1, MRP2, MRP3 views which are not being sent from Agile |
| LANGUAGE_CODE: | This DVM is used to populate the Multi languages, as of now we are dealing with only English language |
| **ITEM_STATUS_CODE:** | This DVM is used to maintain the Life Cycle Phase codes and Special Plant Material Status of SAP |
| **ITEM_PRIMARYCLASSIFICATIONCODE:** | This DVM is used to populate the Material type of SAP |
| ECO_STATUS_CODE: | The DVM is used to handle the Status Codes of Change Order in SAP. |
| AGILE_SITE_TARGET_MAPPING | This DVM is used to map SAP Plants to Agile Sites. This is used for Agile to SAP flows |
| AGILE_TARGET_SITE_MAPPING | This DVM is used to map SAP Plants to Agile Sites. This is used for SAP to Agile flows |

| Domain Value Map | Description |
|---|---|
| ECO_ENGINEERINGCHANGEORDERLINE_REVISEDBILLOFMATERIALS_BILLOFMATERIALSCOMPONENTITEM_CHANGETYPECODE: | We are using this DVM to recognize the Creation or updation of Bill of Materials in Agile |

# Application Interfaces

Application Interfaces are the Web Services and APIs that communicate and transact between Application and Integration Layers.

## Agile PLM Interfaces

These are the Agile PLM interfaces used in this integration:

### WSDLs

| | |
|---|---|
| ItemABS.wsdl | Used to create an item in SAP |
| ItemABS.wsdl | Used to update an Item in SAP |
| ChangeABS.wsdl | Used to create an EngineeringChangeOrder in SAP |

### XSDs

| | |
|---|---|
| ItemABM.xsd | Contains the Item Request and Response ABM |
| ItemABM.xsd | Contains the Update Item List Request and Response ABM |
| EngineeringChangeOrderABM.xsd | Contains CreateEngineeringChangeOrderListABM and CreateEngineeringChangeOrderListResponse ABM |

## SAP Interfaces

These are the SAP interfaces used in this integration:

## WSDLs

| | |
|---|---|
| ReserveItemservice.wsdl | The interface to the SAP Process Item API |
| GenerateItemNumberService_Reserve.wsdl | BAPI_STDMATERIAL_GETINTNUMBER |
| ItemServiceESB.wsdl | The interface to the SAP Process Item API - |
| CreateReferencematerial.wsdl | BAPI_MATERIAL_GET_ALL |
| CreateItemService.wsdl | BAPI_MATERIAL_SAVEDATA |
| CreateDetailReferencematerial.wsdl | BAPI_MATERIAL_GET_DETAIL |
| CreateItemRevisionService.wsdl | CCAP_REV_LEVEL_MAINTAIN |
| BOMServiceESB.wsdl | The interface to the SAP Process BOM API |
| BOMPlantExtensionService.wsdl | CSAP_MAT_BOM_ALLOC_CREATE |
| CreateBillOfMaterialService.wsdl | CSAP_MAT_BOM_CREATE |
| BillOfMaterialExistenceCheckService.wsdl | BAPI_MAT_BOM_EXISTENCE_CHECK |
| BOMServiceESB.wsdl | The interface to the SAP Process BOM API - |
| BOMPlantExtensionService.wsdl | CSAP_MAT_BOM_ALLOC_CREATE |
| UpdateBillOfMaterialService.wsdl | CSAP_MAT_BOM_MAINTAIN |
| BillOfMaterialExistenceCheckService.wsdl | BAPI_MAT_BOM_EXISTENCE_CHECK |
| BOMServiceESB.wsdl | The interface to the SAP Process BOM API - |
| BillOfMaterialExistenceCheckService.wsdl | BAPI_MAT_BOM_EXISTENCE_CHECK |
| ECORouterService.wsdl | The interface to the SAP Process ECO API - |
| CreateEngineeringChangeOrderService.wsdl | CCAP_ECN_MAINTAIN |
| MATMAS01_receive.wsdl | The interface to the SAP Process Item API - MATMAS01 |
| ProcessBOMService.wsdl | The interface to the SAP Process BOM API |
| ProcessBillOfMaterialsListImpl.wsdl | CSAP_MAT_BOM_READ |

## XSDs

| | |
|---|---|
| BAPI_MAT_BOM_EXISTENCE_CHECK_request.xsd | Contains the BillOfMaterialExistenceCheck Request ABM |
| BAPI_MAT_BOM_EXISTENCE_CHECK_response.xsd | Contains the BillOfMaterialExistenceCheck Response ABM |
| CCAP_ECN_MAINTAIN_request.xsd | Contains the CreateEngineeringChangeOrderList Request ABM |
| CCAP_ECN_MAINTAIN_response.xsd | Contains the |

| | |
|---|---|
| | CreateEngineeringChangeOrderList Response ABM |
| MATMAS01_event.xsd | Contains the UpdateItemList ABM |
| BAPI_STDMATERIAL_GETINTNUMBER_request.xsd | Contains the ReserveItem Request ABM |
| BAPI_STDMATERIAL_GETINTNUMBER_response.xsd | Contains the ReserveItem Response ABM |
| BAPI_MATERIAL_GET_ALL_request.xsd | Contains the CreateReferencematerial Request ABM |
| BAPI_MATERIAL_GET_ALL_response.xsd | Contains the CreateReferencematerial Response ABM |
| BAPI_MATERIAL_GET_DETAIL_request.xsd | Contains the CreateDetailReferencematerial Request ABM |
| BAPI_MATERIAL_GET_DETAIL_response.xsd | Contains the CreateDetailReferencematerial Response ABM |
| BAPI_MATERIAL_SAVEDATA_request.xsd | Contains the CreateItem Request ABM |
| BAPI_MATERIAL_SAVEDATA_response.xsd | Contains the CreateItem Response ABM |
| CCAP_REV_LEVEL_MAINTAIN_request.xsd | Contains the CreateItemRevision Request ABM |
| CCAP_REV_LEVEL_MAINTAIN_response.xsd | Contains the CreateItemRevision Response ABM |
| BAPI_MAT_BOM_EXISTENCE_CHECK_request.xsd | Contains the BillOfMaterialExistenceCheck Request ABM |
| BAPI_MAT_BOM_EXISTENCE_CHECK_response.xsd | Contains the BillOfMaterialExistenceCheck Response ABM |
| CSAP_MAT_BOM_ALLOC_CREATE_request.xsd | Contains the BOMPlantExtension Request ABM |
| CSAP_MAT_BOM_ALLOC_CREATE_response.xsd | Contains the BOMPlantExtension Response ABM |
| CSAP_MAT_BOM_CREATE_request.xsd | Contains the CreateBillOfMaterial Request ABM |
| CSAP_MAT_BOM_CREATE_response.xsd | Contains the CreateBillOfMaterial Response ABM |
| BAPI_MAT_BOM_EXISTENCE_CHECK_request.xsd | Contains the BillOfMaterialExistenceCheck Request ABM |
| BAPI_MAT_BOM_EXISTENCE_CHECK_response.xsd | Contains the BillOfMaterialExistenceCheck Response ABM |
| CSAP_MAT_BOM_ALLOC_CREATE_request.xsd | Contains the BOMPlantExtension Request ABM |
| CSAP_MAT_BOM_ALLOC_CREATE_respons | Contains the BOMPlantExtension Response |

| | |
|---|---|
| e.xsd | ABM |
| CSAP_MAT_BOM_MAINTAIN_request.xsd | Contains the UpdateBillOfMaterial Request ABM |
| CSAP_MAT_BOM_MAINTAIN_response.xsd | Contains the UpdateBillOfMaterial Response ABM |
| CSAP_MAT_BOM_READ_request.xsd | Contains the Read BOM Request ABM |
| CSAP_MAT_BOM_READ_response.xsd | Contains the Read BOM Response ABM |

