

Oracle® Retail Supply Chain Collaboration Cloud Service Implementation Guide



G28759-01
July 2025



Oracle Retail Supply Chain Collaboration Cloud Service Implementation Guide,

G28759-01

Copyright © 2025, Oracle and/or its affiliates.

Primary Author:

Contributing Authors:

Contributors:

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

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

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Send Us Your Comments

Preface

Audience	xxvii
Documentation Accessibility	xxvii
Related Documents	xxvii
Customer Support	xxvii
Review Patch Documentation	xxviii
Improved Process for Oracle Retail Documentation Corrections	xxviii
Oracle Retail Documentation on the Oracle Help Center (docs.oracle.com)	xxviii
Conventions	xxviii

Part I Supplier Evaluation

1 Introduction

Contents of this Guide	1-1
Key Features of Supplier Evaluation Cloud Service	1-2

2 API Overview and Architecture

SOAP and RESTful APIs	2-1
Request and Response XML	2-1
Versioning	2-2
Logging	2-3
Accessing the APIs	2-3
Availability	2-3
Deployment Model	2-3
Authentication	2-3
Authorization	2-7
Listing Available APIs	2-7
REST API Path	2-9

Change History	2-9
Security Considerations	2-10
Password Management	2-10
Rich Text Data	2-10
Excluded Fields	2-10
Secured Connections	2-11
URL Encoding	2-11
Reference Fields	2-11
Glossary and Locale/Language Fields	2-11
Enumeration Fields	2-12
Editable Text Selector Fields	2-12
Key Features of the API Process	2-12

3 APIs Available in Supplier Evaluation

Individual Services	3-1
Services Grouped by Area	3-5
User and Supplier APIs	3-5
Audits API	3-6
Data Privacy API	3-6
Product APIs	3-6
Project/Process & Activity APIs	3-6
Miscellaneous	3-7
Dependencies	3-7
Linking to Related Records	3-11

4 RESTful APIs

Parameters and Filtering	4-2
Error Messages	4-4
Wildcard Searches	4-5
Deleted Records	4-5
UserRestService	4-6
List of Values	4-6
Retrieve Record by ID	4-7
Retrieve Record by Business Key	4-8
Check Record Modification Timestamp	4-9
Create Record	4-10
Update Record	4-14
SupplierRestService	4-15
List of Values	4-16
Retrieve Record by ID	4-18

Retrieve Record by Business Key	4-19
Check Record Modification Timestamp	4-20
Create Record	4-20
Update Record	4-24
SiteRestService	4-25
List of Values	4-26
Retrieve Record by ID	4-29
Retrieve Record by Business Key	4-30
Check Record Modification Timestamp	4-31
Create Record	4-32
Update Record	4-34
ContactRestService	4-35
List of Values	4-36
Retrieve Record by ID	4-38
Retrieve Supplier Contact Record by Business Key	4-39
Retrieve Site Contact Record by Business Key	4-40
Check Record Modification Timestamp	4-40
Create Record	4-41
Update Record	4-44
TaskRestService	4-45
List of Tasks	4-45
UrgentItemsRestService	4-46
Number of Urgent Items	4-46
ArtworkRestService	4-47
Started Activities	4-47
Update Record	4-49
BusinessCategoryService	4-50
List of Values	4-51
Retrieve Record by ID	4-52
Check Record Modification Timestamp	4-53
Create Record	4-54
Update Record	4-57
Delete Record	4-59
AuditRestService	4-59
List of Values (Audit)	4-60
Retrieve Record by ID (Audit)	4-61
Retrieve Record by Business Key (Audit)	4-62
Check Record Modification Timestamp (Audit)	4-63
Create Record (Audit)	4-64
Update Record (Audit)	4-68
List of Values (Checklist)	4-70
Retrieve Record by ID (Checklist)	4-71

Retrieve Record by Business Key (Checklist)	4-72
Create Record (Checklist)	4-73
Update Record (Checklist)	4-80
Delete Record (Checklist)	4-81
DataPrivacyService	4-81
Right to Access	4-82
Right to be Forgotten	4-84
AttachmentRestService	4-88
Retrieve Report Attachment	4-90
Retrieve Attachment	4-90
ScorecardRestService	4-91
List of Values	4-91
Retrieve Record by ID	4-93
Retrieve Record by Business Key	4-94
Check Record Modification Timestamp	4-95
Update Record	4-95
Update Sub Status	4-101
ProjectRestService	4-102
List of Values	4-103
Retrieve Record by ID	4-104
Retrieve Record by Business Key	4-106
Check Record Modification Timestamp	4-107
Update Record	4-107
ActivityRestService	4-111
List of Values	4-111
Retrieve Record by ID	4-113
Retrieve Record by Business Key	4-114
Check Record Modification Timestamp	4-115
Update Record	4-115
Update Status	4-117
ProductRecordRestService	4-118
List of Values	4-119
Retrieve Record by ID	4-120
Retrieve Record by Business Key	4-121
Check Record Modification Timestamp	4-122
Create Record	4-123
Update Record	4-127

5 SOAP APIs

Parameters and Filtering	5-1
Project Activities	5-2

6 Appendix: Secure Development Guide

Best Practices for Web Service Use	6-1
Store External System Credentials Securely	6-1
Configure External Systems Permissions Appropriately	6-1
Maintain Strong Passwords for External System Records	6-1
Make Use of the Web Service Log	6-2
Ensure Your Client Supports TLS v1.2 or Higher	6-2
Limit the Size of Web Service Payloads	6-2

7 Appendix: Using the Data Privacy API

Definition of Personal Data	7-2
Personal Data Flow	7-2
Right to Access Requests	7-3
Personal Data Searched and Returned	7-3
Security and Logging	7-5
Example of Using the Right to Access Service	7-5
Right to be Forgotten Requests	7-7
Personal Data Updated	7-8
Exceptions	7-9
Purging Inactive User Accounts	7-9
Security and Logging	7-9
Example of Using the Right to be Forgotten Service	7-10
Other Aspects of Data Privacy	7-13

8 Appendix: Using the Report Attachment API

9 Appendix: Accessibility

User Interface	9-1

10 Appendix: LCA Third-Party Integration

Accredited Bodies	10-1
Assessments Integration	10-2
Using the Scorecard API	10-3

Part II Supply Chain Hub

11 Overview

Getting Started	11-3
Oracle Retail Data Store (RDS)	11-3
Modules	11-3

12 Supply Chain Hub

Functionalities	12-1

13 Procure to Pay

Purchase Orders	13-1
Implementation Overview	13-1
Functionality	13-1
Status Workflow	13-1
Integration High Level	13-2
Configuration	13-3
POM Configuration	13-3
Other Cloud Services Configuration	13-3
Batches	13-3
Other Implementation Considerations	13-3
Custom Validations - Purchase Orders	13-3
Materialized Views	13-4
Codes	13-6
Templates	13-6
Interactive Grid Additional Columns	13-9
References	13-10
Operations	13-10
Integration by Modules	13-10
Reference	13-11
Materialized Views Refresh Detail	13-11
Shipments	13-17
Implementation Overview	13-17
Status Workflow	13-17
Integration High Level	13-18
Configuration	13-18
POM Configuration	13-19
Other implementation Considerations	13-19
Operations	13-23

Finance Documents	13-25
Implementation Overview	13-25
Status Workflow	13-26
Integration High Level	13-27
Configuration	13-27
POM Configuration	13-27
Other Cloud Services Configuration	13-28
Other implementation Considerations	13-28
Operations	13-30

14 Return to Vendor

Implementation Overview	14-1
Functionality	14-1
Status Workflow	14-1
Configuration	14-2
POM Configuration	14-2
Schedule Overview	14-2
Other Cloud Services Configuration	14-2
Reason Codes	14-2
Batches	14-2
Others	14-3
Other Implementation Considerations	14-3
Custom Validations	14-3
SPRV_RTV_SUBMIT	14-3
SPRV_RTV_REJECT	14-3
SPRV_RTV_APPROVE	14-3
SPRV_RTV_CANCEL	14-3
Materialized Views	14-3
ORCV_MV_SPRV_RTV_STAT_APPR	14-3
ORCV_MV_SPRV_RTV_STAT_SHPD	14-4
ORCV_MV_SPRV_RTV_BY_DEPT	14-4
ORCV_MV_SPRV_RTV_BY_LOC	14-4
ORCV_MV_SPRV_DASH_RTV_PENDNG	14-4
Codes	14-4
Templates	14-5
Template Return to Vendor	14-5
Codes	14-6
References	14-6
Operations	14-7
Integration by Module	14-7
Materialized Views Refresh Details	14-7

ORCV_MV_SPRV_RTV_STAT_APPR	14-7
ORCV_MV_SPRV_RTV_STAT_SHPD	14-8
ORCV_MV_SPRV_RTV_BY_DEPT	14-8
ORCV_MV_SPRV_RTV_BY_LOC	14-9
ORCV_MV_SPRV_DASH_RTV_PENDNG	14-10

15 Inventory and Sales

Functionalities	15-1
Configuration	15-1
POM Configuration	15-1
Schedule Overview	15-1
Other Implementation Considerations	15-2
Materialized Views	15-2
ORCV_MV_SPIS_INV_DAILYN_IL	15-2
ORCV_MV_SPIS_INV_WKLYN_IL	15-2
ORCV_MV_SPIS_INV_WKLY_DEP	15-2
ORCV_MV_SPIS_INV_WKLY_IL	15-2
References	15-2
Operations	15-3
Materialized Views Refresh Details	15-3
ORCV_MV_SPIS_INV_DAILYN_IL	15-3
ORCV_MV_SPIS_INV_WKLYN_IL	15-3
ORCV_MV_SPIS_INV_WKLY_DEP	15-4
ORCV_MV_SPIS_INV_WKLY_IL	15-5

16 Items Management

Implementation Overview	16-1
Functionality	16-1
Status Workflow	16-1
Integration High Level	16-2
Configuration	16-3
POM Configuration	16-3
Similar Items	16-3
Database Objects	16-3
Other Cloud Services Configuration	16-4
System Options	16-4
Reason Codes	16-5
Code Head	16-5
Code Detail	16-5
UDAs	16-5

Batches	16-6
Other Implementation Considerations	16-6
Notifications and Alerts	16-6
Notifications	16-6
Alerts	16-7
Custom Validations	16-7
SPIM_ITEM_APPROVE	16-7
SPIM_ITEM_SUBMIT	16-8
Custom Flex Attributes	16-8
Specific Batch Configuration	16-8
Check Status Batch	16-9
Codes	16-9
Templates	16-9
Template SPIM Item Data	16-10
Template SPIM Supplier Default	16-14
Codes	16-15
References	16-15
Operations	16-15
Integration by Module	16-16
Batch by Module	16-16
Check Item Status Batch	16-16
Alerts	16-18
Service by Module	16-18
Recalculate Defaults	16-18
Reference	16-19

17 Cost Changes Management

Implementation Overview	17-1
Status Workflow	17-1
Integration High Level	17-2
Configuration	17-3
POM Configuration	17-3
Schedule Overview	17-3
Other Cloud Services Configuration	17-3
System Options	17-3
Reason Codes	17-3
Other Implementation Considerations	17-3
Notifications and Alerts	17-4
Notifications	17-4
Alerts	17-4
Specific Batch Configuration	17-5

Cost Changes Events Pending Batch	17-5
Publish Cost Change Batch	17-5
Materialized Views	17-6
ORCV_MV_SPCC_DASH_TOP_DEPT	17-6
ORCV_MV_SPCC_DASH_SUBCHG	17-6
Codes	17-6
Custom Validations	17-6
Templates	17-7
Template SPCC Cost Change	17-7
Interactive Grid Additional Columns	17-8
References	17-8
Operations	17-8
Integration by Module	17-9
Batch by Module	17-9
Cost Changes Events Pending Batch	17-9
Publish Cost Changes	17-9
Materialized Views Refresh Details	17-10
ORCV_MV_SPCC_DASH_SUBCHG	17-10
ORCV_MV_SPCC_DASH_TOP_DEPT	17-11

18 Supplier Performance

Implementation Overview	18-1
Functionality	18-1
Configuration	18-1
POM Configuration	18-1
Schedule Overview	18-2
Other Implementation Considerations	18-2
Specific Batch Configuration	18-2
On Time Delivery	18-2
Quantity Compliance	18-3
Invoice Compliance	18-3
Return Percentage	18-4
Defect Percentage	18-4
Overall Compliance	18-5
Materialized Views	18-5
Codes	18-6
Templates	18-6
Template SPPF Questionnaire	18-6
Template SPPF Questionnaire Reply	18-7
Operations	18-7
Batch by Module	18-7

On Time Delivery	18-7
Quantity Compliance	18-9
Invoice Compliance	18-10
Return Percentage	18-11
Defect Percentage	18-12
Overall Compliance	18-13
Materialized Views Refresh Details	18-15
ORCV_MV_SPPF_SUP_T_ORD	18-15
ORCV_MV_SPPF_SUP_ORDERS	18-15

19 Quality Control

Implementation Overview	19-1
Functionality	19-1
Status Workflow	19-1
Integration High Level	19-2
Configuration	19-2
POM Configuration	19-2
Other Implementation Considerations	19-3
Notifications and Alerts	19-3
Notifications	19-3
Alerts	19-3
Custom Validations	19-4
QLCT_JOB_CREATE	19-4
QLCT_JOB_MODIFY	19-4
QLCT_JOB_SUBMIT	19-4
QLCT_JOB_CANCEL	19-5
QLCT_JOB_COMPLETE	19-5
QLCT_JOB_RELEASE_ITEM	19-5
Specific Batch Configuration	19-6
PO ASN Received Batch	19-6
Old Stock Batch	19-7
Codes	19-7
Templates	19-8
Template QLCT Job	19-8
Codes	19-9
Interactive Grid Additional Columns	19-9
References	19-10
Operations	19-10
Integration by Module	19-10
Batch by Module	19-11
PO ASN Received Batch	19-11

20 Deals Management

Implementation Overview	20-1
Status Workflow	20-1
Integration High Level	20-2
Configuration	20-3
POM Configuration	20-3
Schedule Overview	20-3
Other Cloud Services Configuration	20-3
System Options	20-3
Batches	20-4
Other Implementation Considerations	20-4
Custom Validations	20-4
SPDM_DEAL_DRAFT	20-4
SPDM_DEAL_SUBMIT	20-5
SPDM_DEAL_APPROVE	20-5
SPDM_DEAL_REJECT	20-6
SPDM_DEAL_CANCEL	20-6
Specific Batch Configuration	20-7
Publish Deal Batch	20-7
Codes	20-7
Templates	20-8
Template Deal	20-8
Codes	20-10
References	20-11
Operations	20-11
Integration by Module	20-11
Batch by Module	20-11
Publish Deal Batch	20-12

21 Supplier Penalties

Implementation Overview	21-1
Status Workflow	21-1
Integration High Level	21-2
Configuration	21-2
POM Configuration	21-2
Schedule Overview	21-3
Other Cloud Services Configuration	21-3
System Options	21-3

Batches	21-3
Other Implementation Considerations	21-3
Notifications and Alerts	21-3
Notifications	21-4
Alerts	21-4
Custom Validations	21-4
SPPY_PENALTY_CREATE	21-5
SPPY_PENALTY_APPROVE	21-5
Specific Batch Configuration	21-5
Early Late Shipment Batch	21-5
QA Defect Batch	21-6
Cancel PO Batch	21-7
Penalties Update IMCS IDs	21-8
Penalties Events Changes Batch	21-8
Rule Attributes	21-8
Early Late Shipment Rule	21-8
QA Defect Rule	21-9
Cancel PO Rule	21-9
Codes	21-10
References	21-10
Operations	21-10
Integration by Module	21-11
Batch by Module	21-11
Early Late Shipment Batch	21-11
QA Defect Batch	21-11
Cancel PO Batch	21-11

