|  |
| --- |
| A picture containing drawing  Description automatically generated  Business / Technical Brief |
| Design Specification  Domino-INT005– INV WMS Receipt Advice -RMA  27th May 2024, Version 1a.  Copyright © 2024, Oracle and/or its affiliates  Confidential – Oracle Restricted |



Document Control

Change History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Description |
| 22-May-2024 | Amit Sharma | 1a | Initial Version |
| 28-Jun-2024 | Guna and Iqbal | 1a | Review and Amended the Cross functional points from OM perspective |
| 06-June-2024 | Akanksha Sachan | 1b | Technical Design Update |
|  |  |  |  |

## Reviewed By

|  |  |  |
| --- | --- | --- |
| Date | reviewer | role |
| 15-06-2024 | Rakesh Mishra | Technical Lead |
|  |  |  |
|  |  |  |
|  |  |  |

Table of contents

[Document Control 2](#_Toc169777905)

[Change History 2](#_Toc169777906)

[Reviewed By 2](#_Toc169777907)

[Introduction 4](#_Toc169777908)

[Purpose 4](#_Toc169777909)

[Scope 4](#_Toc169777910)

[Functional Overview 4](#_Toc169777911)

[Business Objectives 5](#_Toc169777912)

[Process Overview 6](#_Toc169777913)

[Mapping 7](#_Toc169777914)

[Pre-Conditions 7](#_Toc169777915)

[1. The items should be interfaced to WMS from PDH (INT001) 7](#_Toc169777916)

[Risk Matrix 7](#_Toc169777917)

[Other Integrations Design Dependency 7](#_Toc169777918)

[Frequency 8](#_Toc169777919)

[Open Items 8](#_Toc169777920)

[Technical Overview 8](#_Toc169777921)

[Scope 8](#_Toc169777922)

[Glossary 8](#_Toc169777923)

[Document References 9](#_Toc169777924)

[Sample Data & Mapping Template 9](#_Toc169777925)

[Frequency 9](#_Toc169777926)

[Assumptions 9](#_Toc169777927)

[High Level Approach 10](#_Toc169777928)

[Pre-Requisites 11](#_Toc169777929)

[Integration Analysis – XxScmReceiptRmaRequestErpToWms, XxScmReceiptRmaRequestProcessErpToWms 11](#_Toc169777930)

[Integration Components 11](#_Toc169777931)

[Required Connections 12](#_Toc169777932)

[Integration Outline 12](#_Toc169777933)

[Global Fault 22](#_Toc169777934)

[Migration Steps 24](#_Toc169777935)

[Unit Testing 24](#_Toc169777936)

[Open and Closed Issues 24](#_Toc169777937)

Introduction

Purpose

The objective of this document is to provide a Design Specification to:

* Provide the necessary information with enough detail so that the Project team, have a clear understanding of the design and functional requirements.

## Scope

This document specifies the business context, functional requirements, and details to integrate Oracle Fusion to Oracle Cloud WMS for Return Material Authorization Orders interfaced as ASNs with the ASN type as “RMA”.

A warehouse management system (or WMS) is a software application designed to support warehouse or distribution center management and staff. They facilitate management of available resources to move and store materials into, within, and out of a warehouse, while supporting staff in material movement and storage.

Coordinating the communication with these systems to support purchase order receipts, returns to vendors, internal material transfers, inventory transactions, and sales order shipments is a substantial integration effort.

Inventory Management Cloud supports a central integration framework across Oracle Procurement Cloud, Oracle Order Management Cloud, and Oracle Inventory Management based on web services to support the interactions between WMS and Fusion Cloud.

Using these services, Inventory Management Cloud communicates Receipt Advice to Oracle Cloud WMS for processing.

## Functional Overview

Domino process flow requires the Return Order to initiated in Order Management system to facilitate the return of physical product from Customer in pre-defined facility. The return order is created as source document in order management. Warehouse person will Generate Receipt Advice from Inventory to WMS for further execution of returns. The sourcing details such as item number, return quantity, return order, and return to warehouse etc. are get published to WMS.

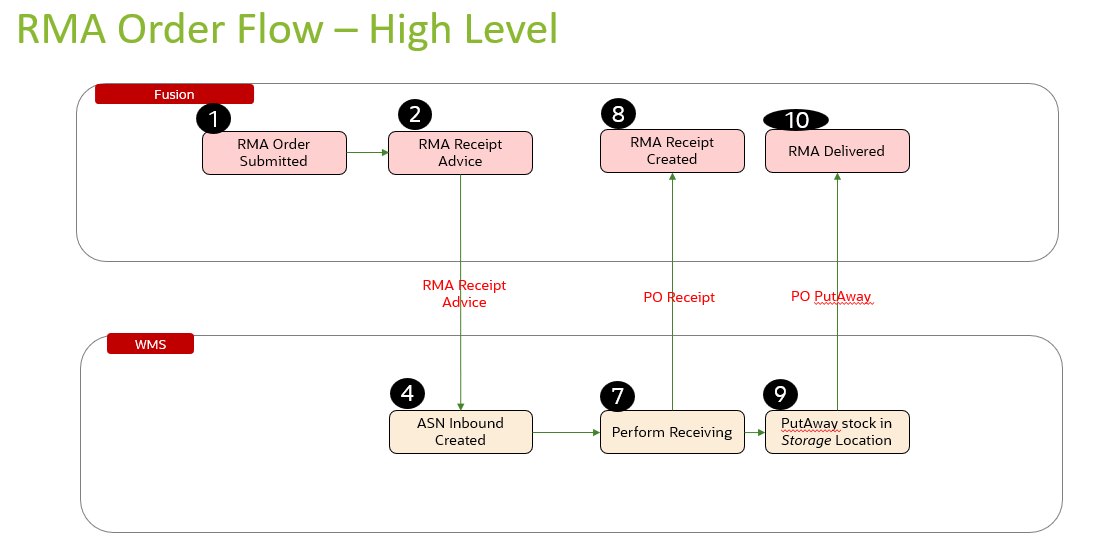
Allows for below transactions:

* Creation

Business Objectives

When integrating Oracle Inventory Management Cloud with a warehouse management system (or WMS), there is the requirement to send the RMA information as Receipt Advice to allow the Receiving Inventory Organization in WMS to Receive and Put away the RMA stock.

**RMA Business Flow:**



The above picture depicts the general steps in the Return Material Authorization (RMA) business flow. The steps initiate in Fusion Order Management with the creation of the RMA. This RMA is then interfaced to WMS via OIC integration to create an Inbound Shipment. In WMS the user receives the requested items and locates them into a put away location. The receipt information along with the put away are then interfaced back to Fusion to update the ASN and adjust the inventory stock quantities to reflect the received quantity.

The Generate Receipt Advice from Inventory to WMS will push the RMAs to WMS for further execution in WMS. The details such as RMA (ASN) number, as item number, return quantity, return order, and return to warehouse etc. are get published to WMS.