# Handling Errors

Based on the roles defined for the services, email notifications are sent if an error occurs. The roles below can be assigned at various levels in a hierarchy (service, process, domain) so that when a service errors out -- the Error Handling Framework will use the role value to derive the user(s) that need to be notified of the error. The Error Handling Framework then notifies the user(s) via their preferred notification method, puts the error in the user's Oracle Worklist, and puts the error in the error log.

**Role**: Actor role associated with the error notification. Actor roles receive notifications for and are assigned to error scenarios occurring in Oracle AIA integration flows. An example of an Actor role is a task editable in the Error Console and is meant to be worked on, and resolved by the actor assigned to the task.

**FYI Role**: FYI role associated with the error notification. This role receives for-your-information (FYI) notifications for error scenarios occurring in Oracle AIA integration flows. An example of an FYI role is a customer service representative. The task is displayed in read-only view in the Error Console.

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

# Chapter 6: Customizations

This chapter discusses:

- Customizing the Transformations
- User Exits

The Integration provides two approaches of customizations:

1. **Customizing the Transformations**: This approach allows to modify the mappings between the attributes of the participating applications.
2. **User Exits**: These pre-defined extension points are provided in the OOTB BPEL flow. You can plug-in your own logic at these exit points to validate, enrich and transform data.

## Customizing the Transformations

There may be a need for out-of-the-box and user defined attributes mapping between applications, which are not covered as part of standard transformations. Considering this, the transformation files (XSLs) have been externalized to facilitate the implementers to carry out the following:

- Modify the out-of-the-box transformations:

  Add new mappings for the Agile attributes to EBM attributes. These Agile attributes could either be the ones that have not been mapped out-of-the-box including any of the flex-fields.

  The transformations support the following requirements:

- In the integration flow, there are multiple transformations involving multiple ABO/Ms and EBO/Ms.

  The transformations support the flex field mappings

  In this mechanism of transformations, customer can provide XSLs for complex transformations that are not part of the standard transformations. It is required to support the transformations for user-defined EBO extensions (ex. Custom tags) and in places where customer want to override the standard transformation logic.

### Transformation Rules

These are the transformation rules:

- For a transformation from ABM into EBM, all the flex-fields (interchangeably user defined

attribute / Flex attribute) go under Specification Group element under the main EBM element with a matching type like ValueText (for text values), ValueNumeric (for numeric values) etc carrying the values.

- For the transformations from EBM into ABM depending upon the identification element of Specification and Specification Group pull processing is done to populate the UDAs.
- For Classification elements, like Item Classification etc, the field values are translated using a configuration, into EBM and vice-versa. The configuration states are <<Name in Agile>> <<Name on EBO>> <<Name SAP ERP>>. For rest of the classification elements like part type, product family etc fields need to be packed/ unpacked on/from the XXCatalog element on the EBM.

# Customization in Agile

The XSL transformations in Agile PLM Integration are externalized, i.e., these are hosted on implementation server under:

<AIA_Home>/AIAComponents/Transformations/Agile/

Refer Customization Points of each Process (in corresponding chapters).

The implementers modify the XSLs based on the customization points against each process to modify, include new mappings for the Agile attributes to the EBM elements. The entire behavior of the OOTB mappings can be enhanced using this approach.

**Note**: Server has to be restarted to bring the changes into effect.

## Sample Customization

A few sample customizations have been provided at <AIA_HOME>\PIPS\Core\Agile\Samples. A sample customization of ECO attributes is given below:

```
User-Defined ECO - in Sites Tab Under Items
```

**Mapping:**

| Agile | EBM | SAP |
|---|---|---|
| Item.Site:List01 | corecomEBO:RevisedItem/corecomEBO:BaseUOMCode | Primary UOM Code |

**Template:**

- createEngineeringChangeOrderLines_With_SiteData

  createEngineeringChangeOrderLines_With_OrgData

  createEngineeringChangeOrderLines_With_DefaultMasterOrgData

# Customization in SAP

To customize a Mapping File:

1. Pickup the <flow>_Custom XSL file from either the Requester/Provider/EBS service where <flow> is the process name, such as ECO_CHANGE_ORDER_TYPE.

   **a.** If only flex-field transformation has to be modified then the custom targets can be modified to include the new mappings.

   For example, ECO_CHANGE_ORDER_TYPE_Custom in Create Eco flow for the change order flex attributes

   **b.** The user need not set the custom transformations property in this case as the targets are already included in the base transformation file.

   If the entire mapping has to be modified then the user can copy the base mappings into the custom target (e.g. Custom in Create ECO flow) in the custom file and modify the mappings as desired.

   **c.** After modifying the file the user has to set the Custom Transformations property in the AIA Configurations file and update the configuration file from MDS respository

2. Restart the server after step 2 and 3. The server must be restarted for the new transformations to load into the JVM.

## Templates in the Custom Files

1. **CreateEngineeringChangeOrderListEbmToAbm**: The following templates are used in the custom transformation files that are used to map the flex field attributes.

   **a.** ECO_CHANGE_ORDER_TYPE_Custom

   **b.** ECO_REVISED_ITEM_TYPE_ITEM_Custom

   **c.** REFERENCE_DESIGNATOR_TBL_Custom

   **d.** COMPONENT_ITEM_TBL_Custom

   **e.** SUBSTITUTE_COMPONENT_TBL_ITEM_Custom

   **f.** STRUCTURE_HEADER_Custom

   The **Custom** template is used for modifying the entire mapping.

3. **ItemEbmToAbm**: The template CUSTOM_OBJ_TYPE_Custom is used to map the flex field attributes. The template “Custom” is used to replace the entire mapping.
4. **UpdateItemListABMToEBM**: The template ItemSpecificationGroup_Custom is used the map the flex field to the specification group in the EBM. The “UpdateItemListABMToEBMCustom” template is used to replace the entire file.

**Note**: Since the templates are directly included inside the objects only the specific attributes being mapped should be put in the custom xsl and not the entire hierarchy.

### Replacing the Entire Mapping

Use this template, **UpdateItemListABMToEBMCustom** for complete ABM to EBM transformation.

Change the following property in AIA Configurations file:

1. File Path: $AIA_HOME/config/AIAConfigurationProperties.xml.

   This step is not required if the flex field mapping is done.

**Note**: The use custom transformation property is per integration flow and has to be set accordingly.

2. Reload the AIA Configurations file.
3. Restart the Server.

## User Exits

The Agile to SAP integration provides user exits to allow custom transformations or filtration routines a customer may want to add in the process without affecting the main integration flow. The user exit points for each process are listed in their respective chapters.