22 Retail Price

Implementation Overview	22-1
Status Workflow	22-1
Integration High Level	22-3
Configuration	22-3
POM Configuration	22-3
Promotion and Offer Aggregator	22-3
Promotion Aggregator	22-4
Offer Aggregator	22-4
Other Cloud Services Configuration	22-5
System Options	22-5
Reason Codes	22-5
Other Implementation Considerations	22-5
Notifications and Alerts	22-5

Notifications	22-5
Alerts	22-6
Custom Validations	22-6
SPPM_PRC_SUBMIT	22-6
SPPM_PRC_APPROVE	22-7
SPPM_PRC_REJECT	22-7
SPPM_PRC_CHANGE_PROPOSAL	22-8
SPPM_PRC_PUBLISH	22-8
SPPM_PRC_EDIT	22-8
SPPM_PRC_SUB_CHANGES	22-8
SPPM_SPROMO_SUBMIT	22-9
SPPM_SPROMO_APPROVE	22-9
SPPM_SPROMO_REJECT	22-10
SPPM_SPROMO_CANCEL	22-10
SPPM_SPROMO_CHANGE_PROPOSAL	22-10
SPPM_SPROMO_PUBLISH	22-11
SPPM_SPROMO_EDIT	22-11
SPPM_SPROMO_SUB_CHANGES	22-11
Specific Batch Configuration	22-11
Publish Price Event Batch	22-11
Update Price Event Status Batch	22-12
Price Event Emergency Alert Batch	22-12
Codes	22-13
Templates	22-14
Template Price Events	22-14
Template Simplified Price Event Creation	22-15
Codes	22-16
References	22-16
Operations	22-16
Integration by Module	22-17
Batch by Module	22-17
Publish Price Event Batch	22-17
Update Price Event Batch	22-19
Price Event Emergency Alert Batch	22-20

23 Financial Summary

Implementation Overview	23-1
Integration High Level	23-1
Configuration	23-1
POM Configuration	23-1
Schedule Overview	23-1

Other Cloud Services Configuration	23-2
Others	23-2
Other Implementation Considerations	23-2
Materialized Views	23-2
ORCV_MV_SPFS_PAYMENT_BLNC	23-2
Codes	23-2
References	23-3
Operations	23-3
Integration by Module	23-3
Materialized Views Refresh Details	23-3
Service by Module	23-4
Payments Upload Service	23-4

24 Appointments Management

Implementation Overview	24-1
Functionality	24-1
Status Workflow	24-1
Configuration	24-2
POM Configuration	24-2
Schedule Overview	24-2
Other Implementation Considerations	24-2
Notifications and Alerts	24-2
Notifications	24-2
Alerts	24-3
Custom Validations	24-3
Specific Batch Configuration	24-4
Materialized Views	24-4
ORCV_MV_SPAP_APPT_EST_QTY	24-5
Codes	24-5
Templates	24-5
Template Appointments Data	24-5
Template Slots Data	24-6
Operations	24-7
Batch by Module	24-7
Appointments Pending Action Alert Batch	24-7
Materialized Views Refresh Details	24-7
ORCV_MV_SPAP_APPT_EST_QTY	24-7

25 Customer Order Fulfillment

Implementation Overview	25-1
Functionality	25-1
Status Workflow	25-1
Integration High Level	25-3
Configuration	25-3
POM Configuration	25-4
Schedule Overview	25-4
Other Cloud Services Configuration	25-4
System Options	25-4
Other Implementation Considerations	25-4
Notifications and Alerts	25-4
Notifications	25-5
Alerts	25-5
Custom Validations	25-5
Specific Batch Configuration	25-6
MFCS Control Tables Update Batch	25-6
Shipments and Finance Documents Publish Batch	25-6
Customer Order Status Update Batch	25-7
Events Notifications and Alerts Batch	25-7
Materialized Views	25-8
ORCV_MV_SPCO_DASH_COF_BY_DEP	25-8
ORCV_MV_SPCO_DASH_CMPL_COF	25-8
ORCV_MV_SPCO_DASH_MISG_INV	25-8
Codes	25-9
Templates	25-9
Template Customer Order Fulfillment Data	25-9
Codes	25-10
References	25-10
Operations	25-10
Integration by Module	25-10
Services by Module	25-11
Fulfillment Order Service	25-11
Materialized Views Refresh Details	25-19

26 Reports

Implementation Overview	26-1
Functionality	26-1
Integration High Level	26-1
Configuration	26-1

Report Configuration	26-1
Database Objects	26-1
Configuration Steps	26-3
Operations	26-4
Integration by Module	26-4

27 Supply Chain Hub Frameworks

Getting Started	27-1
ReST Web Services Endpoints	27-1
Configure your REST client	27-1
Batch Framework	27-2
Batch Engine Components	27-2
Configuration	27-2
Pre Processor	27-2
Processor	27-3
Post Processor	27-3
Retry Mechanism	27-3
Configuration	27-3
Technical Batch Recovery Actions	27-3
Batch Dependency Framework	27-4
Batch Flow Diagram	27-4
Main Database Functions	27-4
System Parameters	27-5
Batch Dependency Configuration Example	27-6
Custom Validations Framework	27-7
Process Overview	27-7
Database Objects	27-8
Configuration	27-8
Custom Flex Attributes	27-9
Database Objects	27-9
Tables	27-9
Packages	27-9
Process Overview	27-10
Configuration	27-10
Other implementation Considerations	27-12
Custom Validations	27-12
CFA Group View Generator Batch	27-12
Rules Engine Framework	27-14
Process Overview	27-14
Database Objects	27-14
Configuration	27-15

Web Service	27-16
Profile	27-17
Implementation Overview	27-17
Functionality	27-17
Configuration	27-18
Security	27-18
Application Configuration	27-19
POM Configuration	27-19
Other Implementation Considerations	27-20
Specific Batch Configuration	27-20
Codes	27-20
Operations	27-21
Batch by Module	27-21
Configuration Management	27-22
Inventory	27-23
Functionalities	27-23
Configuration	27-23
POM Configuration	27-23
Other Implementation Considerations	27-24
Materialized Views	27-24
Specific Batch Configuration	27-25
Operations	27-26
Materialized Views Refresh Details	27-26
Batch by Module	27-27
Queries	27-30
Functionalities	27-30
Configuration	27-30
Application Options	27-30
Security	27-30
System Parameters	27-30
Codes	27-31
Batch WebServices	27-31
Upload and Download Framework	27-32
Functionalities	27-32
Architecture	27-33
Process Overview	27-34
Upload	27-34
Download	27-36
ORDS Download Processing Flow	27-37
Web Service	27-38
ORDS	27-38
Materialized View Framework	27-40

Process Overview	27-40
Batch Flow	27-40
Run	27-41
Pre Processor	27-41
Processor	27-42
Post Processor	27-42
Refresh Materialize View Web Service	27-42
Database Objects	27-43
Configurations	27-44
Purge Framework	27-44
Overview	27-44
Batch Flow	27-45
Configuration	27-45
Purge Web Service	27-45
Events Framework	27-46
Process Overview	27-47
Events Process Flow	27-48
Configuration	27-49
Configuration Tables	27-49
System Parameters	27-53
Codes	27-54
Batch	27-55
Web Service	27-55
POM (Process Orchestration and Monitoring)	27-56
Batch Schedule Download	27-56
Download Through Upload Download Application UI	27-56
Download through Web Service	27-56
Batch Schedule Configuration Download	27-57
Download Through Upload Download Application UI	27-57
Download through Web Service	27-57
Callback Service	27-58
Web Service	27-58
Aggregator Framework	27-59
Process Overview	27-60
Database Objects	27-60
Tables	27-60
Packages	27-61
Configuration	27-63
View Join Framework	27-63
Process Overview	27-64
Database Objects	27-64
Tables	27-64

Packages	27-65
Views	27-66
Configuration	27-68
Database Integration Framework	27-71
Publish	27-71
Process Overview	27-71
Configuration	27-75
Interactive Grid Additional Columns	27-79
Database Objects	27-80
Tables	27-80
Views	27-80
Dynamic Execution Framework	27-80
Process Overview	27-81
Dynamic Button Rendering Overview	27-81
Extended Application Flow Helpers	27-82
Database Objects	27-84
Tables	27-84
Views	27-84
Sequences	27-85
Types	27-85
Packages	27-85
Configuration Example - Cost Changes Management	27-85
ORCV_CODE_DETAIL	27-87
ORCV_SEC_GROUPS	27-87
ORCV_DE_GROUP_HEAD	27-87
ORCV_DE_ENTITY_STEPS	27-88
ORCV_DE_GROUP_DETAIL	27-88
ORCV_DE_GROUP_DETAIL_CUSTOM	27-88
ORCV_DE_GROUP_DTL_EXE_COND	27-88
ORCV_DE_GROUP_USER_GROUPS	27-89
ORCV_DE_ELEMENT_RENDER	27-89
ORCV_DE_ELEMENT_RENDER_FILTER	27-90
ORCV_DE_ELEMENT_RND_FILTER_DTL	27-90
Integration	27-91
Price Change Integration to RPCS	27-91
Process Overview	27-91
Process Price Changes	27-93
Pre Processor	27-94
Processor	27-95
Set Integration Status	27-96
Configuration	27-96
System Parameters	27-96

Web Service	27-97
Promotion Induction to RPCS	27-98
Process Overview	27-98
Process Promotion	27-99
Pre Processor	27-100
Processor	27-101
Post Processor	27-102
Set Integration Status	27-102
Configuration	27-103
System Parameters	27-103
Web Service	27-103
Clearance Induction to RPCS	27-105
Process Overview	27-105
Process Clearance	27-107
Pre Processor	27-108
Processor	27-109
Set Integration Status	27-110
Configuration	27-110
System Parameters	27-110
Web Service	27-111
Clearance XREF Batch Service	27-112
Purchase Order Induction to MFCS	27-114
Process Overview	27-114
Process Purchase Order	27-116
Pre Processor	27-117
Processor	27-118
Post Processor	27-119
Set Integration Status	27-120
Batch Flow	27-121
Configuration	27-121
Web Service	27-122
Shipment ASNIN Integration to RICS	27-123
Process Overview	27-124
Process Shipment ASNIN	27-125
Pre Processor	27-126
Processor	27-127
Post Processor	27-128
Set Integration Status	27-129
ASNIN Targets Configuration	27-129
Configuration	27-131
Web Service	27-131
Finance EDI Documents to IMCS	27-133

Process Overview	27-134
System Parameters	27-135
Process EDI	27-136
Pre Processor	27-137
Processor	27-138
Post Processor	27-139
Set Integration Status	27-140
Configuration	27-140
Batch Flow	27-141
Web Service	27-141
Finance EDI Document XREF Service	27-144
Item Induction to MFCS	27-145
Process Overview	27-146
Process Item	27-147
Pre-Processor	27-148
Processor	27-149
Set Integration Status	27-150
Batch Flow	27-151
Configuration	27-152
System Parameters	27-152
Web Service	27-152
Item CFAs Rest to MFCS	27-158
Process Overview	27-158
Configuration	27-161
Item Rest Publish Asynchronous Batch	27-163
Item Location Integration to MFCS	27-166
Process Overview	27-166
Process Item Loc	27-168
Pre-Processor	27-169
Processor	27-170
Set Integration Status	27-171
Batch Flow	27-172
Configuration	27-172
Web Service	27-173
Cost Change Induction to MFCS	27-174
Process Overview	27-174
Process Cost Change	27-176
Pre-Processor	27-177
Processor	27-178
Set Integration Status	27-179
Batch Flow	27-180
Configuration	27-181

System Parameters	27-181
Web Service	27-181
Return to Vendor to MFCS	27-183
Process Overview	27-183
Configuration	27-186
Deals to MFCS	27-187
Process Overview	27-188
Process Deal	27-189
Pre-Processor	27-190
Processor	27-191
Post Processor	27-192
Set Integration Status	27-192
Configuration	27-193
System Parameters	27-193
Web Service	27-194
Deal XREF Batch Service	27-195
Fulfillment Order to MFCS	27-197
Process Overview	27-198
Configuration	27-202
Security	27-202
Authentication and Single Sign-On	27-202
ReST Services Authentication	27-203
Database	27-203
Data Filter Policy	27-203
GDPR	27-203
Web Services	27-203

Glossary

Send Us Your Comments

Oracle® Retail Supply Chain Collaboration Cloud Service Implementation Guide, Release 25.1.301.0

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

**Note:**

Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at <http://www.oracle.com>.

Preface

The *Oracle® Retail Supply Chain Collaboration Cloud Service Implementation Guide* describes all available functions of Release 25.1.301.0.

Audience

This guide is for technical personnel who configure, maintain and support, or use Oracle Retail Supply Chain Collaboration Cloud Service.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Retail Supply Chain Collaboration Cloud Service documentation library at the following URL:

<https://docs.oracle.com/en/industries/retail/index.html>

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

<https://support.oracle.com>

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 25.0) or a later patch release (for example, 25.0.1). If you are installing the base release or additional patches, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Help Center (docs.oracle.com) Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Oracle Help Center (docs.oracle.com) at the following URL:

<https://docs.oracle.com/en/industries/retail/index.html>

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

Oracle Retail Documentation on the Oracle Help Center (docs.oracle.com)

Oracle Retail product documentation is available on the following web site:

<https://docs.oracle.com/en/industries/retail/index.html>

(Data Model documents can be obtained through My Oracle Support.)

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Supplier Evaluation

This part of the *Oracle® Retail Supply Chain Collaboration Cloud Service Implementation Guide* contains information about Supplier Evaluation.

- [Introduction](#)
- [API Overview and Architecture](#)[APIs Available in Supplier Evaluation](#)
- [RESTful APIs](#)
- [SOAP APIs](#)
- [Appendix: Secure Development Guide](#)
- [Appendix: Using the Data Privacy API](#)
- [Appendix: Using the Report Attachment API](#)
- [Appendix: Accessibility](#)
- [Appendix: LCA Third-Party Integration](#)

1

Introduction

Oracle Retail Supplier Evaluation Cloud Service is a collaborative cloud service for the onboarding and evaluation of merchandising suppliers, enabling the assessment and governance of ethical, environmental, safety, and quality performance. It manages the selection of suppliers against Environmental, Social, and Governance (ESG), brand standards and governance policies, incorporating supplier self-certification survey and assessment, audit and action management, vendor performance, and incident alert notifications.

Supplier Evaluation includes a number of exposed Application Programming Interfaces (APIs) that can be called by external systems. This document details all of the exposed APIs and the information required in order to access them.

An API is a way of exchanging data between computer systems. The owners of an external system can interact with Supplier Evaluation by developing the necessary functionality to trigger the web service submissions, and handle the returned messages from Supplier Evaluation.

In order to access Supplier Evaluation using its APIs, the portal owner must grant the external system access. This is achieved by creating an External Systems account in the Admin area and assigning the unique access credentials in the form of a login ID and password.



Note:

The terminology used for Scorecard and Project has changed. Scorecard has been renamed to Assessment. Project has been renamed to Process. The associated API names and endpoints/functions are not changed. For Assessments, see the ScorecardRestAPI. For Processes, see the ProjectRestAPI.

Contents of this Guide

This implementation guide addresses the following topics:

- [API Overview and Architecture](#): Overview of the Supplier Evaluation APIs and architecture used.
- [APIs Available in Supplier Evaluation](#): Summary of the APIs, including the release in which the API is available and the type of API.
- [RESTful APIs](#): Details of the available APIs that are based on the RESTful architecture.
- [SOAP APIs](#): Details of the available APIs that are based on the SOAP architecture.
- [Appendix: Secure Development Guide](#): Description of some best practices for web service use.
- [Appendix: Using the Data Privacy API](#): Additional detail on using the Data Privacy API, including examples.
- [Appendix: Using the Report Attachment API](#): Details on using the Report Attachment API.

Key Features of Supplier Evaluation Cloud Service

The application is composed of the following modules:

- Library enables the issue, receipt, and acceptance of policies, guidelines, and key working documents.
- Process supports the development of process briefs, plans, and workflow management.
- Supplier enables the identification, selection, and approval of suppliers.
- Reports provides a reporting tool for reporting across the system, using standard templates and custom reports.
- Product supports the development of products.