In WMS, warehouse employee is responsible in performing below operations:

* Creation of Receipt
* Put Away of material in specified return facility along with additional details of Locator/Serials/Lot if any.

Once Put Away is performed in WMS, the item is physically returned into warehouse. An inventory person will now send receipt confirmation details (detailed put away material transaction) to Inventory and same is responsible in progressing the return sales order to Credit the customer or closure without credit.

## Process Overview

1. Create RMA in Fusion Order Management.
2. Schedule the process in Oracle Inventory Management “Generate Receipt Advice”. The scheduled process Generate Receipt Advice publishes an event to notify the WMS that receipt advice is ready to be interfaced. The web service Send Receipt Advice provides the ability for the WMS to receive the actual receipt advice with the expected shipment lines for Return Material Authorization (RMAs).
3. After the WMS receives the receipt advice for RMA, an ASN gets created with details like Customer Name, return quantity etc. from return order and performs receiving-related processes including receiving and put away.

Mapping

Pre-Conditions

1. The items should be interfaced to WMS from PDH (INT001)
2. Shipment Type as “RMA” should be created in WMS.

Risk Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Impact  (1 to 5) | Probability  (1 to 5) | Mitigation |
| RMA Interface Failure to WMS due to missing item in WMS | 5 | 2 | Valid error message stating the integration failure. Re-run the item integration first and then the RMA receipt advice integration |
| RMA Interface Failure to WMS due to missing shipment type in WMS | 2 | 2 | Configure the shipment type in WMS and then re-run the RMA receipt advice integration |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Other Integrations Design Dependency

|  |  |  |
| --- | --- | --- |
| Name | Dependency | comments |
| WMS Items (INT001) |  | Items should be interfaced to WMS from Fusion PDH |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Frequency

|  |  |  |
| --- | --- | --- |
| Responsiveness | Frequency | Data volume |
| Batch | ESS ‘Generate Receipt Advice’ and ‘Generate Changed Receipt Advice‘ job runs every 5 mins |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Open Items

This section is used to summarize any open items related to the design.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | open item | data required | status | impact if not closed |
| 001 |  |  |  |  |
| 002 |  |  |  |  |
| 003 |  |  |  |  |
| 004 |  |  |  |  |
| 005 |  |  |  |  |
| 006 |  |  |  |  |
| 007 |  |  |  |  |

# Technical Overview

### Scope

The objective of this document is to provide Domino Printing with a Technical Specification for Outbound Integration for Publishing Return Material Authorization from Fusion to WMS.

### Glossary

|  |  |
| --- | --- |
| **Annotation** | **Definition** |
| **RMA** | **Return Material Authorization** |
| ERP | Enterprise Resource Planning |
| SCM | Supply Chain Management |
| OIC | Oracle integration cloud |
| API | Application Programming Interface |
| SaaS | Software as a service |
| PaaS | Platform as a Service |
| UCM | Universal Content Management |
| ESS | Enterprise Scheduler Service |
| WMS | Warehouse Management System |

### 

### Document References

|  |  |  |
| --- | --- | --- |
| Ref No | Name | File location |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

### Sample Data & Mapping Template

|  |  |
| --- | --- |
| Description | File |
| Data Mapping Template for Fusion to WMS |  |
| Sample WMS XML Payload |  |

### Frequency

|  |  |  |
| --- | --- | --- |
| Responsiveness | Frequency | Volume |
| Batch | Planned as daily with Ad Hoc option | Depends on the RMA orders created in the last 24 hours by users |