These are the user exit points for requester and provider flows:

- Requester
- Provider

### Requester Flows

User exits points for the Requester flow:

- Just prior to the execution of transformation of ABM to EBM
- Just prior to the invocation of Enterprise Business Service

- Just prior to the execution of transformation of EBM to ABM
- Just prior to the invocation of callback service or response return

## Provider Flows

User exits points for the Provider flow:

- Just prior to the execution of transformation of EBM to ABM
- Just prior to the invocation of Application Service
- Just prior to the execution of transformation of ABM to EBM
- Just prior to the invocation of callback EBS or return of response message

## Steps for User Exits

These are the steps for implementing user exits:

1. Identify which out-of-box flow is to be extended.
2. Identify the suitable exit point in the flow.
3. Develop the flow.
4. Configure the out-of-box flow to include the newly developed flow.
5. Test the developed flow.

# Appendix A: National Language Support in Agile – SAP Integration

National Language Support (NLS) has been implemented in the Agile to SAP Integration.

## Requester Flows

Before sending the EBM to Agile, the Language code is converted to the common value, which is configured from the DVM **LANGUAGE_CODE**.

**Note**: Have to make sure that the values are configured for the DVM's LANGUAGE_CODE before the flows are triggered.

In AIA Release 2.3 PIP, a change in ECO status returns the SID and the Change ID as an input to the Requester ABCS. However, from AIA Release 2.4, 2.5 and 3.1 PIP onwards, another field called Language_Code has been added to input parameters in the transformations, which has a default value as en-US.

## Provider Flows

By default, in AIA Release 3.1 PIP, the language code value in Agile is set to en-US in the Language_Code DVM during PIP installation**.** This language code value is sent over to the ABCS provider in SAP without any conversion. However, in AIA Release 2.4 and 2.5 PIP, the language code value, en-US, is manually converted from Agile PLM to a SAP system value in the Language_Code DVM after installing the PIP.

# Appendix B: Functionalities Available

The following functionalities are available across different versions of Agile PLM and SAP combination deployments:

| Features/Agile and SAP Versions | Agile 9.2.2.6 and SAP 4.7 | Agile 9.2.2.6 and SAP ECC 6.0 | Agile 9.2.2.7 and SAP 4.7 | Agile 9.2.2.7 and SAP ECC 6.0 | Agile 9.3 and SAP 4.7 | Agile 9.3 and ECC 6.0 |
|---|---|---|---|---|---|---|
| NPR (Action PX) | N | N | N | N | N | N |
| NPR (Auto Number PX) | Y | Y | Y | Y | Y | Y |
| PREL(ECO Forward Flow<br><br>From Agile to SAP) | Y | Y | Y | Y | Y | Y |
| ECO Update Flow<br><br>(From Agile to SAP) | Y | Y | Y | Y | Y | Y |
| Item Balance Update Flow<br><br>(From SAP to Agile) | N | N | N | N | N | N |
| Item Operational Attribute Update Flow (Cost Attribute Updates only)<br><br>(From SAP to Agile) | Y | Y | Y | Y | Y | Y |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Queue Functionality**<br><br>**(AQ (Database Persistent))** | **Y** | **Y** | **Y** | **Y** | **Y** | **Y** |
| **NPR (Action PX)** | **N** | **N** | **N** | **N** | **N** | **N** |
| **NPR (Auto Number PX)** | **Y** | **Y** | **Y** | **Y** | **Y** | **Y** |
| **PREL(ECO Forward Flow**<br><br>**From Agile to SAP)** | **Y** | **Y** | **Y** | **Y** | **Y** | **Y** |
| **ECO Update Flow**<br><br>**(From Agile to SAP)** | **Y** | **Y** | **Y** | **Y** | **Y** | **Y** |
| | | | | | | |
| **Item Balance Update Flow**<br><br>**(From SAP to Agile)** | **N** | **N** | **N** | **N** | **N** | **N** |
| **Item Operational Attribute Update Flow (Cost Attribute Updates only)**<br><br>**(From SAP to Agile)** | **Y** | **Y** | **Y** | **Y** | **Y** | **Y** |
| **Queue Functionality**<br><br>**(AQ (Database Persistent))** | **Y** | **Y** | **Y** | **Y** | **Y** | **Y** |

# Appendix C: Agile to SAP Entity Maps

This appendix contains information on the following mappings:

- Bill of Materials
- Item EBO
- Item Attribute Update
- Engineering Change Order EBO

## Bill of Materials Mappings

| Agile Entity Attribute | Bill of Materials EBO | SAP Entity: Attribute<br>Group: Attribute | Comments |
|---|---|---|---|
| **Part/Document. Title Block. Number** | \\BillOfMaterialsEBO\ItemReference\Identification\ID | MATNR:Material Number | |
| **Changes.changeNumber** | \\BillOfMaterialsEBO\Identification\ContextID | AENNR:Change Number | |
| **BOM.ITEM_NUMBER** | \\BillOfMaterialsEBO\BillOfMaterialsComponentItem\ ItemReference\ID | IDNRK:BOM Component | |
| **BOM.FIND_NUMBER** | \\BillOfMaterialsEBO\BillOfMaterialsComponentItem\ Identification\ContextID | POSNR:Position Number | |
| **REFDESIG.LABEL** | \\BillOfMaterialsEBO\BillOfMaterialsComponentItem\ProcessingInstruction/Identification/ID | EBORT:Installation point for sub-item | |
| **BOM.QUANTITY** | \\BillOfMaterialsEBO\BillOfMaterialsComponentItem\ Quantity | Menge:Quantity | |

| Agile Entity Attribute | Bill of Materials EBO | SAP Entity: Attribute Group: Attribute | Comments |
|---|---|---|---|
| **BOM.DESCRIPTION** | \\BillOfMaterialsEBO\Identification\ItemReference\ Description | POTX1:BOM item text (line 1) | |
| | \\BillOfMaterialsEBO\BillOfMaterialsComponentItem\ StorageUnitCode | MEINS:Comp Unit Of Measure | |
| | \\BillOfMaterialsEBO\BillOfMaterialsComponentItem\ ItemReference\ClassificationCode | POSTP:Item Category | |
| | \\BillOfMaterialsEBO\BillOfMaterialsComponentItem\ItemReference\ TypeCode | STLAN:BOM Usage | |
| **BOM. Sites. Site Name** | \\BillOfMaterialsEBO\Identification\ContextID | werks:Plant | |