2

API Overview and Architecture

The APIs exposed by Supplier Evaluation are implemented in one of two formats, SOAP web services or RESTful web services. In both cases, requests and responses are normally in an XML format.

The APIs are already built and available for clients Supplier Evaluation to build services for their external applications to utilize. The external system makes a call to Supplier Evaluation using HTTPS. If the external system is validated (that is, it is enabled in the Admin area of Supplier Evaluation and the login user ID and password match), the call is accepted.

Calls will typically allow for the retrieval, creation, update, and in some cases deletion, of data through the API. Each call is logged in the Web Service Log in Supplier Evaluation, with the request and response messages being recorded as appropriate.

A call may contain a number of parameters or filters to determine the records to be returned or updated, or to identify where to create a new record, such as to filter Audit of a certain type, status and so on.

SOAP and RESTful APIs

The Supplier Evaluation APIs are web services in both SOAP and RESTful architectural styles. In general, the SOAP services are developed for specific requirements or are historical services. Unless there is a definite requirement for a SOAP implementation of a service, any new services or new versions of services are implemented using the RESTful architectural style.

Request and Response XML

In general, APIs exposed by Supplier Evaluation as web services use XML messages for both requests and responses. Exceptions to this are RESTful services where the use of an XML request is replaced by the use of URI parameters in a GET call. The APIs do not support other message formats, such as JSON.

The XML data schemas are written to allow for both forwards and backwards compatibility. All elements within the XML schemas are marked as optional. The element names are ordered so that new fields are added after existing elements and, within each sequence of elements, the last element is always an *any* element so that fields added in future versions are ignored by clients that are using an old version of the schema. For example:

```
<xs:complexType name="fpsMicroTestLocaleDataFullDTO">
  <xs:sequence>
    <xs:element form="qualified" minOccurs="0" name="description" type="xs:string"/>
    <xs:element form="qualified" minOccurs="0" name="locale" type="xs:string"/>
    <xs:element form="qualified" minOccurs="0" name="id" type="xs:long"/>
    <xs:element form="qualified" minOccurs="0" name="createdOn" type="xs:string"/>
    <xs:element form="qualified" minOccurs="0" name="updatedOn" type="xs:string"/>
    <xs:any maxOccurs="unbounded" minOccurs="0" namespace="##other"
processContents="lax"/>
  </xs:sequence>
</xs:complexType>
```

The element names in the XML Schemas are defined by the core system and do not change in specific implementations. If an implementation has additional fields that have been added to the system (the implementation is non-core), those fields may be exposed after the core fields (before the `xs:any` element) and will incur a maintenance overhead in later versions to cope with any additional core fields added after the custom fields.

Data types in the XML Schemas are the standard datatypes for XSDs, WSDLs, and WADLs. Values should be entered as described on the following web site:

<http://www.w3.org/TR/xmlschema-2/>

The following table lists the common data types used in the Supplier Evaluation schemas:

Data Types

Type	Example
<code>xs:string</code>	A normal string
<code>xs:string</code>	<code><![CDATA[A String with
 html data]]></code>
<code>xs:long</code>	9223372036854775807
<code>xs:int</code>	2147483647
<code>xs:short</code>	-32768
<code>xs:boolean</code>	true or false
<code>xs:boolean</code>	1 or 0
<code>xs:date</code>	2015-01-30
<code>xs:dateTime</code>	2015-01-30T23:59:59

Date Time Parsing

For date time strings in inbound data, the preferred format is `YYYY-MM-DDThh:mm:ss.fffZ`, however all the following formats are permitted:

```
YYYY-MM-DD hh:mm:ss
YYYY-MM-DDThh:mm:ss
YYYY-MM-DDThh:mm:ss.fff
YYYY-MM-DDThh:mm:ss.fffZ
```

The format of date time strings in outbound data is `YYYY-MM-DDThh:mm:ss.fffZ`.

Versioning

Due to the format of the XML schemas as described above, most revisions of the web services should be forwards and backwards compatible to allow clients of those web services to be upgraded and developed without binding the lifecycle of the client services to the upgrade cycle of Supplier Evaluation.

Where the necessary changes to a service cannot be made compatible with previous versions, a new endpoint will be created to expose the changed service. If possible, the previous version of the service will remain available at the original endpoint to enable existing and legacy clients to continue to work, but will be noted as deprecated in documentation to encourage developers of clients to the services to avoid using it.

Logging

All calls to APIs exposed as web services are logged within the application and can be viewed by logging into Supplier Evaluation as a Power User, navigating to Company > Admin> Notifications, and choosing the Web Service Log option. For each call, the name of the web service, date and time of the call, status (in progress/completed/failed), duration of the call (not available for In Progress calls), and a list of error messages if the call failed are shown.

The request and response XMLs are attached to each web service log entry. If a request does not contain XML, no attachment is created.

Accessing the APIs

This section describes how to access the APIs.

Availability

If enabled within the portal, an API is accessible while the portal is online. The support for a portal's availability is subject to the client's service level agreement.

Updates to the APIs are applied as per changes and fixes to the application. Each release involves a scheduled period of downtime. Details of the change and impact are covered in the supporting Release Notes.



Note:

After a Supplier Evaluation upgrade release, it may be necessary to change the URLs that the external system uses to call the APIs.

Deployment Model

The APIs, made available by Supplier Evaluation, are all web services implemented using either a SOAP or RESTful architecture.

The web services are exposed over HTTPS and require OAuth 2.0 authentication in order to access the service requested.

Authentication

OAuth 2.0 is the industry standard protocol for authorization. Supplier Evaluation REST Services now support OAuth 2.0. To invoke these services, you need to obtain an access token and use it as a bearer token.



Note:

Basic Authentication access is still available for these services, but this has now been deprecated. See the deprecation notice on My Oracle Support at [Doc ID 2921721.1](#).

Steps to Enable

To obtain a token and call the services, use the following steps:

1. Set up an External System record within Supplier Evaluation and allocate only the required Service Access and/or End Point Access that this External System requires.

To see the list of existing External System records, log in to Supplier Evaluation as a Power User, navigate to Company > Admin > Roles and Permissions, and select the External Systems option:

- a. To create a new External System: Select the Actions > New action to open up a tab for creating an External System.
- b. Enter a Login ID (User Name), Email Address, Comment (optional), and select the appropriate Service Access and/or End Point Access that is required, then select the Actions > Save & Exit option.
- c. An Oracle Identity Cloud Service (IDCS) or Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) profile will be automatically created and a notification email sent to the specified email address, containing instructions for setting the password.

The passwords used for External Systems are validated against the same criteria as the passwords for users of Supplier Evaluation. It is likely that a minimum length, mixed case, and digits will be required for the password, depending on the setup of the IDCS or OCI IAM tenant.

2. These steps should be performed one time for each External System record created and requiring access to Supplier Evaluation through OAuth 2.0. This can be done directly from your Retail Home instance.

Perform these steps using the Retail Home feature:

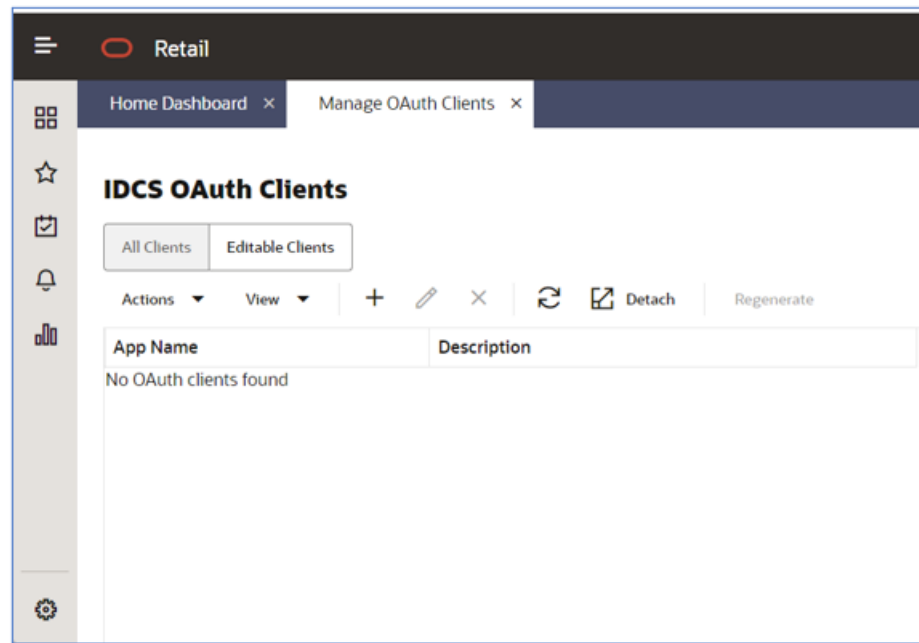
- a. Access the Retail Home associated with the cloud service as a user with Administration rights to Retail Home.

See the OAuth Application Configuration section in the *Oracle Retail Home Administration Guide*. Retail Home documentation can be accessed from the OHC Retail Documentation page:

<https://docs.oracle.com/en/industries/retail/index.html>

- b. From the Retail Home Dashboard, choose the Settings option in the bottom left of the screen. This opens a menu. Select the Application Administration option and then the Manage OAuth Clients option.

The IDCS OAuth Clients page is displayed.



- c. Choose the ADD action to create a new OAuth Client. A dialog window is displayed ready for completion.

App Name: This should be populated with the External System account name that was specified within Supplier Evaluation.

Description: A description for the use of the OAuth client.

Scope 1: For the purposes of Supplier Evaluation, this should be populated with the following:

- Stage environments: rgbu:secs:STG1
- Production environments: rgbu:secs:PRD1

- ANPE environments: rgu:secs:DEV1
- d. Complete the fields and click **OK**. The fields will be validated, and if all the entries are valid, a dialog window is shown displaying the Client ID and Client Secret for the created OAuth Client.

New IDCS OAuth 2.0 Client

Display Name: ORACLE_OAUTH_EXAMPLE

Client ID: [Redacted] [Copy]

Client Secret: [Redacted] [Copy]

Done

- e. Make note of the Client ID and Secret for use in the following steps for using the OAuth client that has been created.
- Note:** The Client ID and Secret can be retrieved from the IDCS admin console at any time.
3. Once the setup steps are completed, accessing an API requires that a Bearer token is requested from IDCS, and then this token is used on subsequent calls to Supplier Evaluation rather than sending a Username/Password for each request per Basic Authentication. To obtain the Bearer token, call IDCS with the appropriate details.

This example uses CURL to obtain the Bearer token:

- a. To obtain a Bearer token, an initial call to IDCS is required. Several values will need to be supplied all of which have been mentioned above in the setup of the Application record.

{SCOPE} = rgu:secs:STG1 / rgu:secs:PRD1 **Note:** The Scope should be URL encoded to be this format... rgu%3Asecs%3ASTG1 / rgu%3Asecs%3APRD1

{IDCS_CLIENTID} = The IDCS client id that was provided during the Application setup in IDCS.

{IDCS_SECRET} = The IDCS secret that was provided during the Application setup in IDCS.

{IDCS_URL} = The IDCS URL for the tenant being accessed, including the "https://".

```
curl -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -d
"grant_type=client_credentials&scope={SCOPE}" --user {IDCS_CLIENTID}:
{IDCS_SECRET} {IDCS_URL}/oauth2/v1/token
```

Note: When substituting the values, ensure that the {} are also replaced.

- b. Once the CURL command has been constructed, this can be executed through a Command Prompt interface. This will then return a Bearer token for use when calling the actual API.

```
{"access_token":"{token}","token_type":"Bearer","expires_in":3600}
```

4. Once the Bearer token is available, this can then be used to call the appropriate API. This example uses CURL to call the User API.

To call the appropriate API, several values will need to be supplied.

{APPLICATION_URL} = The Supplier Evaluation url for the application instance.

{API_NAME} = The Service being executed, and any parameters if required.

```
curl --url "{APPLICATION_URL}/secs/services/rest/{API_NAME}" --header  
"Authorization: Bearer {token}"
```

 Note:

IDCS or OCI IAM is used to authenticate access to the services. Calling a service using a web browser does not generate the necessary header for IDCS or OCI IAM authentication, thus resulting in a 401 or 407 authorization error. Instead, a web service tool should be used.

A web service tool such as PostMan, SoapUI, or a lower-level tool such as curl or wget must be used instead.

Authorization

Authorization is controlled by providing the Service and Endpoint access that is specified on the External System record. If no Service or Endpoint access is specified, then no access will be granted to any API.

These two Service and Endpoint access fields allow the External System to be restricted to specific APIs and to specific access methods on the APIs (such as READ, WRITE, and UPDATE).

When creating an External System, attention should be given to what the purpose is for the usage of the External System and only provide the necessary access to information that is required.

Listing Available APIs

In all installations of Supplier Evaluation, a full list of the deployed web services can be found at /services.

For example, for a system where the application is deployed to URL <https://www.example.com/supplierEvaluation>, you can retrieve a list of all services from <https://www.example.com/supplierEvaluation/services>.

 Note:

In order to access the /services page, the user must first be logged in/authenticated with the Supplier Evaluation portal.

To download the REST API WADL files:

1. Run the Windows *Command Prompt* app.
2. Retrieve a Bearer token. See the [Authorization](#) section.
3. Enter the command

```
curl [url]/services/rest/[api]?_wadl=true --header "Authorization: Bearer [token]" > [api].wadl
```

Where [url] is the portal address; [api] is the name of the REST API; [token] is the Bearer token retrieved. For example:

```
curl https://<hostname>/services/rest/user?_wadl=true --header "Authorization: Bearer [token]" > user.wadl
```

The file will be downloaded to your default directory. To download to a specific directory, provide the full path for the .wadl file. For example:

```
... > c:\Users\JSmith\Downloads\APIs\23.0.101.0\user.wadl
```

The file is saved to the specified location.

4. Import the WADL file into SoapUI.

 Note:

If using SoapUI and the HTTP 401 error is received, click the Auth tab on the request in SoapUI and add OAuth 2.0 authorization. Enter the appropriate details to request a Bearer token.

To download the WSDL files for SOAP APIs:

1. Run the Windows *Command Prompt* app.
2. Enter the command

```
curl [url]/services/[api]?_wsdl --header "Authorization: Bearer [token]" > [api].wsdl
```

Where [url] is the portal address; [api] is the name of the SOAP API; [token] is the Bearer token retrieved. For example:

```
curl https://<hostname>/services/activityManagement?wsdl --header "Authorization: Bearer [token]" > activityManagement.wsdl
```

The file will be downloaded to your default directory. To download to a specific directory, provide the full path for the .wsdl file. For example:

```
... > c:\Users\JSmith\Downloads\APIs\23.0.101.0\activityManagement.wsdl
```

The file is saved to the specified location.

 Note:

If using SoapUI and the HTTP 401 error is received, click the Auth tab on the request in SoapUI and add OAuth 2.0 authorization. Enter the appropriate details to request a Bearer token.

REST API Path

A REST API Path system parameter allows for an alternative path to the REST APIs to be specified. This is of use where for example an alternative data center is being used, or where HTTP is being used instead of HTTPS.

By default, the path of the Base URL system parameter is used for the application and the APIs. The Rest API Path system parameter is only used if there is a need to override the base URL.

Change History

Change History Log

Each record contains a Change History log, which is maintained by the system as a result of the user creating or updating the record in the UI. The log is also maintained by the system if the record is created or updated using the API.

The log comprises:

- Changed On - the date and time the record was last updated.
- Changed By - the name and email address of the user who last updated the record (or 'System' if updated by the API).
- No. Changed Fields - the number of fields that were changed by the update.

Opening an entry shows the details of the individual field changes:

- Field label
- From (before) value
- To (after) value

The log is not included in the API's XML payload so it cannot be retrieved or updated by the API.

Status Change History Log

Records that have a workflow status also have a Status Change History log - a table that has a row added by the system each time an update to the record results in a status change.