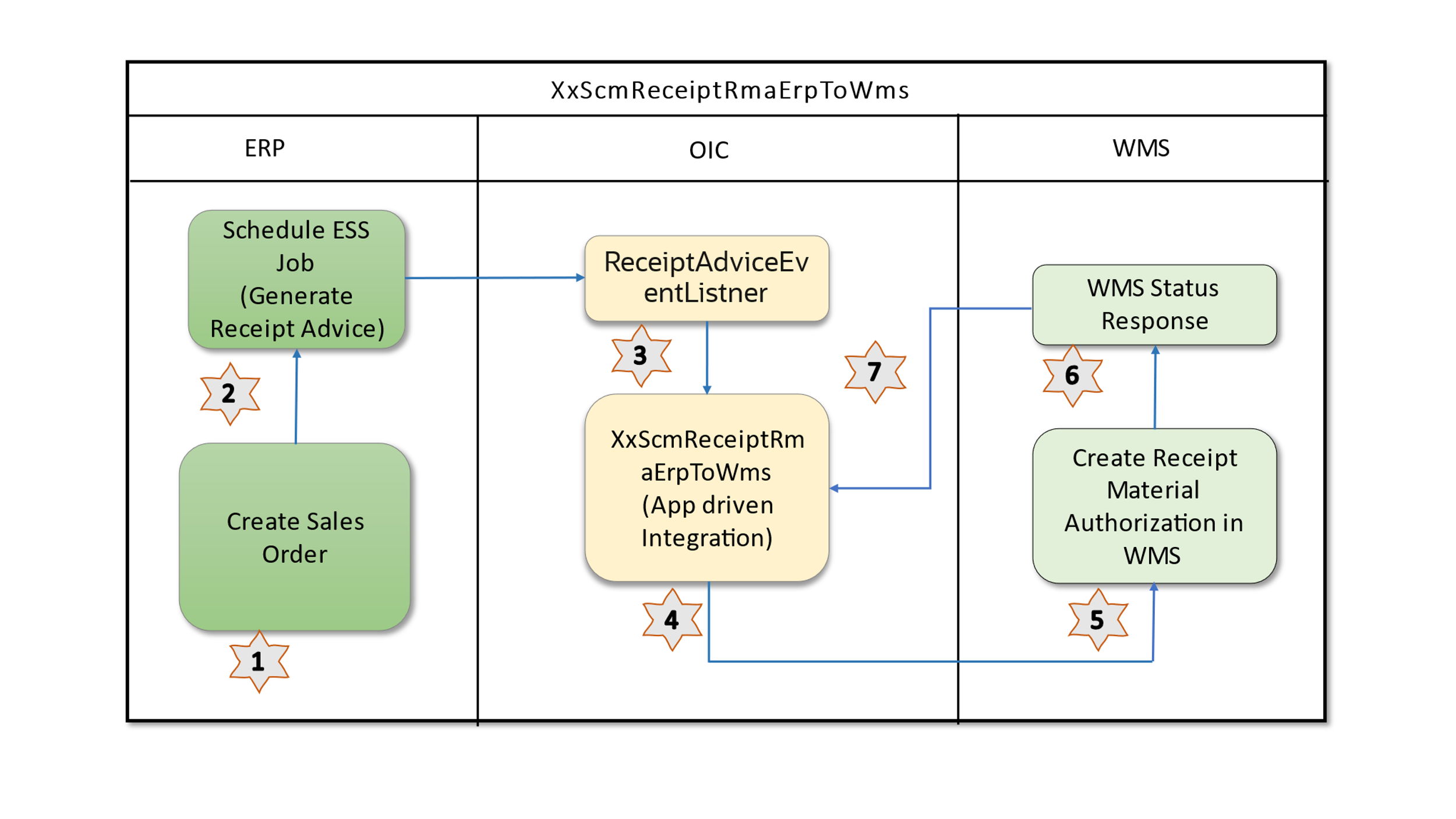
### 

### Assumptions

1. Mapping between Sales Orders in SCM Cloud and WMS Facilities needs to be setup for this integration to extract data.
2. This integration will not take in consideration lines which have been manually assigned to a Shipment in SCM Cloud. It is WMS role to execute shipments so shipments should not be created in SCM Cloud for WMS controlled organizations.
3. This integration will take in consideration lines for inventory items only. Orders for services or recurring charges, etc. which do not use inventory items will not be taken in consideration.
4. Draft sales orders and sales order revisions are not taken in consideration.
5. In case of any success/error in the integration, a notification email will be sent to the mail list configured in the OIC lookup.

### High Level Approach

1. The integration will trigger once the ESS job – “Generate Receipt Advice” is submitted. Integration will be subscribing to the event for “Receipt Advice Event Notification”.
2. “Generate Receipt Advice” job will be scheduled in fusion as per frequency decided to extract the sales order details created after the last successful run of job.
3. Integration will be triggered by subscribing the event Business event “Receipt Advice Event Notification” is applied in the source ERP adapter responsible for the invocation of OIC integration.
4. The payload received from contains from SCM will contains the RMA order details and shipment line details.
5. The payload will be then converted into WMS XML required format by invoking the Rest API echoLgfDataOrder which will do the XML transformation internally.
6. The transformed XML data will be interfaced to WMS cloud using WMS InitStageInterfaceAPI service.



### Pre-Requisites

* Send email and SMS integration *“XxCommonErrorHandler”* should be deployed in the OIC.
* Lookup *“XXDominoCommonLookup”* should be present with all the required details regarding success/error email notify flag, success and error notify ToEmail Address in OIC.
* Lookup *“XxDominoConstantLookup”* should be present with all the required details regarding success and error notify FromEmail Address in OIC.
* *XxScmReceiptRmaRequestErpToWms*, *XxScmReceiptRmaRequestProcessErpToWms* should be deployed in OIC.
* " *XxParkingLotProducer(1.0*)" should be deployed in OIC.
* " *XxGetLookupVariable(1.0)"* should be deployed in OIC.
* BIP Report should be present in the path mentioned in *XXDominoCommonLookup.*

# 

# Integration Analysis – XxScmReceiptRmaRequestErpToWms, XxScmReceiptRmaRequestProcessErpToWms

This Integration component is responsible for exporting items from Fusion ERP to WMS system.

### Integration Components

Below are the required components for this integration.

|  |  |  |  |
| --- | --- | --- | --- |
| **Integration Name** | **Integration Pattern** | **Package Name** | **Lookup Name** |
| XxScmReceiptRmaRequestErpToWms | App Driven Orchestration |  | XXDominoCommonLookup, *XxDominoConstantLookup,*  XxParkingLotRetryConfigLookup |
| XxScmReceiptRmaRequestProcessErpToWms | App Driven Orchestration |  | XXDominoCommonLookup, XxDominoConstantLookup,  XxParkingLotRetryConfigLookup,  XXDominoWmsCompanyLookup |

### Required Connections

|  |  |  |
| --- | --- | --- |
| **Connection Name** | **Connection Type** | **Connection Role** |
| XxDominoErpCloudConnection\_WMSINT | Oracle ERP Cloud | Trigger and invoke |
| XxDominoWMSConnection | Oracle WMS Cloud | Trigger and invoke |
| XxOicRestTrigger | REST | Trigger |
| XxAtbDb | Oracle ATP | Trigger and invoke |
| XxDominolgfDataInvoke | REST | Trigger and invoke |
| XxDominoBIReportConnection\_WMSINT | SOAP |  |

### 

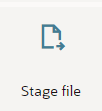
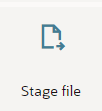
### Integration Outline





App Driven Integration

*XxScmReceiptRmaRequestErpToWms*, *XxScmReceiptRmaRequestProcessErpToWms*



Collect the details of the Sales order for which the Return Material Authorization must be created.

Write Stage file

Read Stage file

Completion of the ESS job triggers the “XxScmReceiptRmaRequestErpToWms” integration to send the

RMA details to WMS Cloud

Write the RMA details from Fusion to ICS

Read the RMA details from ICS to send into WMS cloud.

Create RMA in WMS

The corresponding status of file records (Success/Error) will be sent back by WMS API to User.

ESS Job

***Generate Receipt Advice***

Figure 1: Architecture Diagram

The Following section explains the step – by- step process of publishing items from Fusion to WMS:

1. **Integration** – XxScmReceiptRmaRequestErpToWms
2. ESS Job Parameters: The ESS Job have three parameters which are as below, the integration will be triggered with those values assigned to these parameters.

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Data type** | **Value** |
| Organization | String | Return to warehouse company code. |
| Resend Receipt Advice | Boolean | No – First time publishing the event.  Yes – Tries to re-publishing the event |
| RMA | String | Specify Return Order number |
| RMS Line | String | Specify lines from return order which need to be received.  Ignore if all the eligible lines to be returned |

1. Global Assignment: The Global variable will be the Integration code which will be fetched from maintained lookup –“XXDominoCommonLookup*”* with the reference of this lookup all other local variables will fetched which are given below.

|  |  |  |
| --- | --- | --- |
| **Variable name** | **Data type** | **Value** |
| IntegrationCode | String | dvm:lookupValue('XXDominoCommonLookup','IntegrationCode', ' XXSCMRECEIPTRMAREQUESTERPTOWMS, 'IntegrationCode', '')  IntegrationCode will be fetched from OIC lookup “XXDominoCommonLookup”, with reference to this lookup all other local variables will be fetched.  Value for IntegrationCode variable is Integration identifier. |

1. Local Assignment:

|  |  |  |
| --- | --- | --- |
| **Variable name** | **Data type** | **Value** |
| CurrentPotentionalErrorMessage | String | 'Error while calling Echo LgfData Integration.' |
| IntegrationID | String | dvm:lookupValue('XXDominoCommonLookup', 'IntegrationCode', $IntegrationCode, 'RiceID', '') |
| EmailFrom | String | dvm:lookupValue('XxDominoConstantLookup', 'ID', 'FromEmailId', 'Value', '') |
| ErrNotifyEmail | String | dvm:lookupValue('XXDominoCommonLookup', 'IntegrationCode', $IntegrationCode, 'FailureNotify', '') |
| SucessNotifyEmail | String | dvm:lookupValue('XXDominoCommonLookup', 'IntegrationCode', $IntegrationCode, 'SuccessNotify', '') |

1. Scope: Event Response
   1. WriteStage: This is a stage write action. In this step, fusion payload is being captured and stored in temp file.