# Item EBO Mappings

| Agile Entity Attribute | Item EBO | BAPI_MATERIAL_SAVEDATA | Comments |
|---|---|---|---|
| Part/Document. Title Block. Number | \\ItemEBO\Name | BAPIMATHEAD: MATNR: Material number: | Associate type Field Description |
| Part/Document. Title Block. Description | \\ItemEBO\Description | BAPI_MAKT: MATL_DESC Material Description: | |
| | \\ItemEBO\TypeCode | BAPIMATHEAD: MATl_TYPE Material Type | |
| Mass | \\ItemEBO\BaseUOM Code | BAPI_MARA: BASE_UOM: Base Unit of Measure | |

| Agile Entity Attribute | Item EBO | BAPI_MATERIAL_SAVEDATA | Comments |
|---|---|---|---|
| | \\ItemEBO\PrimaryClassificationCode | BAPI_MARA: MATL_GROU:P Material group | |
| | \\ItemEBO\TemplateItemReference\ ClassificationCode | BAPI_MARA: MAT_GRP_SM: Material group | |
| | \\ItemEBO\ItemCatalog\Identification\ID | BAPI_MARA: ITEM_CAT: General Item Category Group: | |
| Part/Document. Title Block. Lifecycle Phase | \\ItemEBO\ItemLifecycleCharacteristics\Lifecycle Code | BAPI_MARA: MSTAE: X-Plant matl status | |
| Part/Document. Title Block. Description | \\ItemIdentification\BaseUOMCode\Revison/Label | BAPI_MLTX: TEXT_ID / TEXT_NAME: Basic Data text | |
| | \\itemEBO\ItemPhysicalCharacteristics\ VolumeMeasure | BAPI_MARM: VOLUME: | |
| | | VOLUMEUNIT: Volume Unit of Measure | |
| | \\ItemEBO\ItemPurchasingCharacteristics\ UnitListPrice\Amount | BAPI_MBEW: STD_PRICE: Standard Price | |
| | \\ItemPhysicalCharacterstics\heightmeasure \\ItemPhysicalCharacterstics\lengthmeasure \\ItemPhysicalCharacterstics\widthmeasure | BAPI_MARM: HEIGHT,LENGTH,WIDTH | |
| | | UNIT_DIM Unit of Dimension for height,length,width | |
| | \\ItemOrderManagementCharacteristics\custom/Delivery Plant | BAPI_MVKE: DELYG_PLNT: Delivery Plant | |

| Agile Entity Attribute | Item EBO | BAPI_MATERIAL_SAVEDATA | Comments |
|---|---|---|---|
| Part/Document. Sites. Site Name | \\ItemIdentification\ContextID | BAPI_MARC: PLANT: PLANT | |
| | \\ItemEBO\InventoryLocation\SerialNumberSpecificationCode | BAPI_MARC: SERNO_PROF: Serial Number Profile | |
| | \\ItemEBO\relatedItem\relationshipCode | BAPI_MARA: BASIC_MATL WRKST_NEW | |
| | \\ItemEBO\itemClassification\classificationCode | BAPI_MARA: DIVISION: DIVISION | |
| | \\itemEBO\itemManufacturingCharacteristics\ custom\laborOffice | BAPI_MARA: DSN_OFFICE LABOR | |
| | \\ItemIdentification\Gtin | BAPI_MARM: EAN_UPC: International Article Number | |
| | \\ItemIdentification\EANCode | BAPI_MARM: EAN_CAT: Category of International Article Number | |
| | \\ItemEBO\itemRevisionHistory\identification\id | BAPI_MARA: OLD_MAT_NO: Old material number | |
| | \\ItemEBO\itemPurchasingCharacteristics\custom\purchasingGroup | BAPI_MARC: PUR_GROUP: Purchasing Group | |
| | \\ItemPurchasingCharacteristics/ AssetClassificationCode/Codetype | BAPI_MARC: COMM_CODE: Harmonized code | |
| | \\ItemPlanningCharacteristics\inventoryPlanning Code | BAPI_MARC: AVAILCHECK: Availability check | |

| Agile Entity Attribute | Item EBO | BAPI_MATERIAL_SAVEDATA | Comments |
|---|---|---|---|
| | \\ItemAttachment\attachment\note\status | BAPI_MARA: DOC_CHG_NO: Document change number | |
| Part/Document. Changes. Lifecycle Phase | \\ItemEBO\ItemLifecycleCharacteristics\LifecyclePhaseCode | BAPI_MARA: SAL_STATUS: Cross distribution chain status | |
| | \\Item Order Management Characteristics\Custom\Transportation Group | BAPI_MARA: TRANS_GRP: Transportation group | |
| | \\ItemEBO\InventoryLocation\Identification\ID | BAPI_MARD: STGE_LOC: Storage Location | |
| Part/Document. Sites. Make/Buy | \\ItemPlanningCharacteristics\MakeOrBuycode | | |
| | \\ItemPlanningCharacteristics\MakeOrBuycode | BAPI_MARC: SPPROCTYPE: Special procurement type | |
| | \\itemPlanningCharacteristics\ProcessingleadTimeCharacteristics\ fixedDuration | BAPI_MARC: GR_PR_TIME: Goods receipt processing time in days | |
| | \\ItemPlanningCharacteristics\ProcessingleadTimeCharacteristics\ postProcessingDuration | BAPI_MARC: PLND_DELRY: Planned Delivery Time | |
| | \\ItemEBO\Custom/Product Hierarchy | BAPI_MARA: PROD_HIER: Product Hierarchy | |
| | \\ItemPhysicalCharacteristics\custom/Netweight | BAPI_MARA: NET_WEIGHT: Net weight | |

| Agile Entity Attribute | Item EBO | BAPI_MATERIAL_SAVEDATA | Comments |
|---|---|---|---|
| | | UNIT:OF_WT<br>Unit Of weight:<br>Weight | |
| | \\ItemEBO\ItemPhysicalCharacteristics\WeightMeasure | BAPI_MARAM:<br>GROSS_WT:<br>Gross Weight | |
| | | UNIT_OF_WT<br>UNIT OF WEIGHT:<br>Unit of weight | |
| | \\itemPlanningCharacteristics\Custom\MRPController | BAPI_MARC:<br>MRP_CTRLER:<br>MRP Controller | |
| | \\ItemOrderManagementCharacteristics\Custom\LoadingGroup | BAPI_MARC:<br>LOADINGGRP:<br>Loading Group | |
| | \\ItemOrderManagementCharacteristics\custom\Sales Organization | BAPI_MVKE:<br>SALES_ORG:<br>Sales Organization | |
| | \\ItemEBO\Custom/Product Allocation | BAPI_MARA:<br>PROD_ALLOC:<br>Product Allocation Determination Procedure | |
| | \\ItemEBO\Custom\Authorisation Group | BAPI_MARA:<br>AUTHORITYGROUP<br>Authorization Group | |
| | \\ItemManufacturingCharacteristics\Custom\CountryOfOrigin | BAPI_MARC:<br>COUNTRYORI:<br>Country of Origin | |

# Item Attribute Update Mappings

| Agile Entity Attribute | Item EBO | MATMAS01 | Comments |
|---|---|---|---|
| Part/Document. Title Block. Number | \\ItemEBO\Name | E1MARAM: MATNR<br>Material number: | |
| Part/Document. Title Block. Unit Cost Attribute | \\ItemEBO\ItemPurchasingCharacteristics\UnitListPrice\Amount | E1MBEWM: STPRS Standard Price<br>E1MBEWM: VERPR<br>Moving Average Price | |
| Part/Document. Sites. Site Name | \\ItemIdentification\ContextID | E1MARCM: WERKS<br>Plant | |