The table comprises the columns:

- Status Before - the record's status before the update.
- Status After - the record's status after the update.
- Changed On - the date and time of the update.
- Changed By - the name of the user who made the update (or 'System' if updated by the API).
- Comments - any comments entered as part of the status change action.

The log is included in the API's XML payload for retrieval, within the `statusHistory` element:

```
<statusHistory>
  <comments>
  <localeData>
    <locale>
```

```
<statusFrom>
<statusTo>
<id>
<id>
<createdOn>
<updatedOn>
<statusChangedBy>
  <code>
  <disabled>
  <externalAuthenticationUser>
  <timeZone>
  <userType>
  <id>
```

For records with a Status Change History log:

- The APIs currently allow for rows to be added, removed, and updated in the table.
- The update process is locked down to only allow the Comments column to be updated.
- The Changed By column can be populated with a valid user's name with the `statusChangedBy` element.
- If no data is supplied for `statusChangedBy`, the Changed By field in the UI will default to 'System'. In this case, the underlying value is null, so when retrieving that data using the API, the `statusChangedBy` element will be omitted (this is per the standard rule for omitting elements that have no value).

Security Considerations

This section provides information on aspects of security you need to consider.

Password Management

Unlike interactive logins for users, the use of web services does not permit warnings to be included in the returned messages. Also, there is no guarantee that there is someone handling the call in order to interpret any warnings about password expiry.

Password expiry is managed within IDCS and is the responsibility of the owner to be aware of when passwords need to be rotated. With OAuth, the client id and secret should be kept secure.

Rich Text Data

In Supplier Evaluation, certain fields allow the display of HTML-formatted data. Such data is cleansed before display in Supplier Evaluation to ensure that any attempts at Cross-Site Scripting (XSS) attacks are mitigated, but this cleansing is not applied within the APIs. Therefore, it is advisable to cleanse any HTML-formatted data before it is displayed to users.

Excluded Fields

Many of the APIs published by Supplier Evaluation provide access to manage all of the records of a given type. With certain records, the fields accessible through the API are a subset of the fields within Supplier Evaluation. Any updates leave those fields unchanged.

Secured Connections

As all of the APIs are exposed as web services over HTTPS, it is necessary for callers to communicate over SSL/TLS using the ciphers that the Supplier Evaluation servers accept. To ensure optimal security, it is advised that standard certificate checks are in place and not disabled. In the case of Apache CXF, the Client TLS Parameter `disableCNCheck` should be left as false.

URL Encoding

Certain characters within URLs are reserved for a special meaning. In order to avoid calls that contain these characters being blocked by the API's security, the characters must be encoded and passed as plain text.

The reserved characters include ";", "/", "?", ":", "@", and "=", all of which could be present in the parameters passed to Supplier Evaluation APIs - such as within email addresses or within the names of users or suppliers.

UTF-8 URL encoding should therefore be applied to all calls to the Supplier Evaluation APIs. The APIs will automatically decode the values on receipt.

For example, an un-encoded call of: `.../services/rest/user/byKey/john.smith@example.com` should be encoded as: `.../services/rest/user/byKey/john.smith%40example.com`

The use of forward slash in parameters should be encoded as "%2f", for example an un-encoded call of: `.../services/rest/user/byKey/24/7 Support`

should be encoded as: `.../services/rest/user/byKey/24%2f7 Support`

If a call comprises multiple values, each must be encoded individually.

Reference Fields

This section describes how data is returned for fields that have their values derived by referencing a glossary or system text, such as the selections made from a configurable list of values.

Glossary and Locale/Language Fields

Fields that have their values selected from a glossary of configurable options generally have the code, description, and language translations of the selected value returned, along with the id reference to the glossary record.

The language translations are contained within a `localeData` element. Prior to release 18.0, a separate `localeData` element was returned for each language the portal had configured, now just the languages that the portal actually has active are returned. If a translation is not present for a language, the element is omitted, so if the portal has no alternative languages active, just a `localeData` element for the base language is returned.

**Note:**

Some APIs allow a locale parameter to be passed to optionally return a specific language, or languages. However, use of the parameter can have an impact on the call's response time, due to the additional selection processing.

Enumeration Fields

Certain reference fields have their options defined as system text rather than a configurable glossary. Referred to as enumerations, these are used where the selections are generally static, such as Yes/No options. Prior to release 18.0, just the code of an enumeration was returned, now the description of enumeration value is also returned as a description element.

Editable Text Selector Fields

Certain reference fields allow for a statement to be selected from a glossary and inline values entered after selection. Referred to as editable text selectors, these are used to build statements with variable content, but using a consistent format; for example, to describe cooking instructions in a common way without having to have an entry in the glossary for every permutation of time and temperature. Prior to release 18.0, just the separate statement and value elements were returned, now a packCopyText element containing the formed statement is also returned.

Key Features of the API Process

This section summarizes the key aspects of the API process:

- The external system makes a call to Supplier Evaluation using HTTPS. The external system must be registered in Supplier Evaluation, and an Application record created within IDCS corresponding to the External System.
- Calls to RESTful services use the GET, POST, PUT, and DELETE methods to retrieve, create, update, and delete records respectively. Calls to SOAP services use a predefined set of methods to perform the operations.
- Each call is logged in the Web Service Log in Supplier Evaluation. The request and response XML messages are attached to the logs.
- A predefined set of parameters is available to apply filters to the call, to retrieve a specific record, or set of records.
- Data is returned as XML, as defined by the associated WADL or WSDL schema. Only elements that contain data are returned; empty elements are omitted.
- For RESTful APIs, an initial GET call to the service returns the unique internal key ID of the returned records in either the recordId or the id element. The record ID is then used in a subsequent call using a URI to retrieve the record's full XML data set, or to perform an update operation.

Update functions require the full set of fields and their values to be passed, as the full contents of the record are replaced. If all fields are not present in the external system, the update process would need to first retrieve the current contents of the record from Supplier Evaluation before applying the changes to that data set, and submitting the full data set as the update.

- There could be several initiators for the data exchange, including the following:

- Human initiation, where an individual makes the request, such as by clicking a button to run the request.
- End user action, where the completion of an activity in the external system triggers the exchange.
- Scheduled request (or polling), where the external system sends or checks for data on a regular basis. For example, it may be looking for newly created Suppliers or Sites with a change in status.

For security purposes, the APIs rely on the external system polling, retrieving, or submitting calls to the APIs. Supplier Evaluation does not push data externally.

- The nature of the APIs may require multiple requests to be made, for example:
 - Request a list of Sites with their IDs.
 - Request the timestamp reference for each Site.
 - Identify any Sites changed or created within a given time scale.
 - Request the full details for that Site.

These types of requests can be automated. The external system is altered to enable the operations to take place in sequence according to a pre-defined algorithm.

Certain modifications can also be made to the external system to enable the storage of data returned from Supplier Evaluation. For example, if the external system requests a record to be created in Supplier Evaluation, a confirmation response is sent, containing the record ID. This ID may then be stored in the external system for future reference.

**Note:**

An External System must be specifically granted access to an API, either by the service or its endpoints. If none are specified, the External System will not have access to any of the APIs.

If a service is selected, it grants access to all of its endpoints, so there is no need to select the individual endpoints (if access is required to all of them); but selecting the service and all its endpoints would have the same effect.

Endpoints can be selected individually to grant granular access to specific endpoints of a service. In this case, it appears that if the service is also selected, that overrides the endpoint selections, thus providing access to them all anyway. So if wanting to grant access to just specific endpoints, the way to do it is not to select the service, but just the individual endpoints.

3

APIs Available in Supplier Evaluation

The APIs deployed in a particular implementation of Supplier Evaluation can be found at / services: <https://<hostname>/<context>/services>.

Individual Services

The following table lists the individual services available in Supplier Evaluation, indicating at which version each was introduced and, where appropriate, at which version it was deprecated.

Supplier Evaluation API Services

Service Name and Functions	Description	Type	From Version	To Version
UserRestService				
List of Values	Retrieves a list of retailer or supplier users in a paged list.	REST	NA	NA
Retrieve Record by ID	Retrieves a single user's details. There are no request parameters, but the URL contains the id parameter that determines the record to retrieve.	REST	NA	NA
Retrieve Record by Business Key	Retrieves a single user's details. There are no request parameters, but the URL contains the login id parameter that determines the record to retrieve.	REST	NA	NA
Check Record Modification Timestamp	Retrieves the last modification time for a user. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
Create Record	Creates a new retailer or supplier user.	REST	NA	NA
Update Record	Updates an existing retailer or supplier user.	REST	NA	NA
SupplierRestService				
List of Values	Retrieves a list of suppliers in a paged list.	REST	NA	NA
Retrieve Record by ID	Retrieves a single supplier's details. There are no request parameters, but the URL contains the id parameter that determines the record to retrieve.	REST	NA	NA
Retrieve Record by Business Key	Retrieves a single supplier's details. There are no request parameters, but the URL contains the key code parameter that determines the record to retrieve.	REST	NA	NA
Check Record Modification Timestamp	Retrieves the last modification time for a supplier. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA

Service Name and Functions	Description	Type	From Version	To Version
Create Record	Creates a new supplier.	REST	NA	NA
Update Record	Updates an existing supplier .	REST	NA	NA
SiteRestService				
List of Values	Retrieves a list of sites in a paged list.	REST	NA	NA
Retrieve Record by ID	Retrieves a single site's details. There are no request parameters, but the URL contains the id parameter that determines the record to retrieve.	REST	NA	NA
Retrieve Record by Business Key	Retrieves a single site's details. There are no request parameters, but the URL contains the supplier and site code parameters that determine the record to retrieve.	REST	NA	NA
Check Record Modification Timestamp	Retrieves the last modification time for a site. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
Create Record	Creates a new site.	REST	NA	NA
Update Record	Updates an existing site.	REST	NA	NA
ContactRestService				
List of Values	Retrieves a list of contacts in a paged list.	REST	NA	NA
Retrieve Record by ID	Retrieves a single contact's details. There are no request parameters, but the URL contains the id parameter that determines the record to retrieve.	REST	NA	NA
Retrieve Supplier Contact Record by Business Key	Retrieves a single supplier contact's details. There are no request parameters, but the URL contains the supplier code, name, and email address parameters that determine the record to retrieve.	REST	NA	NA
Retrieve Site Contact Record by Business Key	Retrieves a single site contact's details. There are no request parameters, but the URL contains the supplier and site codes, name, and email address parameters that determine the record to retrieve.	REST	NA	NA
Check Record Modification Timestamp	Retrieves the last modification time for a contact. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
BusinessCategoryService				
List of Values	Retrieves a list of Business Categories in a paged list.	REST	NA	NA
Retrieve Record	Retrieves a single Business Category (a path of all levels). There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA

Service Name and Functions	Description	Type	From Version	To Version
Check Record Modification Timestamp	Retrieves the last modification time for a Business Category. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
Create Record	Creates a new Business Category.	REST	NA	NA
Update Record	Updates an existing Business Category.	REST	NA	NA
Delete Record	Deletes an existing Business Category.	REST	NA	NA
AuditRestService				
List of Values (Audit)	Retrieves a list of Audits or Visits in a paged list.	REST	NA	NA
Retrieve Record by ID (Audit)	Retrieves a single Audit or Visit (and its Non Conformances/Issues). There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
Retrieve Record by Business Key (Audit)	Retrieves a single Audit or Visit (and its Non Conformances/Issues). There are no request parameters, but the URL contains the key code parameter that determines the record to retrieve.	REST	NA	NA
Check Record Modification Timestamp (Audit)	Retrieves the last modification time for an Audit or Visit. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
Create Record (Audit)	Creates a new Audit or Visit.	REST	NA	NA
Update Record (Audit)	Updates an existing Audit or Visit.	REST	NA	NA
List of Values (Checklist)	Retrieves a list of Checklists in a paged list.	REST	NA	NA
Retrieve Record by ID (Checklist)	Retrieves a single Checklist. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
Retrieve Record by Business Key (Checklist)	Retrieves a single Checklist. There are no request parameters, but the URL contains the key code parameter that determines the record to retrieve.	REST	NA	NA
Create Record (Checklist)	Creates a new Checklist.	REST	NA	NA
Update Record (Checklist)	Updates an existing Checklist.	REST	NA	NA
Delete Record (Checklist)	Deletes an existing Checklist.	REST	NA	NA
TaskRestService				
List of Values	Retrieves a list of current tasks for the user in the given language.	REST	NA	NA
UrgentAssignmentsRestService				
Number of Urgent Items	Retrieves the number of urgent assignments pending for a given user.	REST	NA	NA
ArtworkRestService				

Service Name and Functions	Description	Type	From Version	To Version
Started Activities	Retrieves a list of Project Activities that are flagged as Artwork activities, and their status has changed to Started within the specified date range.	REST	NA	NA
Update Record	Updates the sub-status of existing Artwork Project Activities.	REST	NA	NA
DataPrivacyService				
Right to Access	Retrieves personal data from the system relating to the name of an individual.	REST	NA	NA
Right to be Forgotten	Erases personal data from the system for an individual (subject to conditions).	REST	NA	NA
AttachmentRestService				
Retrieve Report Attachment	Retrieves attachment files that have been generated by the system as report outputs.	REST	NA	NA
Retrieve Attachment	Retrieves files that have been attached to records.	REST	NA	NA
ScorecardRestService				
List of Values	Retrieves a list of Assessments in a paged list.	REST	NA	NA
Retrieve Record by ID	Retrieves a single Assessment. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
Retrieve Record by Business Key	Retrieves a single Assessment. There are no request parameters, but the URL contains the key code parameter that determines the record to retrieve.	REST	NA	NA
Check Record Modification Timestamp	Retrieves the last modification time for a Assessment. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
Update Record	Updates an existing Assessment.	REST	24.1.301	NA
ProjectRestService				
List of Values	Retrieves a list of Processes in a paged list.	REST	NA	NA
Retrieve Record by ID	Retrieves a single Process. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
Retrieve Record by Business Key	Retrieves a single Process. There are no request parameters, but the URL contains the key code parameter that determines the record to retrieve.	REST	NA	NA
Check Record Modification Timestamp	Retrieves the last modification time for a Process. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
Update Record	Updates an existing Process.	REST	25.1.301	NA

Service Name and Functions	Description	Type	From Version	To Version
ActivityRestService				
List of Values	Retrieves a list of Process Activities in a paged list.	REST	NA	NA
Retrieve Record by ID	Retrieves a single Process Activity. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
Retrieve Record by Business Key	Retrieves a single Process Activity. There are no request parameters, but the URL contains the key code parameter that determines the record to retrieve.	REST	NA	NA
Check Record Modification Timestamp	Retrieves the last modification time for a Process Activity. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	NA	NA
Update Record	Updates an existing Process Activity.	REST	25.1.301	NA
Update Status	Updates the status of an existing Process Activity.	REST	25.1.301	NA
ProductRecordService				
List of Values	Retrieves a list of Product Records in a paged list.	REST	24.1.301	NA
Retrieve Record by ID	Retrieves a single Product Record's details. There are no request parameters, but the URL contains the id parameter that determines the record to retrieve.	REST	24.1.301	NA
Retrieve Record by Business Key	Retrieves a single Product Record's details. There are no request parameters, but the URL contains the key code parameter that determines the record to retrieve.	REST	24.1.301	NA
Check Record Modification Timestamp	Retrieves the last modification time for a Product Record. There are no request parameters, but the URL contains the record id parameter that determines the record to retrieve.	REST	24.1.301	NA
Create Record	Creates a new Product Record.	REST	24.1.301	NA
Update Record	Updates an existing Product Record.	REST	24.1.301	NA

Services Grouped by Area

The following sections summarize the service features available by area of Supplier Evaluation functionality.

User and Supplier APIs

The following service features are applicable to the functionality available for user and supplier APIs:

Allows Supplier, Site, Contact, and User records to be created, updated, and extracted.
Does not allow deletion of records.
Does not import or export file attachments.
Allows Assessment records to be updated and extracted.
Allows Audit & Visit records to be created, updated, and extracted.
Does not include Alerts.
A system log of service transactions is kept (not reportable).