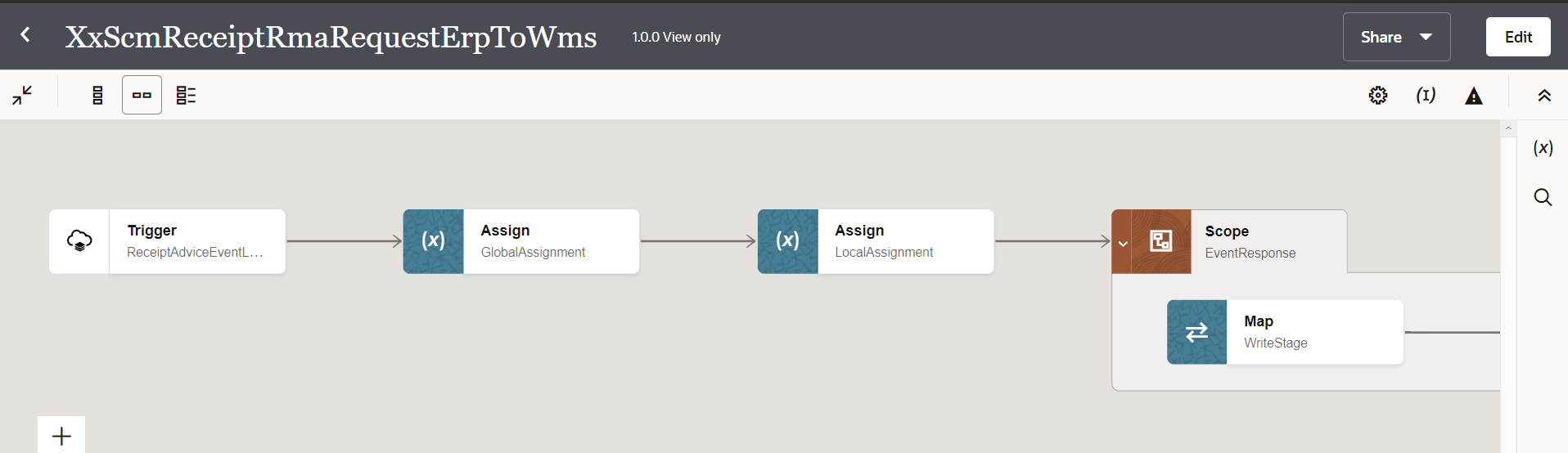
Map Activity WriteStage: In this map activity, we are mapping fields of fusion payload with temp file.

* 1. StageRead: This is a stage read action. In this step, fusion payload stored in temp file is being read.
  2. InvokeParkingLotProducer: In this step, integration XxParkingLotProducer(1.0) is being invoked.

Map Activity InvokeParkingLotProducer: In this map activity, parking lot fields are being mapped.

1. Default Handler: This step is there to capture errors other than API errors.
   1. InvokeErrorHandler: In fault handler section, process will call “*XxCommonNotificationHandler*” local integration, which is to capture the error occured, if there is any service exception occurred in ERP cloud will be handled in scope fault handler and Email Notification will be sent to concern team.

Map Activity InvokeErrorHandler: Error details to be mapped in this map activity.