# Engineering Change Order EBO Mappings

| Agile Entity Attribute | Engineering Change Order EBO | CCAP_ECN_MAINTAIN<br>SAP Entity:<br>Attribute:Description | Comments |
|---|---|---|---|
| CHANGE.CHANGE_NUMBER | \\Identification\ ID | AENR_API01:CHANGE_NO:Change Number | Change Notice |
| CHANGE.DESCRIPTION | \\Identification\ Description | AENR_API01: DESCRIPT: Change number description | |
| CHANGE.RELEASE_DATE | InitiationDate | AENR_API01:VALID_FROM:Valid from date | |
| ECO/MCO/SCO.<br>Affected Item.<br>Effectivity Date | EngineeringChangeOrderLine/Effective Date | AEDT_API01:ALT_DATE:Alternative date external key | |
| CHANGE.REASON | \\Identification\Revision\Reason | AENR_API01:REASON_CHG:Reason for change | |
| | \\Identification\Status | AENR_API01:STATUS:Status of change number | |

| Agile Entity Attribute | Engineering Change | CCAP_ECN_MAINTA | Comments |
|---|---|---|---|
| Changes .Affected Items. New Revision | \\EngineeringChangeOrderLine\EffectivityControlItem<br>Reference\Revision\Number | REVLV:Revison level | |
| Agile Entity Attribute | Engineering Change Order EBO | CCAP_REL_LEVEL_MAINTAIN | |
| CHANGE.CHANGE_NUMBER | \\Identification\ ID | AENNR<br>Change Number | |
| ECO/MCO/SCO.<br>Affected Item. Item Number | \\EngineeringChangeOrderLine\RevisedItem | MATNR<br>Material Number | |
| Change Orders Class.Affected Items. New Revision | \\EngineeringChangeOrderLine\EffectivityControlItemReference\Revision\Number | REVLV<br>Revision level | |

# Appendix D: Queue Management

This appendix contains information on the following:

- Queue Schema
- Queue Controller
- Queue Monitor

The Queue Management feature in this pre-built integration caters to the following requirements:

- Event to produce filtered payload to a file destination or JMS destination.
- Payload is defined using a standard XSD.
- 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 filters configured for the ACS Event defined by Agile provided AXML schema definition. Also the ACS transmits the messages in the order in which the ATOs are released.

- Queue to manage messages.
- Queue Monitoring UI to enable reordering and resubmitting unprocessed messages.

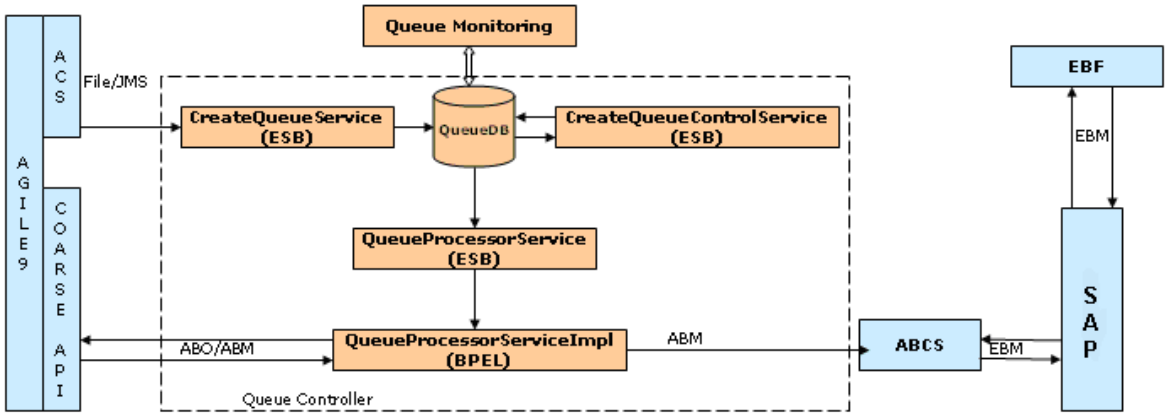
  The Queue manages the payloads based on the Business Process for which the message is produced by the Event.
- Queue controlling mechanism.
  - Triggers the Business Flow based on the business process of message.
  - Processes the messages sequentially depending upon the order specified in the message (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 Queue Manager, see *Agile PLM Integration Pack for SAP Release 3.1 - User Guide* at http://www.oracle.com/technology/documentation/

# 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 Re ordering of priorities of the Queue Messages. Also provides the facility to re submit the un processed messages.




Queue Monitoring

## Queue Schema

To support the queue monitoring 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 messages. Also since all the pending messages are stored in the Queue table, they could be reordered.

## Queue DB Details

The Queue Schema has the following tables:

| | |
|---|---|
| ECO_QUEUE | This table holds the data of Process ECO. The PROCESS_TYPE column is used as identifier for Process ECO. |
| ECO_QUEUE_CONTROL | This table stores the details about the rows that are currently under processing state. |
| ECO_QUEUE_TABLE_CRITERIA | This table contains the data needed for a criteria. One set of criteria forms a filter. |
| ECO_QUEUE_TABLE_FILTER | This table contains the data needed to form a filter, i.e., the criterion to be used to form a specific filter. |
| ECO_QUEUE_STATUS | This table holds the data to control the simultaneous processing and suspending the Queue. By changing the values in the ECO_QUEUE_STATUS column, the number of simultaneously processed ECOs can be changed. |

The structure of ECO_QUEUE_STATUS table is:

| ECO_QUEUE_STATUS_ID | ECO_QUEUE_STATUS | Description |
|---|---|---|
| 1 | 1 or 0 | The status of the Queue - in suspended or resume mode. |
| 2 | 1 | The count of rows that can be processed simultaneously for Process ECO.<br>A value of 1 means sequential processing. |
| 3 | 5 | The count of rows that can be processed simultaneously for Validate ECO. |