Audits API

The following service features are applicable to the functionality available for the audits API:

Allows Audit and Visit records to be created, updated, and extracted.
Includes the associated Issues/Non Conformances and Checklists.
Does not allow deletion of records.
Does not import or export file attachments.
A system log of service transactions is kept (not reportable).

Data Privacy API

The following service features are applicable to the functionality available for the Data Privacy API:

Right to Access requests, to retrieve personal data relating to the name of an individual.
Right to be Forgotten requests, to erase personal data for an individual (subject to conditions).
Certain due diligence approval data is not permitted to be erased.
Data is erased by being overwritten with randomly-generated text.
The erasure of data is not recorded in Change History logs.

Product APIs

The following service features are applicable to the functionality available for product APIs:

Allows Product Records to be created, updated, and extracted.
Does not import file attachments.
Not intended for bulk upload of Product Records.
If the full content of the Product Record is not included in the data feed, subsequent manual data entry may be required.
A system log of service transactions is kept (not reportable).

Project/Process & Activity APIs

The following service features are applicable to the functionality available for the project (process) and activity APIs:

Allows Project (Process) records to be extracted.
Allows Activity records associated to a Project (Process) to be extracted.
Projects (Processes) and Activities can be updated.
Does not allow creation or deletion of records.
Does not import or export file attachments.
A system log of service transactions is kept (not reportable).

Miscellaneous

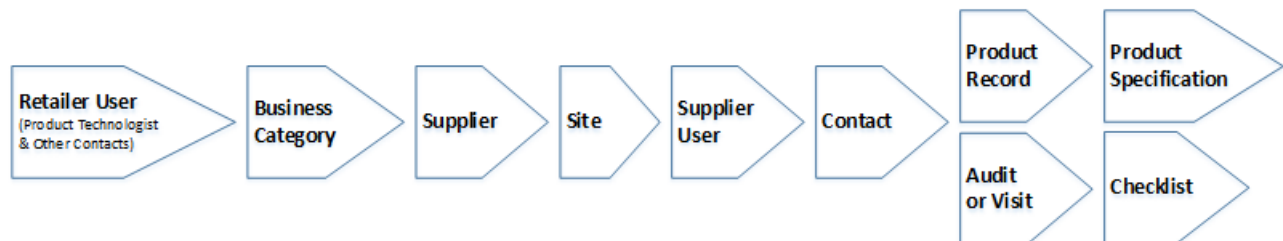
The following service features are also available:

Tasks & Urgent Items - For retrieval of a specific user's list of Tasks or count of Urgent Items.
Business Categories - Allows Business Categories to be created, updated, deleted, and extracted.
Attachments & Reports - For retrieval of file attachments that have been generated by the system as report outputs, and files that have been attached to records.

Dependencies

Following is the typical sequence of records to be created in order to create all of the above services:

Figure 3-1 Dependencies



The following table indicates the dependencies of each type of record that can be created using the APIs, that is, what configuration or other records must be in place for the creation to succeed:

Record Creation Dependencies

Record Type	Configuration	Other Record Types	Comments
User	Company (mandatory) Languages (mandatory) User Roles (mandatory) Authority Profiles Time Zones Users (Managers and Delegates) Areas	Supplier (if a supplier user)	When creating a user, it is associated to either the retailer or a supplier. If it is a supplier user, the ID of the Supplier record must be provided. The notification email sent to a new user is not generated when creating a user through the API.

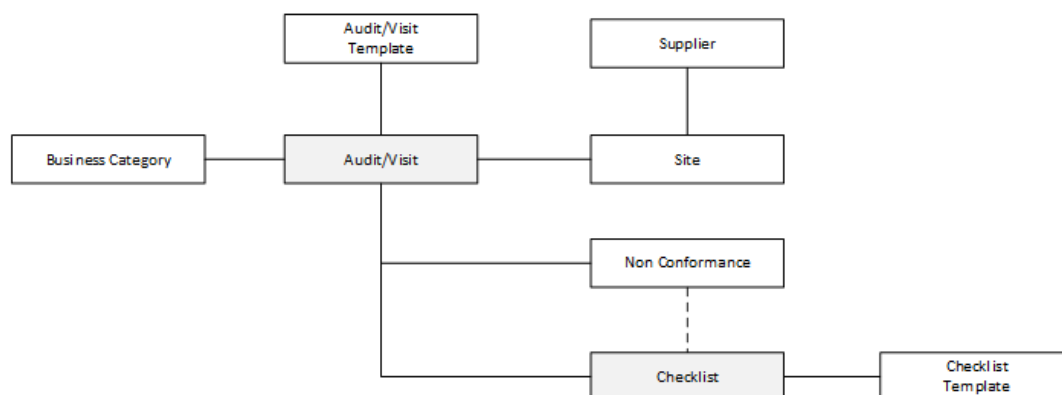
Record Type	Configuration	Other Record Types	Comments
Supplier	Billing Codes (mandatory) Business Units (mandatory) Supplier Types (mandatory) Countries Supplier Top Grades Currencies Invoicing Systems Supplier Statuses Areas	NA	When creating a Supplier, omit the Site and Supplier Contact elements, and omit the Billing Details ID. Create Sites, Contacts, AND Users after creating the Supplier. The initial user account is not automatically created when creating a supplier through the API.
Site	Technologists (mandatory) Business Categories (mandatory) Site Types (mandatory) Business Units Currencies Invoicing Systems Countries Audit Reasons Reference Types Reference Statuses Site Top Grades Site Statuses Risk LevelsOther Production Details	Supplier User (Technologist)	When creating a Site, the ID of the Supplier record must be provided. Omit the Assessments and Audit elements, and omit the Billing Details ID and Reference ID. If assigning a Business Category, the ID of the Business Category record must be provided; if assigning a Technologist, the IDs of the User record must be provided. Create Contacts after creating the Site.
Contact	Contact Roles (mandatory)	User Supplier Site (if a Site contact)	When creating a Contact, the IDs of the associated User and Person records must be provided. The Person record holds an individual's personal contact details; the Contact record holds the individual's contact details in relation to that role (which may be different per role); the User record represents the individual's user account. A contact is associated to a supplier and/or a site; the IDs of the Supplier and Site records must be provided as appropriate. Omit the Site element if not a site-level contact.
Business Category	Business Categories (parent levels) Specification Types	NA	When creating a Business Category, if the category is a lower-level category, the ID of its parent Business Category record must be provided.

Record Type	Configuration	Other Record Types	Comments
Audit/Visit	Audit/Visit Template (mandatory) Audit/Visit Status (mandatory) Technologist (mandatory) People Present (mandatory) Supplier Contacts Business Categories Non Conformance: Assigned to Completed by Approved by	Supplier Site User	An Audit or Visit is linked to a supplier/site; the IDs of the Supplier and Site records must be provided. If assigning a Business Category, the ID of the Business Category record must be provided; if assigning a Technologist or People Present, the IDs of the User record must be provided. Non Conformances may include links to User records for the users who the issue is assigned to, was completed by, or is approved by. If the person is not a user of the application, a corresponding text field can be used to record the name. Also, see notes below.
Checklists	Audit/Visit (mandatory) Checklist Question (mandatory)	Audit/Visit Checklist Template	A Checklist is linked to an Audit/Visit, the IDs of the Audit or Visit record must be provided. The Checklist's questionnaire questions and available answers are derived from the Checklist Templates assigned to the Audit/Visit, the IDs of the Template records must be provided.
Product Record	Product Record Statuses (mandatory) Technologist (mandatory) Other Contacts (mandatory) Business Categories	Supplier (if Active) Site (if linked to a site) User	A Product Record may be linked to a supplier/site; the IDs of the Supplier and Site records must be provided as appropriate. If assigning a Business Category, the ID of the Business Category record must be provided; if assigning a Technologist or Other Contact, the IDs of the User record must be provided. Technologist and Other Contacts are mandatory, however will be defaulted to TBC on creation if not provided (the dummy TBC User record must already exist). For the status to be Active the Product Record must be linked to a Supplier.

Notes on the Creation and Maintenance of Audits/Visits and Checklists

- The following diagram shows the main links between Audits/Visits, their Checklists, and other entities:

Figure 3-2 Links Between Audits/Visits and Other Entities

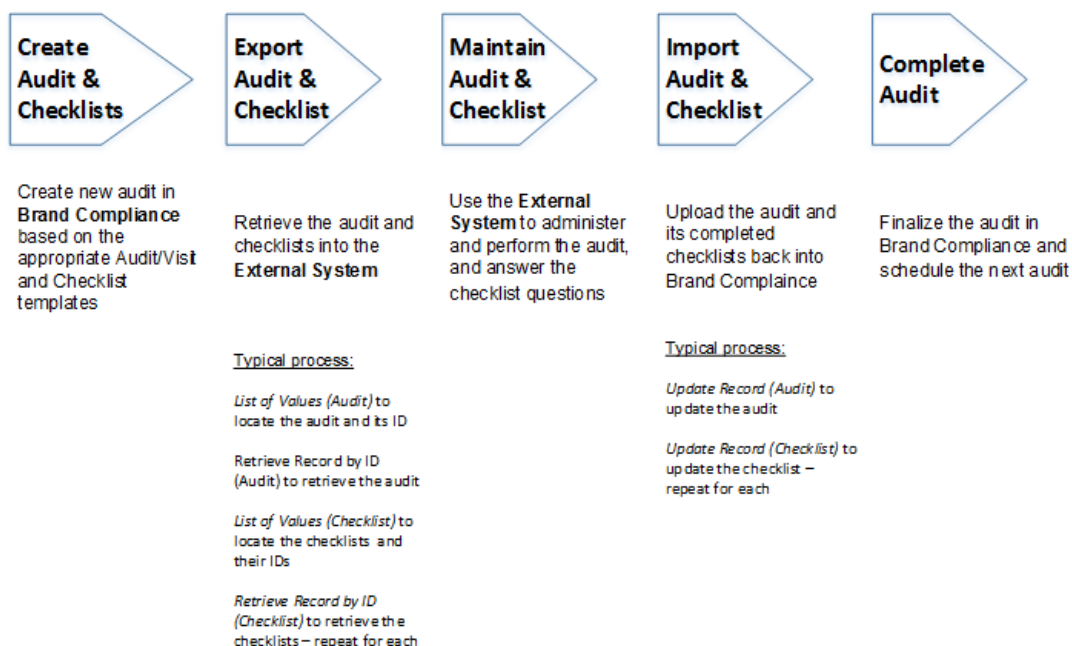


- For the creation of an Audit/Visit and its Checklists, in addition to providing the links to the associated Audit/Visit Template and Checklist Template records, certain values from the templates must also be included in the payload of the Audit/Visit Create and Update functions.
- It is recommended that the creation of Audits/Visits and Checklist be managed within the Supplier Evaluation application. Once created, records can be taken into the external system where the audit is performed, prior to importing back into Supplier Evaluation.

This approach ensures the necessary template fields will have been obtained with the initial Retrieve Record function.

The following diagram shows the stages of a typical scenario:

Figure 3-3 Stages of a Typical Scenario



- If it is necessary to create audits or visits from scratch in the external system, the recommended approach would be to create a set of *gold standard* audits/visits in Supplier Evaluation, which are assigned to a dummy supplier and site.

To create a new audit or visit, the process would be to retrieve the appropriate *gold standard* audit/visit through the GET function, amend the returned XML data to reflect the

audit/visit to be created, and then submit the creation of the new audit/visit (plus any non conformances and checklists) using the POST function.

- If a checklist audit is to be created using the Create Audit service, the Audit/Visit record must first be created using the Create Audit function, followed by the creation of each of its Checklist records.

Creating an audit through the API which is linked to a checklist template does not automatically create the checklists. They must be individually created using the Create Checklist function.

- An audit/visit's non conformances are included as separate elements within the audit/visit data. An audit/visit's non conformances are allocated when creating or updating the audit/visit.

 Note:

The API does not automatically create non conformances based on the answers of checklist questions.

Linking to Related Records

Links to associated records are included in the XML as an id element, which holds the record ID of the related record. In some cases, a code element also holds the related record's business key.

For example, the XML returned from the SiteRestService API will contain a supplier element, which itself contains an id element and a code element. The value in the id element can be used by the SupplierRestService Retrieve Record by ID function to retrieve the full details of the Supplier record. Alternatively, the value in the code element can be used by the SupplierRestService Retrieve Record by Business Key function to retrieve the Supplier record.

The id element is also used in the XML to hold the record ID of a value that is held in a glossary such as a recipe ingredient or a country.

The following table shows where an API provides links to other records, and the id or code elements used to locate those records. The list of notes below the table indicates the service to use to retrieve the related record.

Related Record Links

API	Links to Records
Supplier	Contact: contactDetail/id (Contact record ID link). See note 7. Contact (User): person/id (User record ID link). See note 1. Site: site/id (Site record ID link). See note 3. Site: site/code site (Site business key link). See note 4.

API	Links to Records
Site	Supplier: supplier/id (Supplier record ID link). See note 5. Supplier: supplier/code (Supplier business key link). See note 6. Business Categories: businessCategories/id (Business Category record ID link). See note 8. Lead Business Category: leadBusinessCategory/id (Business Category record ID link). See note 8. Lead Technologist (User): leadTechnicalManager/id (User record ID link). See note 1. Lead Technologist (User): leadTechnicalManager/code (User business key link). See note 2. Other Technologists (User): otherTMs/id (User record ID link). See note 1. Other Technologists (User): otherTMs/code (User business key link). See note 2. Contact: siteContact/id (Contact record ID link). See note 7.
Contact	Contact (User): person/id (User record ID link). See note 1. Supplier: company/id (Supplier record ID link). See note 5. Supplier: Company/code (Supplier business key link). See note 6. Site: site/id (Site record ID link). See note 3. Site: Site/code (Site business key link). See note 4.
User	Supplier: supplier/id (Supplier record ID link). See note 5. Supplier: Supplier/code (Supplier business key link). See note 6.
Audit/Visit	Supplier: supplier/id (Supplier record ID link). See note 5. Supplier: supplier/code (Supplier business key link). See note 6. Site: site/id (Site record ID link). See note 3. Site: Site/code (Site business key link). See note 4. People Present (User): user/id (User record ID link). See note 1. People Present (User): user/code (User business key link). See note 2. Lead Technologist (User): leadTechnicalManager/id (User record ID link). See note 1. Lead Technologist (User): leadTechnicalManager/code (User business key link). See note 2. Business Categories: businessCategories/id (Business Category record ID link). See note 8. Lead Business Category: leadBusinessCategory/id (Business Category record ID link). See note 8. Non Conformance Assigned To (User): assignedTo/id (User record ID link). See note 1. Non Conformance Completed By (User): completedBy/id (User record ID link). See note 1. Non Conformance Approved By (User): approvedBy/id (User record ID link). See note 1.
Checklist	Audit/Visit: audit/id (Audit/Visit record ID link). See note 9.
Scorecard/ Assessment	Supplier: supplier/id (Supplier record ID link). See note 5. Supplier: supplier/code (Supplier business key link). See note 6. Site: site/id (Site record ID link). See note 3. Site: Site/code (Site business key link). See note 4. Product Record (Item Assessment): productRecord/id (Product Record record ID link). See note 8. Product Record (Item Assessment): productRecord/code (Product Record business key link). See note 9. Person Responsible (User): personResponsible/id (User record ID link). See note 1. Business Category: businessCategories/id (Business Category record ID link). See note 9. Submitted By (User): submittedBy/id (User record ID link). See note 1.