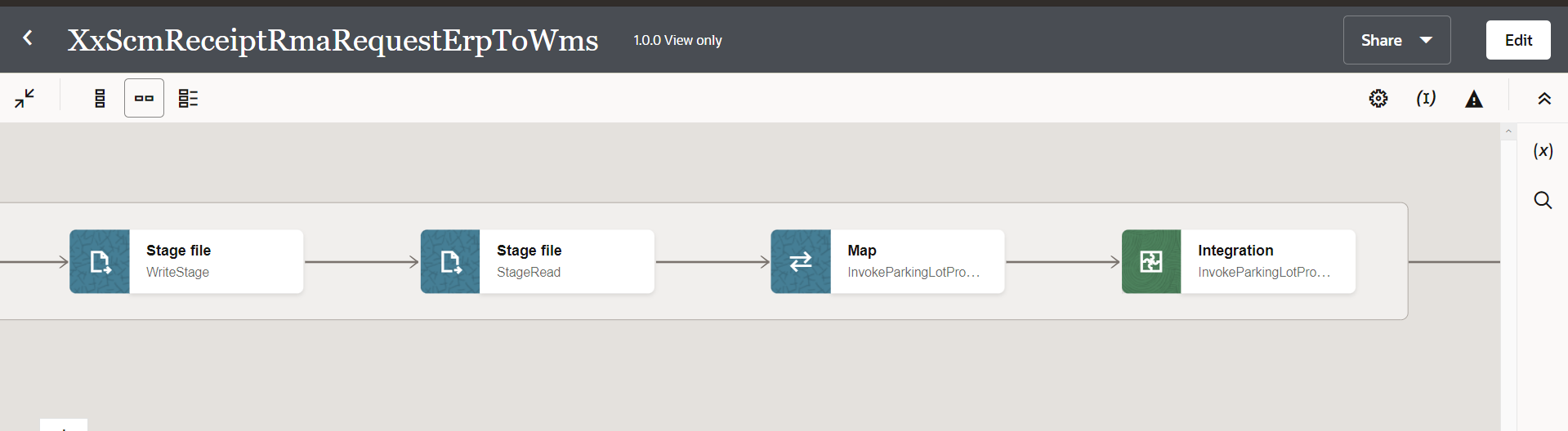


Figure2. Main Code Flow Snippet

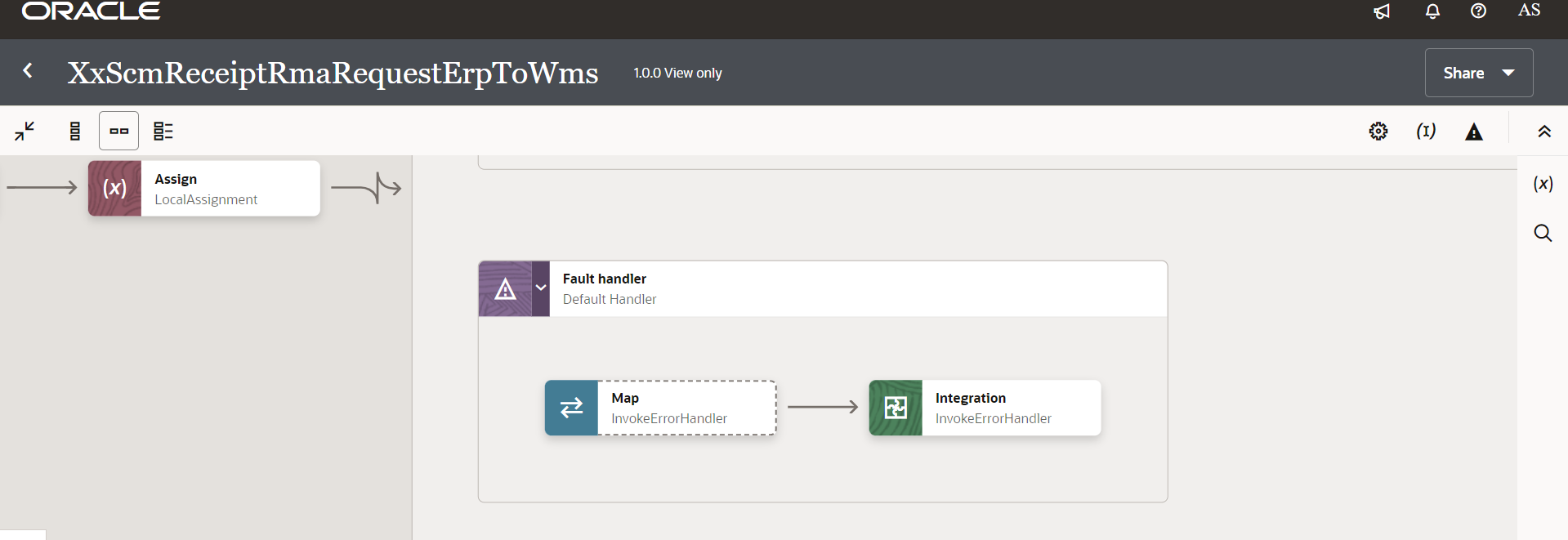


Figure 3. Scope Fault Handler

1. **Integration** -XxScmReceiptRmaRequestProcessErpToWms
2. Main: This is a trigger step to trigger the integration.
3. GlobalVariabledef: The Global variable will be the Integration code which will be fetched from maintained lookup – “XXDominoCommonLookup*”* with the reference of this lookup all other local variables will fetched which are given below.

|  |  |  |
| --- | --- | --- |
| **Variable name** | **Data type** | **Value** |
| IntegrationCode | String | dvm:lookupValue('XXDominoCommonLookup','IntegrationCode', ' XXSCMRECEIPTRMAREQUESTPROCESSERP, 'IntegrationCode', '')  IntegrationCode will be fetched from OIC lookup “XXDominoCommonLookup”, with reference to this lookup all other local variables will be fetched.  Value for IntegrationCode variable is Integration identifier. |
| varRMANumber | String | ‘’ |

1. GetVariables: This step will invoke integration ‘XxGetLookupVariable(1.0)’.

Map Activity GetVariables: In this Map activity, ‘Integration Name’ received from Integration Metadata will be mapped to ‘Integration Code’ in Fetch Variable (request). Once ‘Integration Code’ gets passed to ‘XxGetLookupVariable(1.0)’ integration, values of variables can be retrieved.

1. GetPayload: This step will fetch the fusion payload from ATP db which is in JSON format.

Map Activity: In this map activity, messageId and group\_type are being mapped which is stored in ATP db to fetch fusion payload.

1. StageWrite: This is a stage write operation to write fusion payload being fetched from ATP.

Map Activity: In this map activity, payload coming from ATP is being mapped.

1. StageRead: This is a stage read operation to read file written in previous step.
2. WMSScope:
   1. ForEach loop

AssignRMANumber: In this step, RMA numbers are being concatenated to fetch data from BI Report.

|  |  |  |
| --- | --- | --- |
| Variable Name | Data Type | Value |
| varRMANumber | String | fn:concat($varRMANumber,',',$f0\_ReceiptAdviceLine/ns302:ReceiptAdviceLine/ns302:DocumentNumber) |

* 1. InvokeRMAShipmentDtl: This is oracle SOAP connection which will make a call to BI report.

Map Activity: With this mapping activity we are passing report absolute path and parameters to BI report to fetch the report data.

* 1. AssignBIResponse: In this step we are decoding report data and assigning it to a variable as below:

|  |  |  |
| --- | --- | --- |
| **Variable name** | **Data type** | **Value** |
| varReportOutput | String | oraext:decodeBase64ToReference($InvokeRMAShipmentDtl/ns302:runReportResponse/ns302:runReportReturn/ns302:reportBytes) |

* 1. StageReadRMAShipmentDtl: This step involves using Stage File action to read *varReportOutput* assigned in previous step.
  2. EcholgfData: This is Rest API connection which will invoke the local integration “XXDominoWMSEchoLgfData” which is used to convert the payload received from fusion ERP to WMS Request format. It is internal XML transmission. Response of this API will be sent to WMS as input parameters.

Map Activity EcholgfData – With this mapping activity the payload received from fusion ERP will be mapped to WMS fields. The mapping will be as per the mapping document provided in Sample Data & Mapping Template section in this document.

* 1. postInitStageInterface- This step will call the WMS common api- ***“/wms/api/init\_stage\_interface/”*** which is responsible to post the data on into WMS system. In case of any error while calling this API, notification will be sent through common error handler integration to respective team. There are three query parameters for added for this API to send the request.

Map Activity postInitStageInterface: XML data is mapped in this step as mentioned below.

Query Parameters-

* + 1. Xml\_data- This will be mapped with the response of echoLgfDataOrder which will send the data as parameter to WMS API to post the data in WMS side.

The request will be sent to WMS API as – ***Send query parameters as form data in message body.***

Once the data is successfully loaded into WMS side, WMS API will send consolidated response for all records in payload back to OIC, in case of any error while posting the data to WMS same will be captured by WMS API and will be send to OIC.

1. NotifyUser: Once the data is posted into WMS side and all the operations in Integration are completed normal then success email notification will be sent to respective team through *“*XxCommonNotificationHandler(1.0.1)”.

With map activity, process will map the below values to common error handler Integration.

|  |  |
| --- | --- |
| **Request Parameters** | **Values** |
| IntegrationCode | $IntegrationCode |
| IntegrationID | dvm:lookupValue ("XXDominoCommonLookup", "IntegrationCode", $IntegrationCode, "RiceID", "" ) |
| InstanceID | $self/nsmpr3:metadata/nsmpr3:runtime/nsmpr3:instanceId |
| ErrorCode | NULL |
| ErrMsg | NULL |
| EmailFrom | dvm:lookupValue ("XxDominoConstantLookup", "ID", "FromEmailId", "Value", "" ) |
| EmailTo | dvm:lookupValue ("XXDominoCommonLookup", "IntegrationCode", $IntegrationCode, "SuccessNotify", "" ) |
| EmailSub | concat ("dvm:lookupValue('XxDominoConstantLookup', 'ID', 'InstanceName', 'Value', '')", " - ", "SUCCESS", " - ", dvm:lookupValue ("XXDominoCommonLookup", "IntegrationCode", $IntegrationCode, "IntegrationDesc", "" ), " - ", $self/nsmpr9:metadata/nsmpr9:runtime/nsmpr9:instanceId ) |
| Logging |  |

1. InvokeProcessedParkingLotIntegration: InvokeParkingLotProducer: In this step, integration XxParkingLotProducer(1.0) is being invoked.