## 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

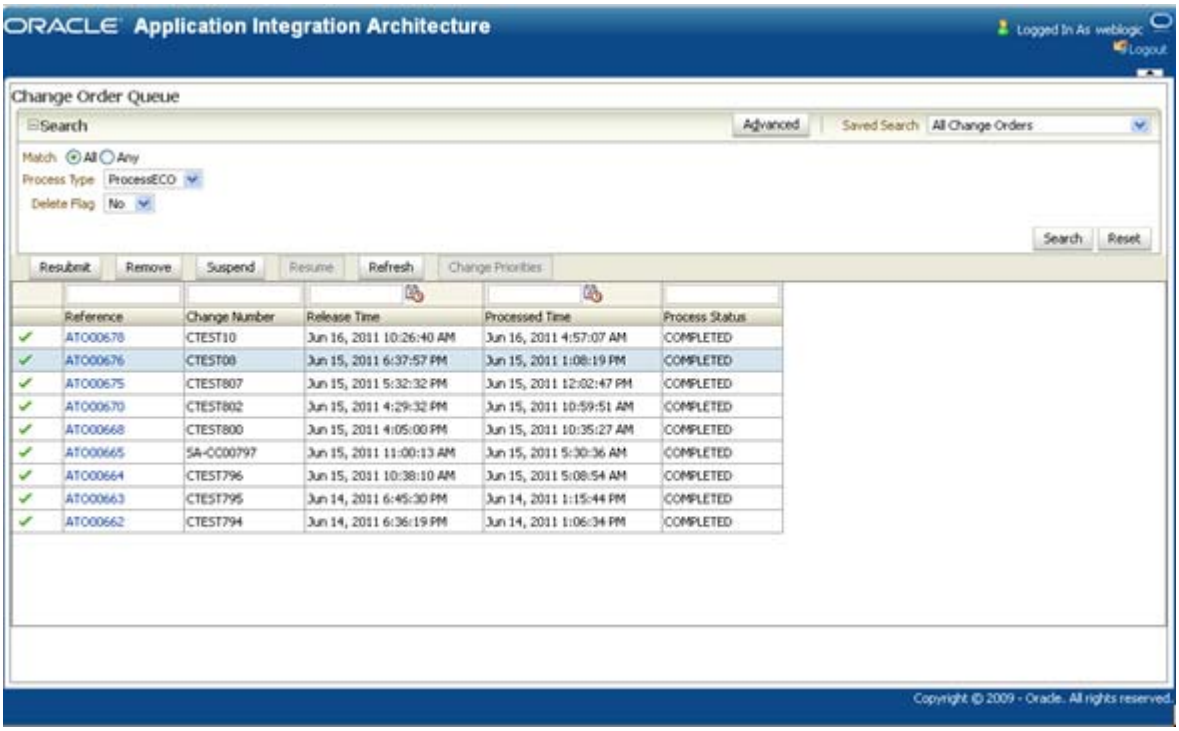
Once the processing of a message in control table is complete, insert the highest priority queue message from the queue table to the control table.

In case the Integration flow errors out, the queue manager will wait until the message is resubmitted or removed.

## Queue 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 awaiting processing and facilitates you to reorder their sequence of processing.

For the first time after deployment, the Queue is in Suspended state. You are required to resume it. For complete details on Queue Monitor, refer Agile to SAP Integration User Guide.




# Queue Manager Services

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

CreateQueueService

1. CreateQueueControlService

2. QueueProcessorService
3. QueueProcessorServiceImpl

### CreateQueueService

The CreateQueueService is implemented as an Mediator 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 Mediator header transformation extension functions.

- PROCESS_STATUS is Pending for the newly inserted row.
- PROCESS_PRIORITY is captured from the file name. (ACS can be configured to append an default order for the file name)

### CreateQueueControlService

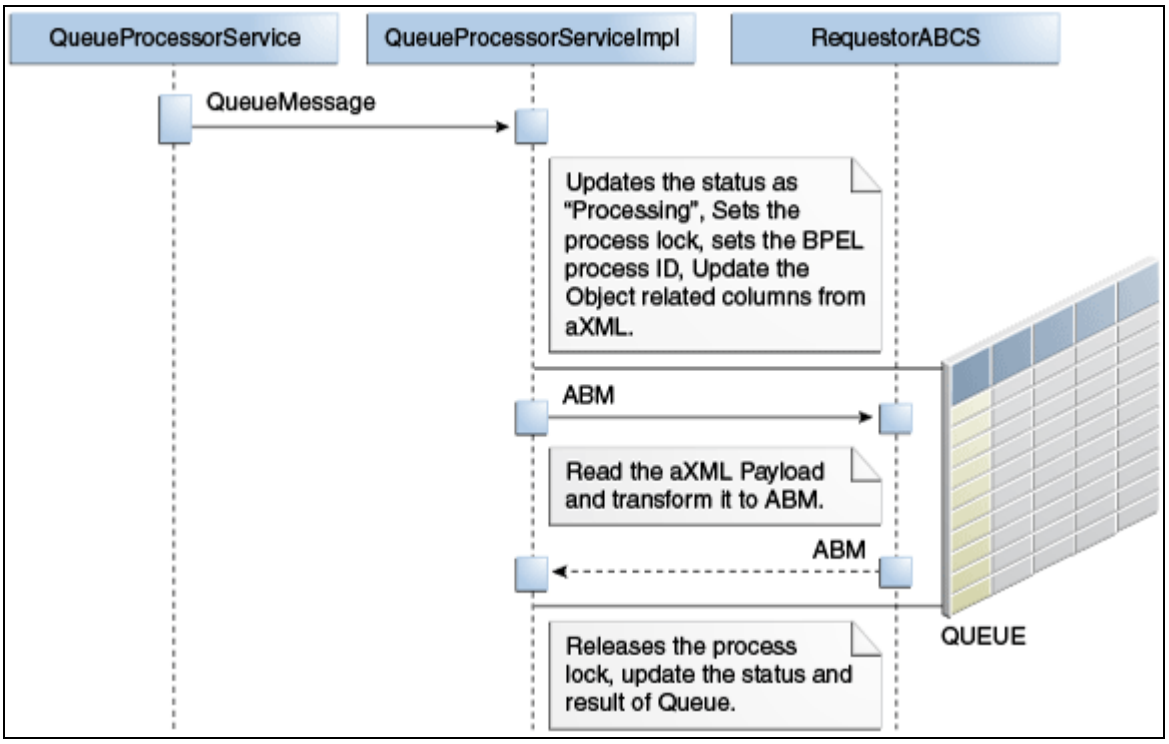
The CreateQueueControlService is implemented as an Mediator Routing Service. A DB Adapter polls on the QUEUE_CONTROL_TABLE table. If there are no rows which are in Pending status, the CreateQueueControlService invokes a DB Adapter service which executes a custom SQL. This SQL identifies the highest priority 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.

QueueProcessorService

The QueueProcessorService is implemented as an Mediator 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.

QueueProcessorServiceImpl




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.

| # | 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<br>PROCESS_STATUS = Processing<br>PROCESS_ID = BPEL Process Id<br>PROCESS_LOCK = 1 |
| 3. | Transform Agile Data(AXML) to ABM | The QueueMessage will have the Agile Data payload which is transformed to ABM |
| 4. | Invoke RequestorABCS | QueueProcessorServiceImpl invokes the |

| # | Name | Step Description |
|---|---|---|
| | | 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 Composite*. 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 Mediator.

**Note**: 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 and EBF. 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.

# Appendix E: Troubleshooting

This appendix covers common troubleshooting issues:

- Engineering Changer Order
- Installation
- Queue
- SAP

## Engineering Change Order Issues

These are the common ECO issue and solutions:

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

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

2. **Issue**: For the Item Cost update update flows, the attributes in Agile 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 attributes in AIAConfigurationProperties.xml are correctly set.

See, Chapter 4 in Agile PLM-SAP Design to Release - Implementation Guide.

# Installation Issues

These are the common installation issue and solutions:

1. **Issue**: While installing the AIA FP in the Linux box, deployment of configurations/deploying the PIP did not succeeded due to SOA server was being unable to restart itself automatically using the deployment script given in the AIA FP.

   **Solution:** To resolve this installation issue:

   1. Stop SOA server.
   2. Restart the SOA server from this location:

      $Middleware_Home\user_projects\domains\soa_domain\bin>.\startManagedWeblogic.sh soa_server1
   3. Click on the retry button on the configuration wizard.