API	Links to Records
Project/Process	Parent Project (Process): parentProject Supplier: supplier/id (Supplier record ID link). See note 5. Supplier: supplier/code (Supplier business key link). See note 6. Site: site/id (Site record ID link). See note 3. Site: site/code (Site business key link). See note 4. Process Manager (User): projectManager/id (User record ID link). See note 1.
Activity	Project (Process): projectId

Notes:

1. User record ID link: UserRestService - Retrieve Record by ID
2. User business key link: UserRestService - Retrieve Record by Business Key
3. Site record ID link: SiteRestService - Retrieve Record by ID
4. Site business key link: SiteRestService - Retrieve Record by Business Key
5. Supplier record ID link: SupplierRestService - Retrieve Record by ID
6. Supplier business key link: SupplierRestService - Retrieve Record by Business Key
7. Contact record ID link: ContactRestService - Retrieve Record by ID
8. Business Category record ID link: BusinessCategoryService - Retrieve Record
9. Audit/Visit Record ID link: AuditRestService - Retrieve Record by ID (Audit)

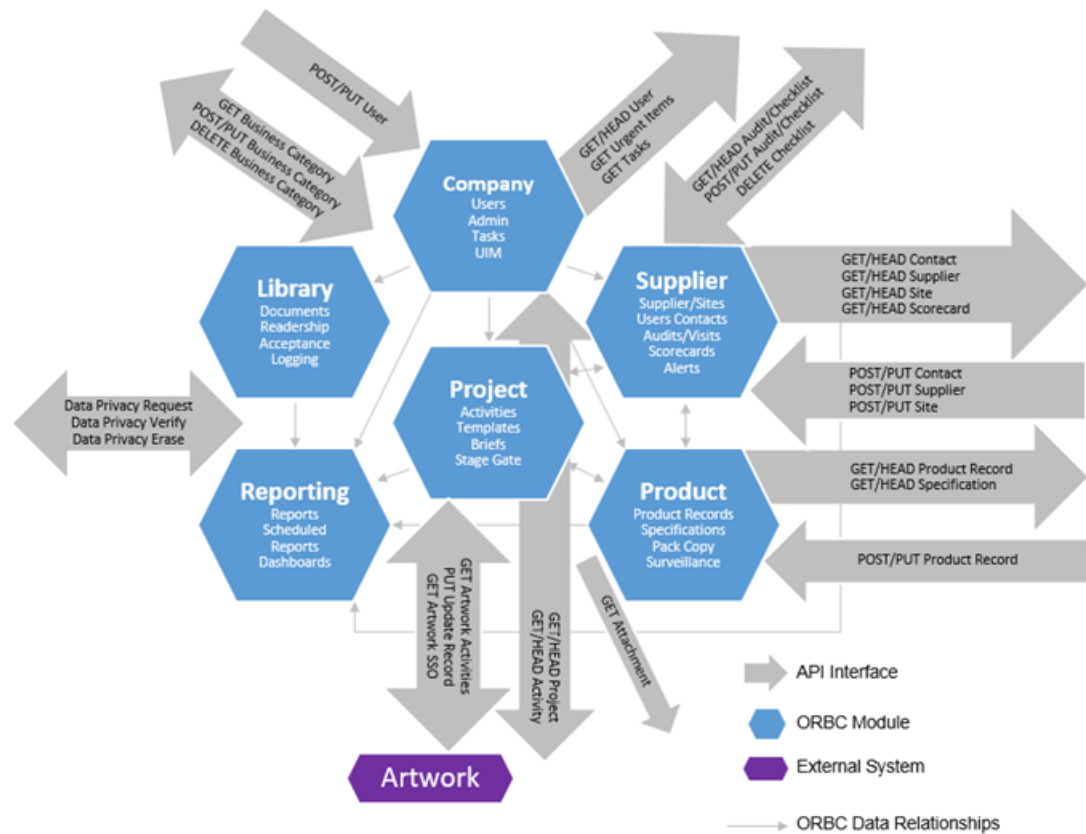
4

RESTful APIs

The following RESTful APIs are available:

- [UserRestService](#): Users
- [SupplierRestService](#): Suppliers
- [SiteRestService](#): Sites
- [ContactRestService](#): Supplier/Site Contacts
- [TaskRestService](#): Users' Tasks
- [UrgentItemsRestService](#): Users' Urgent Items
- [ArtworkRestService](#): Artwork Activities
- [BusinessCategoryService](#): Business Categories
- [AuditRestService](#): Audits and Visits
- [DataPrivacyService](#): Data Privacy Requests
- [AttachmentRestService](#): Attachment Files
- [ScorecardRestService](#): Assessments
- [ProjectRestService](#): Processes
- [ActivityRestService](#): Process Activities
- [ProductRecordRestService](#): Product Records

Figure 4-1 Overview of RESTful APIs

**Note:**

Only minimal business logic and validation is applied by the Supplier Evaluation REST APIs.

When creating or updating records, all values must be included. If a value or element is omitted from the XML, the field contents will be cleared when updating the record.

Generally, only the validation necessary to ensure the system's referential integrity is applied when updating records through the APIs. It is therefore possible to set the contents of some fields with values that are rejected when the user manually edits or validates the record through the UI. In some cases, it may be necessary to regress these records back to a prior status in order to edit and correct values that have been set incorrectly through the UI.

Parameters and Filtering

Various parameters can be included in calls to the APIs, generally to define what data is to be returned, but can also control how records are returned. The following table lists some commonly used parameters.

Common Parameters

Parameter	Type	Description
offset	int	Used with pageSize to control the paging of a returned list of records. Specifies the starting point for the retrieval of records. If not specified, zero is assumed. For example, to retrieve 150 records: 1. A call with offset = 0 and pageSize = 50 returns the first 50 records. 2. Then, a call with offset = 50 and pageSize = 50 returns the next 50. 3. Then, a call with offset = 100 and pageSize = 50 returns the final 50.
pageSize batchSize	int	Used with offset to control the paging of a returned list of records. Specifies the number of entries in each page of returned list of records. If not specified, 30 is assumed. The maximum is 100.
isActive	Boolean	Used to only return records that are flagged as being active (TRUE).
modifiedSince	string	Used to locate records that have been created or updated since a specific date/time.
modifiedUntil	string	Used to locate records that have been updated or created until a specific date/time.

When a list of records is returned, it may include a `previousPage` and `nextPage` element. These elements provide URI links to the previous or next pages of records respectively. A `totalRecords` element is included that shows the total number of records available for retrieval.

For example, using the List of Values function to retrieve a list of User records, where there are 150 records to be retrieved, with the `pageSize` parameter set to 50 returns the first 50 records within entity elements, along with a `nextPage` element containing a URI link to the next 50 records which can be used to retrieve the next 50 records. When the second page of 50 records are returned, the XML includes a `nextPage` URI link to the final 50 records, along with a `previousPage` URI link to the first 50. The following graphic illustrates this example.

Figure 4-2 Example of Retrieving a List of Users



```

<entries xsi:type="userLink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <recordId>100</recordId>
  <recordLink>http://example.com/services/rest/user/rest/user/100</recordLink>
  <loginId>John Smith</loginId>
  <name>John Smith</name>
</entries>

<previousPage>example.com/services/rest/user/?offset=0&pageSize=50</previousPage>
<nextPage>example.com/services/rest/user/?offset=100&pageSize=50</nextPage>

<totalRecords>150</totalRecords>

```

A callout box with the text "Last entry in the list of 50 returned users" points to the `<name>John Smith</name>` element in the XML snippet.

The `totalRecords` element shows the total number of retrievable records. The `previousPage` and `nextPage` elements are only present if there are previous and/or next pages to be retrieved.

**Note:**

Date/Time parameters must be provided in the YYYY-MM-DD hh:mm:ss format.

When retrieving records, the returned XML only includes elements that actually contain data; empty elements are omitted.

The details of the returned records are contained in an entities element, and repeated in an entryArray element. The entryArray element is a representation from the Java objects and can be ignored.

Where a date/time is passed in as a parameter to an endpoint, such as fromDate and toDate, validation checks for a valid date/time. If not valid, an HTTP 417 is returned, along with an XML error message detailing the exact issue with the date/time.

Error Messages

The Supplier Evaluation APIs use standard web services messaging protocols to notify the success or failure of a call to the service. Where an API handles a specific error condition, details of the returned message can be found in the following sections. The following table lists various generic error messages that may be returned by calls to any of the APIs.

Generic Error Messages

Element	Message	Meaning
userId	HTTP Status 401 - Bad credentials	User ID or password is invalid
Password	HTTP Status 401 - Bad credentials	User ID or password is invalid
Disabled	HTTP Status 401 - User is disabled	External System account is not enabled in Supplier Evaluation
offset	HTTP 417 IllegalStateException: Offset must be a positive integer	Invalid offset value - must be between zero and 2,147,483,647
offset	HTTP 404 Not Found	Not numeric or an integer
pageSize	HTTP 417 IllegalStateException: The Page Size must be between 1 and 100	Invalid page size value
	HTTP 404 Not Found	Record cannot be located due to an invalid key or ID
	HTTP 417 Expectation Failed	Record cannot be located due to an invalid ID
	HTTP Status 500 - Internal Server Error	Unspecified internal error occurred
	HTTP Status 500 - Internal Server Error	Non-numeric value in numeric parameters
modifiedSince	HTTP 417 <errorMessage>	Not a valid date, whereby <errorMessage> will identify the issue For example: IllegalStateException: Cannot parse "2018-03-25 01:00:00": Illegal instant due to time zone offset transition (Europe/London) In this example, due to the clocks moving forward (in the server's time-zone), the passed in date is invalid

Element	Message	Meaning
modifiedUntil	HTTP 417 <errorMessage>	Not a valid date, whereby <errorMessage> will identify the issue (see example above)
	HTTP 400 Bad Request	Malformed XML, or general error in content of the XML

Wildcard Searches

The % character can be used as a wildcard filter when locating the records.

Using the % at the start of the string will search for matches where the given string ends with the text being searched for, for example, %smith could be used to search for all users with a surname of smith.

Using the % at the end of the string will look for matches where the given string begins with the text being searched for, for example, mark% could be used to search for all users with the name mark.

The following table shows which APIs and functions support the use of wildcards.

Wildcard Searches

API	Function
User	List of Values
Supplier	List of Values
Site	List of Values
Contact	List of Values Retrieve Supplier Contact Record by Business Key Retrieve Site Contact Record by Business Key
Business Category	List of Values
Task	List of Values
Urgent Items	Number of Urgent Items

Notes:

- Wildcard searches are not case sensitive so, for example, searching a name field for John Smith or john smith will return the same matches.
- When searching glossary parameters, be aware that they relate to the code (not the description) of the glossary entry.
- Wildcard searches can typically be applied to parameters that are string type. See the Wildcard Search Supported column in the APIs' List of Values functions for which URL parameters support it.

Deleted Records

By default, soft deleted records are returned by the APIs (the XML contains a deleted element, which is set to true). To exclude soft deleted records, it is necessary to use a *softDelete=false* parameter when calling the API. For example:

/creations-core/services/rest/site?softDelete=false

UserRestService

This section describes the API for managing retailer and supplier users. The following functions are available:

- **List of Values:** retrieves a list of users
- **Retrieve Record by ID:** retrieves a User record using its unique identifier
- **Retrieve Record by Business Key:** retrieves a User record using its business key
- **Check Record Modification Timestamp:** retrieves the timestamp when a User record was last updated
- **Create Record:** creates a new User record
- **Update Record:** updates an existing User record

List of Values

Description

Retrieves a list of users in a paged list. Use this function to retrieve a simple list of user names and IDs, or to locate User record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/user **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
offset	Optional	int	No	No	60
pageSize	Optional	int	No	No	30
name	Optional	string	No	Yes	John Smith
jobTitle	Optional	string	No	Yes	Process Manager
language	Optional	string	No	Yes	en_GB
country	Optional	string	No	Yes	UK
userRoles	Optional	string	No	Yes	PROCESS ADMINISTRATOR
userType	Optional	string	Yes	No	RETAILER
authorityProfiles	Optional	string	No	Yes	PROCESS MANAGER
siteStatus	Optional	string	Yes	No	ACTIVE-INACTIV E

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
supplierActive	Optional	string	No	No	true returns active users no value returns both active and inactive users
modifiedSince	Optional	string	No	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	No	2015-05-19 13:30:39

Example URLs

.../services/rest/user/?offset=2&pageSize=20 .../services/rest/user/?siteStatus=Active .../
services/rest/user/?name=Frank%

Response Details

For a successful response, XML is returned with a UserLinkList root element containing an entries element for each matched user. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	User record's internal ID
recordLink	string	URI to the UserRestService Retrieve record service for this user
loginId	string	Login ID / business key to the User record
name	string	User's name

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

In the event that an error occurs, an HTTP 500 response is sent.

Retrieve Record by ID

Description

Retrieves a single User record's details using the record's internal unique ID (which is not visible in the UI). Use this function to retrieve the full details of an individual user.

Endpoint address: /services/rest/user/{id} **HTTP method:** GET

Request Details

The URL contains the {id} parameter that determines the record to retrieve.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
locale	Optional	string	Yes	en_GB~fr Locale country code. If used, fields that have language translations have a localeData element returned for the specified locales if data is present.

Example URL

.../services/rest/user/105

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of a userFullDTO root element containing the individual attributes of the requested User record. If an ID is not specified, a list of all users is returned (per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not found	Invalid {id} - not numeric
id	HTTP 417 IllegalStateException: Cannot find the User to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key

Description

Retrieves a single User record's details using its business key (login ID). Use this function to retrieve the full details of an individual user using its Supplier Evaluation login ID.

Endpoint address: /services/rest/byKey/{loginId} **HTTP method:** GET

Request Details

There are no request parameters, but the URL contains the {loginId} parameter that determines the record to retrieve.

Example URL

.../services/rest/user/byKey/Frank Jones

Response Details

If successful, an HTTP 301 redirect response is sent containing a header Location with the URI of the document requested.

The returned URI represents the call to be made to the Retrieve Record by ID function (with the id substituted with the relevant value), which will return a userFullDTO root element containing the individual attributes of the requested User record.



Note:

If your method of calling the API is configured to automatically follow HTTP 301 redirects, the call may automatically return a userFullDTO root element containing the individual attributes of the requested User record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
loginId	HTTP 404 Not found	Invalid {loginId} - blank or not found
loginId	HTTP 404 Not found	Invalid {loginId} - not found

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a User record. Use this function to determine when a user's details were last updated.

Endpoint address: /services/rest/user/{id} **HTTP method:** HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/user/105

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header showing the date and time of the last update of the requested User record. For example:

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 417 Expectation Failed	Invalid {id} - not found
id	HTTP 404 Not found	Invalid {id} - not numeric

Create Record

Description

Creates a new User record. Use this function to create a new user in Supplier Evaluation based on data sourced from the external system.

Dependencies: If creating a supplier user, the Supplier must be present in the application and its record ID obtained. For more information, see [Dependencies](#).

Endpoint address: /services/rest/user **HTTP method:** POST

Request Details

The body of the request contains a UserFullDTO to specify the details of the user to create. Compared to retrieving a user (which uses the same UserFullDTO type), this request is much shorter. Only the attributes that are to be populated on the created User record need to be included. As a minimum, this must include the fields shown in the following table:

Supplier User Mandatory Fields

Field Name	Element Name
Name	person / name
Email	person / email
Login Id	code
User Role	role / code
Supplier Code	person / supplier / code supplier / code
User Type	userType (Fixed value SUPPLIER)
Deleted Record	deleted (Set to false)
Login Id Disabled	disabled (Set to false)
Login Permitted	loginPermitted (Set to true)
Language	language / code

Example Request XML

This example shows the minimum requirement to be able to create a supplier user.

```
<ns0:userFullDTO xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple"
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full">
  <ns0:code>jphillips</ns0:code>
  <ns0:person>
    <ns0:email>jphillips@supplier.co.uk</ns0:email>
    <ns0:name>Jane Phillips</ns0:name>
    <ns0:supplier>
```



```

        <ns1:code>A0174</ns1:code>
      </ns0:supplier>
    </ns0:person>
    <ns0:role>
      <ns1:code>SUPPLIER ADMINISTRATOR</ns1:code>
    </ns0:role>
    <ns0:supplier>
      <ns1:code>A0174</ns1:code>
    </ns0:supplier>
    <ns0:userType>SUPPLIER</ns0:userType>
    <ns0:deleted>false</ns0:deleted>
    <ns0:disabled>false</ns0:disabled>
    <ns0:loginPermitted>true</ns0:loginPermitted>
  </ns0:userFullDTO>

```

**Note:**

When creating a supplier user, the user type must be set to SUPPLIER. The supplier code must be set to match the required supplier from Supplier Evaluation.