Map Activity InvokeParkingLotProducer: In this map activity, parking lot fields are being mapped.

**WMS Request and Response Schemas:**

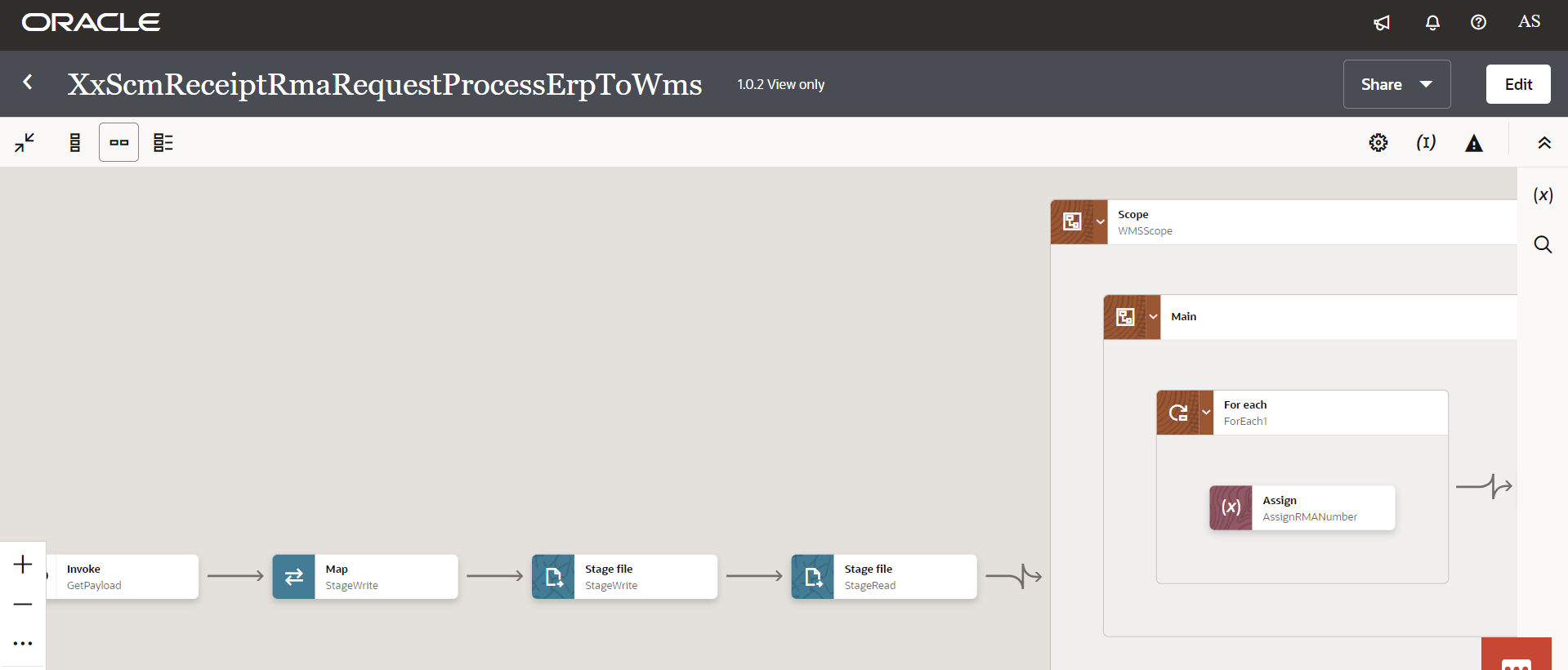
** **

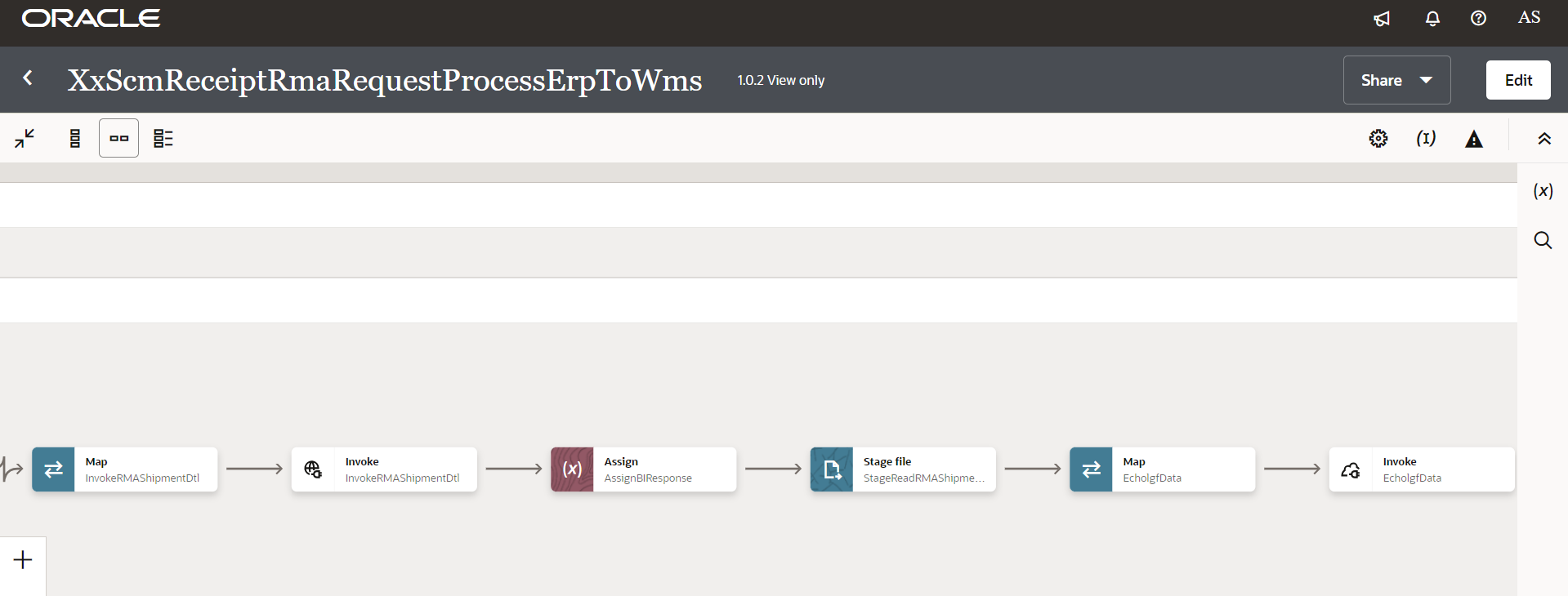
**Sample WMS Response:**

|  |  |
| --- | --- |
| **WMS Response** | **Response Message** |
| SUCCESS-True | <executeResponse xmlns="http://xmlns.oracle.com/cloud/adapter/REST/postInitStageInterfaceOrder\_REQUEST/types">  <root xmlns="http://xmlns.oracle.com/cloud/adapter/nxsd/surrogate/response/postInitStageInterfaceOrder/">  <success>True</success>  <response>  <message>Stage table processing complete</message>  </response>  </root>  <HTTPHeaders/>  </executeResponse> |
| Success-False | Error while initializing stage interface.  <![CDATA[<?xml version="1.0" encoding="utf-8"?>  <root><success>False</success><response>  <message>Processing failed for entity order  and file group 2023-02-20T11:19:47.157Z\_20230220061949165  </message><errors><error><key>107-1</key>  <msg>Exception in process\_stgdtl: Dtl Error:  Invalid item: -----  [Code: EPT2007~^~]</msg></error></errors></response></root>.A 400 Bad Request Error indicates that the target service is unable (or refuses) to process the request sent by the client (Oracle Integration Cloud), due to an issue that is perceived by the server to be a client problem. You can trace the cURL representation of the request sent to the target service from the Oracle Integration Cloud server logs.  Try invoking the target service using cURL. It may also be that one of the intermediaries (proxy, LBR) could be returning this code. ]]> |

In case of WMS API invocation error, it will be mostly due to data issue sent from fusion. In the case WMS API will reject the entire file sent back the response, then the data needs to correct in fusion and resend the data in next run of integration.







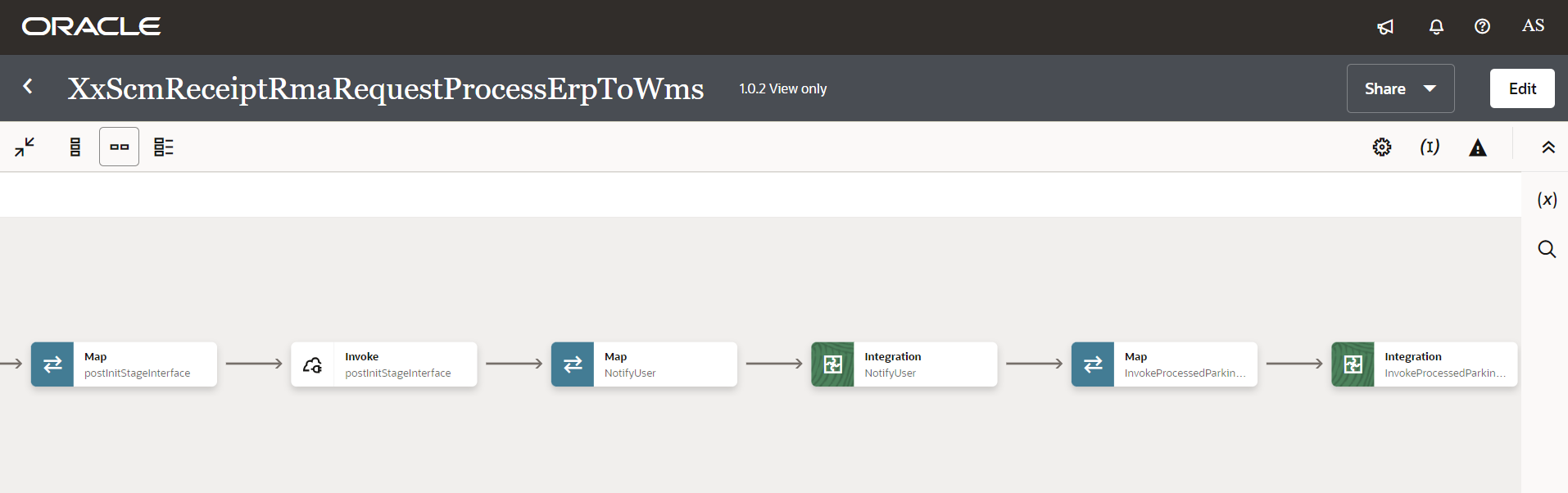


Figure 4: Main flow code snippet

1. Scope fault handler:
   1. InvokeParkingLot: In fault handler section, process will call “XxParkingLotProducer(1.0)” local integration.

Map Activity InvokeParkingLot: In this map activity, parking lot fields are being mapped.