# Queue Issues

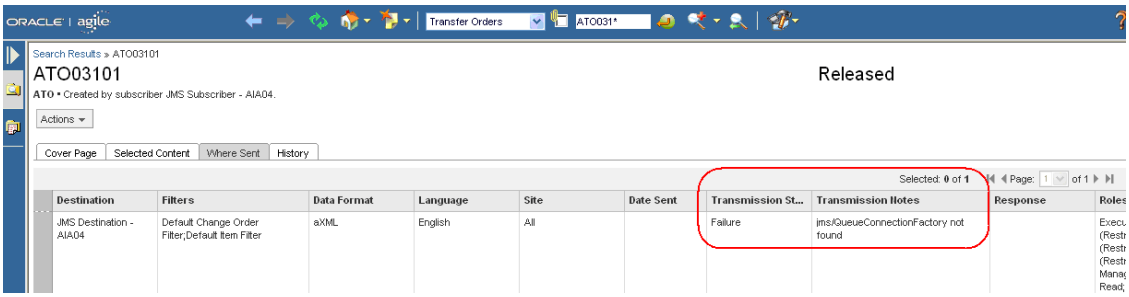
These are the common queue issue and solutions:

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

   **Solution:** To resolve this queue issue:

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

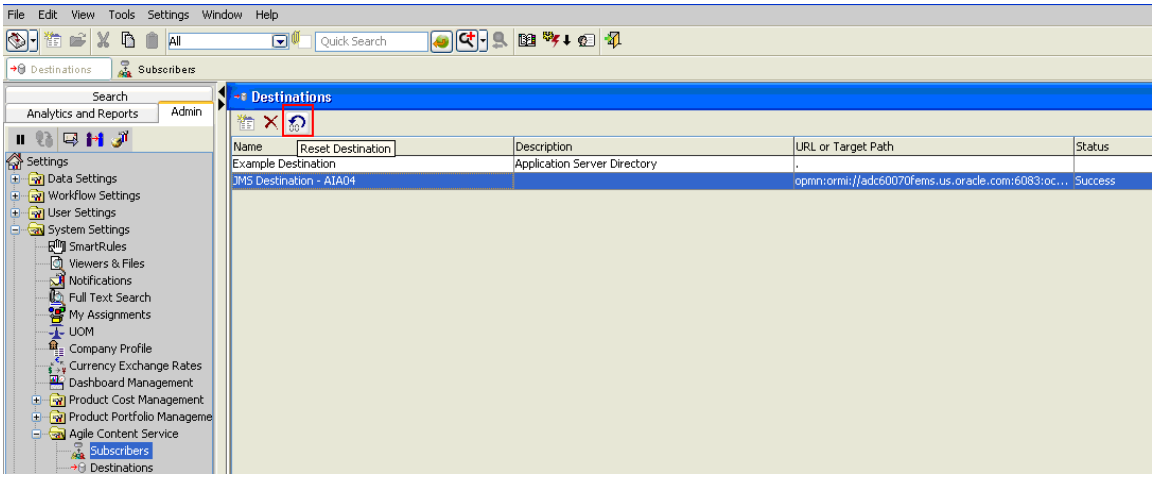
   b. If it shows a *Failure* message, this implies that an error occurred while ACS was processing the publishing of the data to JMS destination. The error message is specified in the Transmission Notes column.




   b. 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.

   c. To reset the destination, in Agile Java client, go to **Admin** > **System Settings** > **Agile Content Service** > **Destinations**.

   d. Select the particular destination and click the Reset icon. The Reset icon is next to the **Delete** icon.

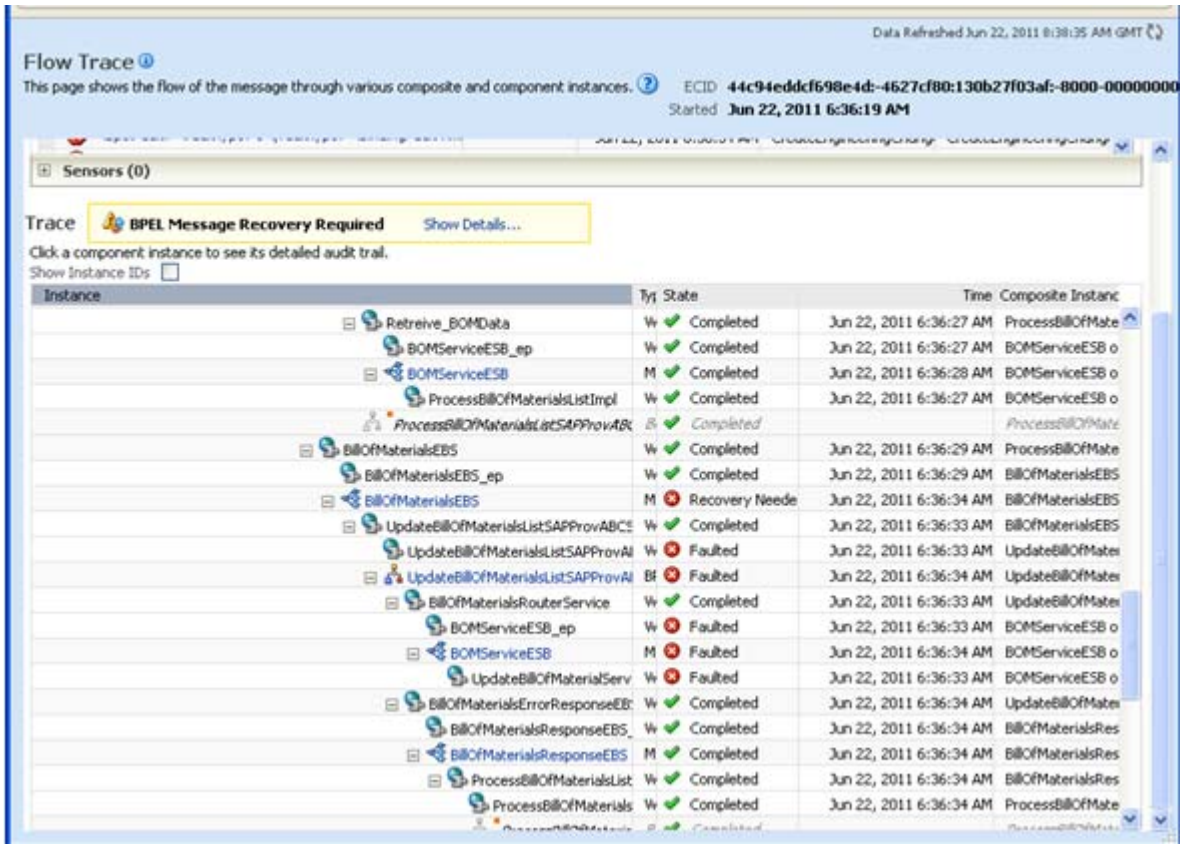



After resetting the Destination, test the Destination to ensure the Test is Successful for the Destination. In case it fails, this has to be resolved, mainly by ensuring all the ECO Queue settings are correct and SOA Port specified in the URL is correct.