Site User Mandatory Fields

Field Name	Element Name
Name	person / name
Email	person / email
Login Id	code
User Role	role / code
Supplier Code	person / supplier / code supplier / code
Sites	sites / code sites/ supplierCode
User Type	userType (Fixed value SITE)
Deleted Record	deleted (Set to false)
Login Id Disabled	disabled (Set to false)
Login Permitted	loginPermitted (Set to true)
Language	language / code

Example Request XML

This example shows the minimum requirement to be able to create a site user.

```

<ns0:userFullDTO xmlns:ns1="http://www.micros.com/creations/core/domain/dto/vlp0/simple"
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/vlp0/full">
  <ns0:code>jphillips</ns0:code>
  <ns0:person>
    <ns0:email>sphillips@supplier.co.uk</ns0:email>
    <ns0:name>Sean Phillips</ns0:name>
    <ns0:supplier>
      <ns1:code>WS0001</ns1:code>
      <ns1:id>22</ns1:id>
    </ns0:supplier>
  </ns0:person>

```

```

<ns0:role>
  <ns1:code>SITE USER</ns1:code>
</ns0:role>
<ns0:sites>
  <ns1:code>WS0001-0001</ns1:code>
  <ns1:id>26</ns1:id>
  <ns1:supplierCode>WS0001</ns1:supplierCode>
  <ns1:id>22</ns1:id>
</ns0:sites>
<ns0:supplier>
  <ns1:code>WS0001</ns1:code>
  <ns1:id>22</ns1:id>
</ns0:supplier>
<ns0:userType>SITE</ns0:userType>
<ns0:deleted>false</ns0:deleted>
<ns0:disabled>false</ns0:disabled>
<ns0:loginPermitted>true</ns0:loginPermitted>
</ns0:userFullDTO>

```

**Note:**

When creating a site user the user type must be set to SITE. The supplier code must be set to match the required supplier from Supplier Evaluation. The site codes must be set to match the required site from Supplier Evaluation.

Retailer User Mandatory Fields

Field Name	Element Name
Name	person / name
Email	person / email
Login Id	code
User Role	role / code
Supplier Code	person / supplier / code supplier / code (Fixed value RETAILER)
User Type	userType (Fixed value RETAILER)
Deleted Record	deleted (Set to false)
Login Id Disabled	disabled (Set to false)
Login Permitted	loginPermitted (Set to true)
Language	language / code

Example Request XML

This example shows the minimum requirement to be able to create a retailer/portal owner user.

```

<ns0:userFullDTO xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple"
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full">
  <ns0:code>james howard</ns0:code>
  <ns0:person>
    <ns0:email>james.howard@example.com</ns0:email>
    <ns0:name>James Howard</ns0:name>
    <ns0:supplier>

```

```

        <ns1:code>RETAILER</ns1:code>
      </ns0:supplier>
    </ns0:person>
    <ns0:role>
      <ns1:code>POWER USER</ns1:code>
    </ns0:role>
    <ns0:supplier>
      <ns1:code>RETAILER</ns1:code>
    </ns0:supplier>
    <ns0:userType>RETAILER</ns0:userType>
    <ns0:deleted>false</ns0:deleted>
    <ns0:disabled>false</ns0:disabled>
    <ns0:loginPermitted>true</ns0:loginPermitted>
  </ns0:userFullDTO>

```

**Note:**

When creating a retailer/portal owner user, the user type must be set to RETAILER. The supplier code must be set to RETAILER.

Enabling User Accounts

Where the record is linked to another record, such as the Role in these cases, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

Set loginPermitted to true to ensure a corresponding IDCS or OCI IAM profile is automatically created for the user. If an IDCS or OCI IAM profile is not required, setting the loginPermitted attribute to false will skip the IDCS or OCI IAM profile creation.

Set the deleted attribute to false to ensure the record can be seen in the UI. If omitted, or set to true, it will exist, but will not be visible.

Where the record is linked to another record, such as the Role in these cases, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

When a new record is created in Supplier Evaluation, a corresponding IDCS or OCI IAM profile is automatically created to manage login authentication. If an IDCS or OCI IAM profile is not required, setting the loginPermitted attribute to false will skip the IDCS or OCI IAM profile creation.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

For a successful response, an HTTP 200 response is sent with a body containing a UserLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created User record's internal ID
recordLink	string	URI to the newly created User record, for use in a GET request

Element	Type	Description
loginId	string	Login ID / business key to the newly created User record
name	string	User's name

Error Responses

If the supplied data does not result in a valid User (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Element	Message	Meaning
code	ERROR: class com.micros.creations.core.domain.User.code - The condition is invalid	Code not provided
email	ERROR: class com.micros.creations.core.domain.User.person.email - The condition is invalid	Email not provided
name	ERROR: class com.micros.creations.core.domain.User.person.name - The condition is invalid	Name not provided
role	ERROR: class com.micros.creations.core.domain.User.role - A user must have an active authority profile from either roles or additional authority profiles	Role or authority profiles not provided

Update Record

Description

Updates an existing User record. Use this function to update a user's details in Supplier Evaluation based on data sourced from the external system.

Endpoint address: /services/rest/{id} **HTTP method:** PUT

Request Details

The body of the request contains a UserFullDTO to specify the updates to the User record. Compared to retrieving a user (which uses the same UserFullDTO type), this request is much shorter. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

The request content is similar to that for creating a user. After the call, the User record is updated to match the request.

**Note:**

When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the User record.

Any values passed for Login Id or Email Address are ignored, as they cannot be updated once the record has been created.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Supplier Evaluation UI.

Process for Disabling User Accounts

A user's account can be disabled using the Update Record function as follows:

1. Call the Retrieve Record by Business Key GET function or the Retrieve Record by Id function, passing the login id or the record id respectively, to retrieve the full XML for the user.
2. Amend the returned XML as follows:
Set the account to disabled with `<ns0:disabled>true<ns0:disabled>` and `<ns0:loginPermitted>false<ns0:loginPermitted>`
Remove the localeData and activeAuthorityProfiles elements.
3. Call the Update Record PUT function, passing the full amended XML to disable the account.

Response Details

If successful, an HTTP 200 response is sent with a body containing a UserLink element. The UserLink element consists of the returned elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	User record's internal ID
recordLink	string	URI to the User record, for use in a GET request
loginId	string	Login ID/business key to the User record
name	string	User's name

Error Responses

If the supplied data does not result in a valid user (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

SupplierRestService

This section describes the API for managing suppliers. The following functions are available:

- [List of Values](#): retrieves a list of suppliers

- **Retrieve Record by ID:** retrieves a Supplier record using its unique identifier
- **Retrieve Record by Business Key:** retrieves a Supplier record using its business key
- **Check Record Modification Timestamp:** retrieves the timestamp when a Supplier record was last updated
- **Create Record:** creates a new Supplier record
- **Update Record:** updates an existing Supplier record

**Note:**

When integrated with Merchandising, using the SupplierRestAPI to make changes that affect the Billing Details table can have an impact on the integration.

List of Values

Description

Retrieves a list of suppliers in a paged list. Use this function to retrieve a simple list of supplier names and IDs, or to locate Supplier record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/supplier **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
offset	Optional	int	No	No	60
pageSize	Optional	int	No	No	30
supplierStatus	Optional	string	Yes	No	AWAITING REGISTRATION
supplierCode	Optional	string	Yes	Yes	A0001
supplierName	Optional	string	Yes	Yes	West Road Site
supplierType	Optional	string	Yes	Yes	AGENT
country	Optional	string	Yes	Yes	UK
leadBusinessUnit	Optional	string	Yes	Yes	UK
isActive	Optional	string	No	No	true returns active suppliers no value returns both active and inactive suppliers
modifiedSince	Optional	string	No	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	No	2015-05-19 13:30:39

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
invoicingRef	Optional	string	No	Yes	ABC123

Example URLs

.../services/rest/supplier/?offset=2&pageSize=20 .../services/rest/supplier/?
supplierStatus=AWAITING REGISTRATION .../services/rest/supplier/?supplierName=API%

Response Details

For a successful response, XML is returned with a SupplierLinkList root element containing an entries element for each matched supplier. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Supplier record's internal ID
recordLink	string	URI to the SupplierRestService Retrieve record service for this supplier
code	string	Supplier code business key to the Supplier record
name	string	Supplier's name
localName	string	Supplier's name in business language (if used)

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

Element	Message	Meaning
supplierStatus	HTTP 417 IllegalStateException: The Supplier Status <###> does not exist	Invalid supplier status
supplierStatus	HTTP 417 IllegalStateException: The Supplier Statuses <###~###~###> do not exist	Invalid supplier status
supplierType	HTTP 417 IllegalStateException: The Supplier Type <###> does not exist	Invalid supplier type
supplierType	HTTP 417 IllegalStateException: The Supplier Types <###~###> do not exist	Invalid supplier type
country	HTTP 417 IllegalStateException: The Country <###> does not exist	Invalid country
country	HTTP 417 IllegalStateException: The Countries <###~###> do not exist	Invalid country
leadBusinessUnit	HTTP 417 IllegalStateException: The Business Unit <###> does not exist	Invalid business unit
leadBusinessUnit	HTTP 417 IllegalStateException: The Business Units <###~###> do not exist	Invalid business unit

Element	Message	Meaning
pageSize	Error Code: INVALIDRESTSERVICEPAGESIZE	Page Size must be between 1 and 100
offset	Error Code: INVALIDOFFSET	Offset must be a positive integer
supplierStatus	Error Code: INVALIDSUPPLIERSTATUS	Invalid Supplier Status
supplierType	Error Code: INVALIDSUPPLIERTYPE	Invalid Supplier Type
country	Error Code: INVALIDCOUNTRY	Invalid Country
leadBusinessUnit	Error Code: INVALIDBUSINESSUNIT	Invalid Business Unit
isActive	Error Code: INVALIDBOOLEAN	Invalid isActive value Permitted values (not case-sensitive) are: true; yes; 1; false; no; 0

Retrieve Record by ID

Description

Retrieves a single Supplier record's details using the record's internal ID (which is not visible in the UI). Use this function to retrieve the full details of an individual supplier.

Endpoint address: /services/rest/supplier/{id} **HTTP method:** GET

Request Details

The URL contains the {id} parameter that determines the record to retrieve.

URI Parameters

Parameter Name	Mandatory/Optional	Value Type	Multiple Value Separator (~) Supported?	Example
locale	Optional	string	Yes	en_GB~fr Locale country code. If used, fields that have language translations have a localeData element returned for the specified locales if data is present.

Example URL

.../services/rest/supplier/9

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of a supplierFullDTO root element containing the individual attributes of the requested Supplier record. If an ID is not specified, a list of all suppliers is returned (per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 IllegalStateException: Cannot find the Supplier to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key

Description

Retrieves a single Supplier record's details using its business key (supplier code). Use this function to retrieve the full details of an individual supplier using the Supplier Evaluation supplier code.

Endpoint address: /services/rest/supplier/byKey/{code} **HTTP method:** GET

Request Details

There are no request parameters, but the URL contains the {code} parameter that determines the record to retrieve.

Example URL

.../services/rest/supplier/byKey/A0901

Response Details

If successful, an HTTP 301 redirect response is sent containing a header Location with the URI of the document requested.

The returned URI represents the call to be made to the Retrieve Record by ID function (with the id substituted with the relevant value), which will return a supplierFullDTO root element containing the individual attributes of the requested Supplier record.



Note:

If your method of calling the API is configured to automatically follow HTTP 301 redirects, the call may automatically return a supplierFullDTO root element containing the individual attributes of the requested Supplier record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
code	HTTP 404 Not Found	Invalid {code} - blank
code	HTTP 404 Not Found	Invalid {code}

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a Supplier record. Use this function to determine when a supplier's details were last updated.

Endpoint address: /services/rest/supplier/{id} **HTTP method:** HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/supplier/9

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header showing the date and time of the last update of the requested Supplier record.

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Responses

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 Expectation Failed	Invalid {id} - not found

Create Record

Description

Creates a new Supplier record. Use this function to create new suppliers in Supplier Evaluation based on data sourced from the external system.

Dependencies: Create sites and contacts after creating the supplier. For more information, see [Dependencies](#).

Endpoint address: /services/rest/supplier **HTTP method:** POST

Request Details

The body of the request contains a SupplierFullDTO to specify the detail of the supplier to create. Compared to retrieving a supplier (which uses the same SupplierFullDTO type), this request is much shorter. Only the attributes that are to be populated on the created Supplier record need to be included. As a minimum, this must include the fields shown in the following table.

Supplier Mandatory Fields

Field Name	Element Name
Supplier Name	name
Contact Name	supplierContactName
Email	email
Supplier Type	supplierType/code
Lead Business Unit	businessUnit/code
Billing Code	billingCode/code
Status	Status
Supplier Code Confirmed Flag	supplierCodeConfirmed
Deleted Record	deleted (Set to false)
Created On	createdOn (set to current date using format YYYY-MM-DD or to current date and time using format YYYY-MM-DDThh:mm:ss)
Is Active	isActive - see below
Potential Supplier	potentialSupplier (Set to false)

Example Request XML

This example shows the minimum requirement to be able to create a supplier.

```

<ns1:supplierFullDTO
  xmlns:ns2="http://www.micros.com/creations/core/domain/dto/v1p0/simple"
  xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/full">
  <ns1:billingCode>
    <ns2:code>SMALL</ns2:code>
  </ns1:billingCode>
  <ns1:businessUnit>
    <ns2:code>UK</ns2:code>
  </ns1:businessUnit>
  <ns1:email>supplier.contactemail@supplier.co.uk</ns1:email>
  <ns1:name>Name of Supplier</ns1:name>
  <ns1:status>AWAITING REGISTRATION</ns1:status>
  <ns1:supplierContactName>Supplier Contact Name</ns1:supplierContactName>
  <ns1:supplierCodeConfirmed>false</ns1:supplierCodeConfirmed>
  <ns1:supplierType>
    <ns2:code>SUPPLIER_TYPE</ns2:code>
  </ns1:supplierType>
  <ns1:deleted>false</ns1:deleted>
  <ns1:createdOn>2020-09-29T09:00:00</ns1:createdOn>
  <ns1:isActive>false</ns1:isActive>
  <ns1:potentialSupplier>false</ns1:potentialSupplier>
</ns1:supplierFullDTO>

```

Set the deleted attribute to false to ensure the record can be seen in the UI. If omitted or set to true, it will exist, but will not be visible.

The Supplier API only creates the Supplier record, just as the Site API only creates the Site record. The same goes for the User API; it only creates the User record, and similarly the Contact API only creates the Contacts records. Each Supplier record must also have at least one Site record created for it.

Supplier Contacts and their User records must be created using the respective APIs, and linked to the supplier/site with their record IDs. When a new Supplier is created through the API, the process does not automatically create the initial Supplier user.

Use of the `isActive` and `supplierCodeConfirmed` Flags

The `isActive` and `supplierCodeConfirmed` flags on the Supplier record are used to control the Registration process within the application UI.

`isActive` is a Boolean TRUE/FALSE value. It is always FALSE when a Supplier is created in the application UI. If set to FALSE in the API, when the supplier user first logs in to the system, the Supplier Registration wizard runs (as it does if the Supplier has been created through the application UI). Successful completion of the Registration wizard sets the `isActive` flag to TRUE, thus ensuring it only runs once. Therefore, if set to TRUE in the API, when the Supplier user first logs in to the system, the Registration wizard will not run.

`supplierCodeConfirmed` is a Boolean TRUE/FALSE value. It is always FALSE when a Supplier is created through the application UI. It is a security measure for the supplier user logging in for the first time, where they are prompted to enter their Supplier Code. The application changes the value to TRUE if there is a successful match.