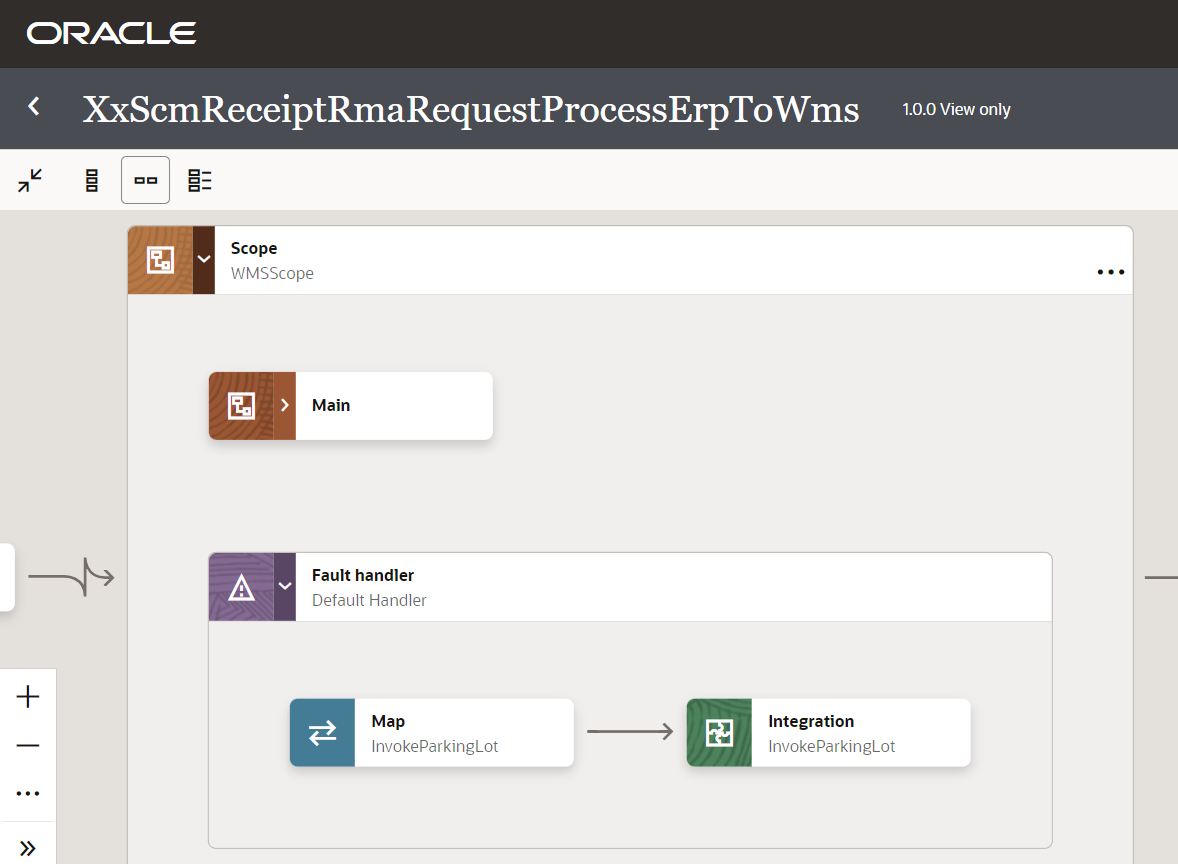


Figure 5: Scope Fault code snippet

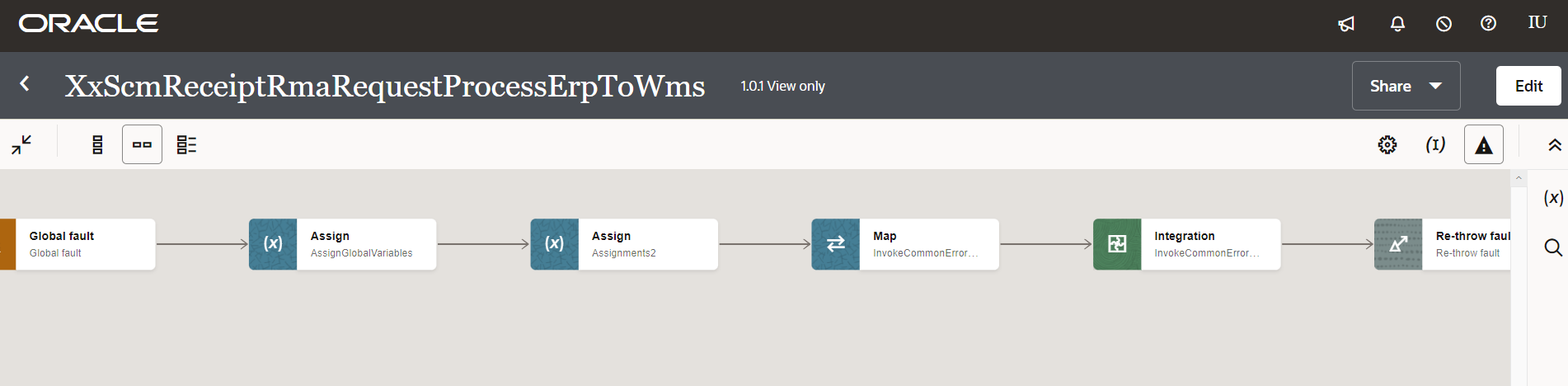
### Global Fault

1. The global fault handler is responsible for the whole execution scope and processes the exceptions occurred in the individual stages.
2. AssignGlobalVariable: The Global variable will be the Integration code which will be fetched from maintained lookup “XXDominoCommonLookup*”* with the reference of this lookup all Error variable values will fetch which are given below.
3. Assignment2: Error Variables have assigned in this step.
   1. Using assign activity, variables will be declared to map the values required for invoking common error handler integration.

|  |  |
| --- | --- |
| **Variable** | **Values** |
| IntegrationID | This value will fetch from common lookup- **XXDominoCommonLookup** with the reference of integration code value from Global variable. |
| ErrNotifyEmail | This value will fetch from common lookup- **XXDominoCommonLookup** with the reference of integration code value from Global variable.It will be failure notify email id. |
| EmailFrom | This value will be fetched from constant lookup-XxDominoConstantLookup |

1. InvokeCommonErrorHandler:
   1. The global fault handler will invoke the local integration “*XxCommonNotificationHandler*” which captures the error occurred any step in overall integration flow and notify over email.
   2. Using the map activity, process would map the below values required for “*XxCommonNotificationHandler*”.

|  |  |
| --- | --- |
| **Request Parameters** | **Values** |
| IntegrationCode | $IntegrationCode |
| IntegrationID | $IntegationID |
| InstanceId | $self/nsmpr1:metadata/nsmpr1:runtime/nsmpr1:instanceId |
| ErrCode | $GlobalFaultObject/nsmpr0:fault/nsmpr0:errorCode |
| ErrMsg | $GlobalFaultObject/nsmpr0:fault/nsmpr0:reason |
| EmailFrom | $EmailFrom |
| EmailTo | $ErrNotifyEmail |
| EmailSub | concat ("dvm:lookupValue('XxDominoConstantLookup', 'ID', 'InstanceName', 'Value', '')", " - ", "ERROR", " - ", dvm:lookupValue ("XXDominoCommonLookup", "IntegrationCode", $IntegrationCode, "IntegrationDesc", "" ), " - ", $self/nsmpr1:metadata/nsmpr1:runtime/nsmpr1:instanceId ) |
| Logging | Y |

****Figure 6: Global Fault code snippet

4

3

2

1

### Migration Steps

2

4

3

1. Import “*XxScmReceiptRmaRequestErpToWms\_01.00.0000.iar”*”, “*XxScmReceiptRmaRequestProcessErpToWms* *\_01.00.0000.iar*” in OIC environment.
2. Update the OIC Connections with Password, details as per the Environment.
3. Validate the "*XxCommonNotificationHandler*" is deployed in OIC.
4. Validate the " *XxParkingLotProducer(1.0*)" is deployed in OIC.
5. Validate the " *XxGetLookupVariable(1.0)"* is deployed in OIC.
6. Update the values of Instance URl, Insatnce Name in Domino Constant lookup.
7. Configure the Connection with Dev Instance URL, Instance Name (whenever updated) which is present in Domino Constant lookup maintained in OIC.
8. Complete the OIC integration configuration and activate OIC service.
9. BIP Report should be present in the path mentioned in *XXDominoCommonLookup.*

### Unit Testing

Please refer the attached file for the Unit Test cases performed for this integration.



# Open and Closed Issues

This section is used to summarise any open and closed questions relating to Error Handler

| ID | **Topic** | **Description** | **Status** | **Owner** |
| --- | --- | --- | --- | --- |
| 1 |  |  |  |  |
| 2. |  |  |  |  |



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 1212

Connect with us

[Blogger-logo-1-.png](https://blogs.oracle.com/) [blogs.oracle.com](https://blogs.oracle.com/)

[facebook_logo_white_banner.png](https://www.facebook.com/Oracle) [facebook.com/oracle](https://www.facebook.com/Oracle)

[Twitterlogo.png](https://twitter.com/oracle) [twitter.com/oracle](https://twitter.com/oracle)

[Oracle_logo-5.pngoracle.com](http://www.oracle.com/index.html)