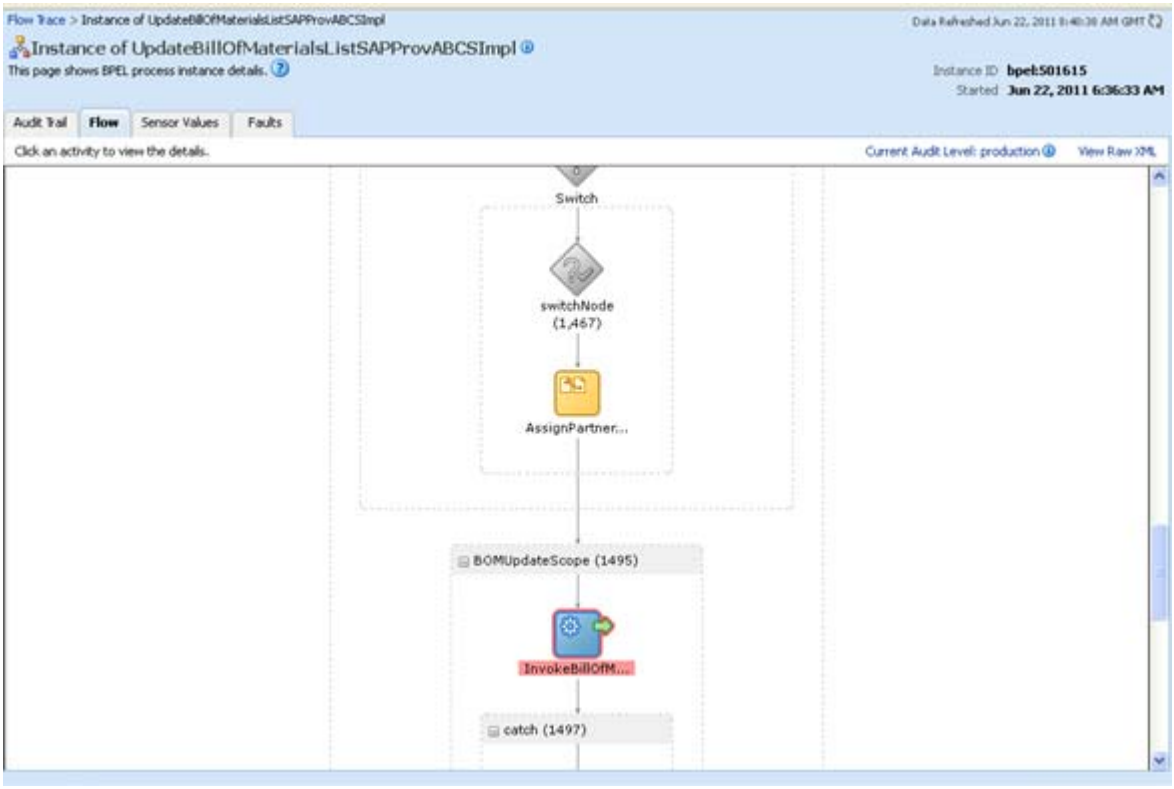
If the status of the ATO transfer is **Success**, it implies that the ACS is publishing the data to JMS queue was successful. Then we need to troubleshoot in the BPEL console.

a. Navigate to the Weblogic EM Console: `http://<host name>:<port number>/em`

b. Click the **Instances** tab. Check for the instance of the *UpdateBillOfMaterialsListSAPProvABCSImpl* for which the error occurred.




c. Click on the Instance Name and go to the *Flow* link. In the BPEL flow shown, find the element at which the error has occurred and then click on the element to view the Activity Audit trail for more details.




# SAP Issues

When implementing the Agile to SAP pre-built integration, you might encounter the following issues.

**Issue**: When updating SAP Item attributes to Agile, if Agile application is down or an error occurs in processing the data, then the data cannot be resubmitted to Agile.

**Solution**: If there is any error in the SAP item attributes update to Agile, an email is sent to the customer/administrator specifying the failure.

To resolve the issue, identify the failed IDOC and resubmit from SAP:

a. Open the error mail that was sent to the customer/user/admin.

b. Click on the hyperlink provided on the error mail.

c. Open the BPELConsole and navigate to the instance that was given in the link.

d. Scroll down to the Assign Fault and check the IDOC number that has failed.

e. Navigate to SAP application and re-submit the IDOC with the item attribute details provided in the email.

**Note:** Please check whether the Agile application server is up and running before initiating the process.

**Issue**: If there is an error in **CreateBillOfMaterialsListSAPProvABCSImpl** with the error description as "*javax.xml.rpc.soap.SOAPFaultException: com.sap.mw.jco.JCO$AbapException: (126) ERROR: Terminate processing*".

**Solution:** Perform these steps:

a. Check whether the components in the Bill Of Materials created in SAP. If the components are not created, add the components to the affected item and then release the ECO.

b. Check whether the components have a valid Item Category, BOM Usage and Effectivity date.

**Issue**: If there is an error in **UpdateBillOfMaterialsListSAPProvABCSImpl** with the error description as "*javax.xml.rpc.soap.SOAPFaultException: com.sap.mw.jco.JCO$AbapException: (126) ERROR: Terminate processing*"

**Solution**: Perform these steps:

a. Check whether the components in the Bill Of Materials created in SAP or not. If the components are not created, add the components to the affected item and then release the ECO.

b. Check whether the components have a valid Item Category, BOM Usage and Effectivity date.

e. Check whether the components defined while creating the Bill Of Material have unique Find Number on Agile side.

**Issue**: If there is an error in **SyncItemListSAPProvABCS** with the error description as "*Exception during SOAP invoke: com.sap.mw.jco.JCO$AbapException: (126) REVISION_LEVEL_EXISTS: REVISION_LEVEL_EXISTS*" even though the process is showing as success.

**Solution**: This error can be ignored as it is only a check for revision existence for the particular item/material.

**Issue**: If there is an error in **SyncItemListSAPProvABCS** or ReserveItemSAPProvABCS with the error description as "*Exception during SOAP invoke: java.lang.Exception: BapiError: Enter a material type; nested exception is: javax.xml.rpc.soap.SOAPFaultException: java.lang.Exception: BapiError: Enter a material type*"

**Solution**: Check whether the ITEM_PRIMARYCLASSIFICATIONCODE DVM is mapped correctly. If not modify the same and re-submit from the queue management console.

Check whether the REFERENCEMATERIALS is mapped with the corresponding material on the SAP system.

**Issue**: If there is an error in **SyncItemListSAPProvABCS** with the error description as "*Exception during SOAP invoke: com.ibi.sap.SapAdapterException: Field: REV_LEVEL_NEW exceeds defined in structure INPUT allowed length 2: ABC; nested exception is: javax.xml.rpc.soap.SOAPFaultException: com.ibi.sap.SapAdapterException: Field: REV_LEVEL_NEW exceeds defined in structure INPUT allowed length 2: ABC*"

**Solution**: Check whether the revision level defined for the affected item is 2 characters.

**Issue**: MediatorMessageProcessingFailed due to Timed out exception

**Solution**: Remove the ECO from the queue with processing state and re-release the ECO from Agile.