Setting both `isActive` and `supplierCodeConfirmed` as FALSE will trigger the Registration wizard process when the supplier user first logs in, prompting for the entry of the Supplier Code.

Setting both `isActive` and `supplierCodeConfirmed` as TRUE indicates that the information captured during the Registration wizard process has effectively been provided, and therefore the wizard will not run.

Setting `isActive` as FALSE and `supplierCodeConfirmed` as TRUE indicates that part of the information captured during the Registration wizard process is present, but the wizard will run to allow the user to complete the data entry.

If a Supplier is created with a status of AWAITING AUTHORISATION, but with `isActive` set as TRUE, the Supplier status is changed to REGISTERED, thus bypassing the Registration process.



Note:

When a new supplier is created manually within the application, the system sends an email to the supplier requesting they complete the registration process. When creating a new supplier through the API, the email is not generated.

Where the record is linked to another record, such as the Supplier Type in this case, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a `SupplierLink` root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Supplier record's internal ID
recordLink	string	URI to the newly created Supplier record, for use in a GET request
code	string	Supplier code business key to the newly created Supplier record
name	string	Supplier's name
localName	string	Supplier's name in business language (if used)

Error Messages

If the supplied data does not result in a valid Supplier (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Element	Message	Meaning
billingCode	ERROR: class <<hostname>>.Supplier.billingCode - The condition is invalid	No tag provided
billingCode	HTTP 417 IllegalStateException: Cannot locate keyword for class <<hostname>>.BillingCode using code:null	No <ns1:code> tag
billingCode	HTTP 417 IllegalStateException: Cannot locate keyword for class <<hostname>>..BillingCode using code:<###>	Invalid <ns1:code>
businessUnit	ERROR: class <<hostname>>..Supplier.businessUnit - The condition is invalid	No tag provided
businessUnit	HTTP 417 IllegalStateException: Cannot locate keyword for class <<hostname>>..domain.BusinessUnit using code:null	No <ns1:code> tag
businessUnit	HTTP 417 IllegalStateException: Cannot locate keyword for class <<hostname>>..BusinessUnit using code:<###>	Invalid <ns1:code>
code	ERROR: class com.micros.creations.core.domain.Supplier.code - ??Another object exists with this value??	Supplier code already exists
email	ERROR: class com.micros.creations.core.domain.Supplier.email - The condition is invalid	No tag provided
email	ERROR: class <<hostname>>.Supplier.email - The attribute is invalid	Malformed email address
supplierContactName	ERROR: class <<hostname>>.Supplier.supplierContactName - The condition is invalid	No tag provided

Element	Message	Meaning
supplierContactName	ERROR: class <<hostname>>.Supplier.supplierContactName - The condition is invalid	No value provided
name (supplier name)	ERROR: class <<hostname>>.Supplier.name - The condition is invalid	No tag provided
name (supplier name)	ERROR: class <<hostname>>.Supplier.name - The condition is invalid	No value provided
supplierType	ERROR: class <<hostname>>.Supplier.supplierType - The condition is invalid	No tag provided
supplierType	HTTP 417 IllegalStateException: Cannot locate keyword for class <<hostname>>.SupplierType using code:null	No <ns1:code> tag
supplierType	HTTP 417 IllegalStateException: Cannot locate keyword for class <<hostname>>.SupplierType using code:<###>	Invalid <ns1:code>

Update Record

Description

Updates an existing Supplier record. Use this function to update a supplier's details in Supplier Evaluation based on data sourced from the external system.

Endpoint address: /services/rest/supplier/{id} **HTTP method:** PUT

Request Details

The body of the request contains a SupplierUpdateDTO to specify the updates to the Supplier record. Compared to retrieving a supplier (which uses the SupplierFullDTO type), this request is much shorter. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

The request content is similar to that for creating a supplier, but crucially, the links to other top-level records (Sites, SiteContact, SupplierContact) are omitted. The omission of those ensures that when updating a supplier, only the supplier details need to be specified, and not the details for the related records that may not require updating (and which should be updated to calls to their respective services). After the call, the Supplier record is updated to match the request.



Note:

When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the Supplier record (except for the Sites, SiteContacts, and SupplierContacts).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a SupplierLink element. The SupplierLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Supplier record's internal ID
recordLink	string	URI to the Supplier record, for use in a GET request
code	string	Supplier code business key to the Supplier record
name	string	Supplier's name
localName	string	Supplier's name in business language (if used)

Error Messages

If the supplied data does not result in a valid Supplier (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Element	Message	Meaning
billingCode	HTTP 417 IllegalStateException: Cannot locate keyword for class <<hostname>>.BillingCode using code:<####>	Invalid <ns1:code>
businessUnit	HTTP 417 IllegalStateException: Cannot locate keyword for class <<hostname>>.domain.BusinessUnit using code:<####>	Invalid <ns1:code>
code	MySQLIntegrityConstraintViolationException: Duplicate entry <####> for key 'c_code'	Supplier code provided that already exists for another supplier
name	PropertyValueException: not-null property references a null or transient value: <<hostname>>.domain.Supplier.name	No tag provided
status	NullPointerException:	No status tag provided
status	IllegalArgumentException: Empty string is not allowed, should be null on method: <<hostname>>.domain.Supplier@1df35f3[id=<####>,code=<####>]	No status code provided

SiteRestService

This section describes the API for managing sites. The following functions are available:

- [List of Values](#): retrieves a list of sites
- [Retrieve Record by ID](#): retrieves a Site record using its unique identifier
- [Retrieve Record by Business Key](#): retrieves a Site record using its business key

- [Check Record Modification Timestamp](#): retrieves the timestamp when a Site record was last updated
- [Create Record](#): creates a new Site record
- [Update Record](#): updates an existing Site record

**Note:**

When integrated with Merchandising, using the SiteRestAPI to make changes that affect the Billing Details table can have an impact on the integration.

List of Values

Description

Retrieves a list of sites in a paged list. Use this function to retrieve a simple list of supplier names and IDs, or to locate Site record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/site **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example	Validation
offset	Optional	int	No	No	60	
pageSize	Optional	int	No	No	30	
siteStatus	Optional	string	Yes	No	AWAITING REGISTRATIO N	
siteCode	Optional	string	Yes	Yes	A0001-0001	
siteName	Optional	string	Yes	Yes	West Road Site	
siteType	Optional	string	Yes	Yes	CANNERY	
country	Optional	string	Yes	Yes	UK	

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example	Validation
businessCategory	Optional	string	Yes	Yes	CHEESE_DAIRY	Code in the Business Categories glossary at any level of the hierarchy. Applies the filter to Business Category and Lead Business Category (if used). Returns a result if any of the values match any of the specification's business categories.
businessUnit	Optional	string	Yes	Yes	UK	
supplierName	Optional	string	Yes	Yes	ABC Ltd	
supplierActive	Optional	String	No	No	true (otherwise will return active and inactive suppliers)	
modifiedSince	Optional	string	No	No	2015-05-19 13:30:39	
modifiedUntil	Optional	string	No	No	2015-05-19 13:30:39	
statusChangedFrom	Optional	string	No	No	2015-06-10 09:00:00	If From and Until are both used, then the Until date must be greater than or equal to the From date.
statusChangedUntil	Optional	string	No	No	2015-06-10 09:00:00	
statusChangedTo	Optional	string	Yes	No	ACTIVE	
leadTechnologist	Optional	string	Yes	Yes	John Smith	The user's Login Id.

Example URLs

.../services/rest/site/?offset=2&pageSize=20 .../services/rest/site/?siteStatus=ACTIVE .../services/rest/site/?supplierName=API%

Response Details

For a successful response, XML is returned with a SiteLinkList root element containing an entries element for each matched site. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Site record's internal ID
recordLink	string	URI to the SiteRestService Retrieve record for this site
code	string	Site code business key to the Site record
name	string	Site's name
localName	string	Site's name in business language (if used)

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

Element	Message	Meaning
siteStatus	HTTP 417 IllegalStateException: The Supplier Status <####> does not exist	Invalid site status
siteStatus	HTTP 417 IllegalStateException: The Supplier Statuses <####~####~####> do not exist	Invalid site status
siteType	HTTP 417 IllegalStateException: The Supplier Type <####> does not exist	Invalid supplier type
siteType	HTTP 417 IllegalStateException: The Supplier Types <####~####> do not exist	Invalid supplier type
country	HTTP 417 IllegalStateException: The Country <####> does not exist	Invalid country
country	HTTP 417 IllegalStateException: The Countries <####~####> do not exist	Invalid country
businessCategory	HTTP 417 IllegalStateException: The Business Category <####> does not exist	Invalid business category
businessCategory	HTTP 417 IllegalStateException: The Business Categories <####~####> do not exist	Invalid business category
businessUnit	HTTP 417 IllegalStateException: The Business Unit <####> does not exist	Invalid business unit
businessUnit	HTTP 417 IllegalStateException: The Business Units <####~####> do not exist	Invalid business unit
modifiedSince	Invalid dates cause stack trace to come back to "response"	
modifiedUntil	Invalid dates cause stack trace to come back to "response"	
pageSize	Error Code: INVALIDRESTSERVICEPAGESIZE INVALIDPAGESIZE	Page Size must be between 1 and 100
offset	Error Code: INVALIDOFFSET	Offset must be a positive integer
siteStatus	Error Code: INVALIDSITESTATUS INVALIDSITESTATUSSES	Invalid Site Status
siteType	Error Code: INVALIDSITETYPE	Invalid Site Type

Element	Message	Meaning
country	Error Code: INVALIDCOUNTRY	Invalid Country
businessCategory	Error Code: INVALIDBUSINESSCATEGORY	Invalid Business Category
businessUnit	Error Code: INVALIDBUSINESSUNIT	Invalid Business Unit
supplierActive	Error Code: INVALIDBOOLEAN	Invalid supplierActive value Permitted values (not case-sensitive) are: true; yes; 1; false; no; 0
leadTechnologist	Error Code: INVALIDUSER	Invalid Lead Technologist
statusChangedFrom	Error Code: INVALIDDATEFORMAT	Invalid Status Changed From Must be a valid date/time format: Year-Month-DayOfMonth Hour:Minute:Seconds such as 2016-12-30 09:59:59
statusChangedUntil	Error Code: INVALIDDATEFORMAT	Invalid Status Changed Until Must be a valid date/time format: Year-Month-DayOfMonth Hour:Minute:Seconds such as 2016-12-30 09:59:59
statusChangedTo	Error Code: INVALIDSTATUSCHANGEDTO	Either Status Changed From or Status Changed Until must be specified if filtering on Status Changed To

Retrieve Record by ID

Description

Retrieves a Site record's details using the record's internal unique ID (which is not visible in the UI). Use this function to retrieve the full details of an individual site.

Endpoint address: /services/rest/site/{id} **HTTP method:** GET

Request Details

The URL contains the {id} parameter that determines the record to retrieve.

URI Parameters

Parameter Name	Mandatory/Optional	Value Type	Multiple Value Separator (~) Supported?	Example
locale	Optional	string	Yes	en_GB~fr Locale country code. If used, fields that have language translations have a localeData element returned for the specified locales if data is present.

Example URL

.../services/rest/site/87

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of a siteFullDTO root element containing the individual attributes of the requested Site record. If an ID is not specified, a list of all sites is returned (per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 IllegalStateException: Cannot find the Site to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key

Description

Retrieves a single Site record's details using its business key (supplier and codes). Use this function to retrieve the full details of an individual site using the combination of the Supplier Evaluation supplier and site codes.

Endpoint address: /services/rest/site/byKey/{supplierCode}/{siteCode} **HTTP method:** GET

Request Details

There are no request parameters, but the URL contains the {supplierCode} and {siteCode} parameters that determine the record to retrieve.

Example URL

.../services/rest/site/byKey/A0001/A0001-0001

Response Details

If successful, an HTTP 301 redirect response is sent containing a header Location with the URI of the document requested.

The returned URI represents the call to be made to the Retrieve Record by ID function (with the id substituted with the relevant value), which will return a siteFullDTO root element containing the individual attributes of the requested Site record.

**Note:**

If your method of calling the API is configured to automatically follow HTTP 301 redirects, the call may automatically return a siteFullDTO root element containing the individual attributes of the requested Site record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
supplierCode	HTTP 404 Not Found	Invalid {supplierCode} - blank
siteCode	HTTP 404 Not Found	Invalid {siteCode}

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a Site record. Use this function to determine when a site's details were last updated.

Endpoint address: /services/rest/site/{id} **HTTP method:** HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/site/87

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header showing the date and time of the last update of the requested Site record.

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 Expectation Failed	Invalid {id} - not found

Create Record

Description

Creates a new Site record. Use this function to create new sites in Supplier Evaluation based on data sourced from the external system.

Dependencies: The Supplier must be present in the application and its record ID obtained. If assigning a Business Category or Technologist, the record must be present in the application and its record ID obtained. Create Contacts after creating the Site. For more information, see [Dependencies](#).

Endpoint address: /services/rest/site **HTTP method:** POST

Request Details

The body of the request contains a SiteFullDTO element to specifying the details of the site to create. Compared to retrieving a site (which uses the same SiteFullDTO type), this request is much shorter. Only the attributes that are to be populated on the created Site record need to be included. As a minimum, this must include the fields shown in the following table.

Site Mandatory Fields

Field Name	Element Name
Site Name	name
Site Type	siteType/code
Categories	businessCategories
Lead Technologist	leadTechnicalManager
Supplier	supplier/code
Site Status	siteStatus
Deleted Record	deleted (Set to false)

Example Request XML

This example shows the minimum requirement to be able to create a site against a supplier.

```
<ns1:siteFullDTO
  xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple"
  xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full">

  <ns0:businessCategories>
    <ns1:code>CATEGORY2A</ns1:code>
  </ns0:businessCategories>
  <ns0:leadTechnicalManager>
    <ns1:code>techadmin</ns1:code>
  </ns0:leadTechnicalManager>

  <ns0:name>Site Name</ns0:name>
  <ns0:siteType>
    <ns1:code>SITE_TYPE_EXAMPLE</ns1:code>
  </ns0:siteType>
  <ns0:siteStatus>
    <ns1:status>ACTIVE</ns1:status>
  </ns0:siteStatus>
  <ns0:supplier>
```

```

        <ns1:code>WS0001</ns1:code>
      </ns0:supplier>
      <ns0:deleted>>false</ns0:deleted>

</ns1:siteFullDTO>

```

Set the deleted attribute to false to ensure the record can be seen in the UI. If omitted or set to true, it will exist, but will not be visible.

Where the record is linked to another record, such as the Site Type in this case, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a SiteLink root element containing the site data, and a SupplierLink element, consisting of the parent supplier data. The returned elements are shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Site record's internal ID
recordLink	string	URI to the newly created Site record, for use in a GET request
code	string	Supplier and site code business key to the newly created Site record
name	string	Site's name
localName	string	Site's name in business language (if used)
supplierLink/recordId	long	Newly created site's Supplier record internal ID
supplierLink/recordLink	string	URI to the newly created site's Supplier record, for use in a GET request
recordLink/code	string	Supplier code business key to the newly created site's Supplier record
recordLink/name	string	Supplier's name
recordLink/localName	string	Supplier's name in business language (if used)

Error Messages

If the supplied data does not result in a valid Site (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Element	Message	Meaning
supplier/id	Null Pointer Exception	Supplier ID not provided
supplier/id	No row with the given identifier exists: [com.micros.creations.core.domain.Suppli er#<###>]	Supplier not found

Element	Message	Meaning
siteStatus	Null Pointer Exception	Site Status not provided
siteStatus	Cannot locate keyword for class com.micros.creations.core.domain.SiteStatus using status:<###>	Site Status not found
siteType	ERROR: class com.micros.creations.core.domain.Site.siteType - The condition is invalid	Site Type not provided
siteType	Cannot locate keyword for class com.micros.creations.core.domain.SiteType using code:<###>	Site Type not found
name	class com.micros.creations.core.domain.Site.name - The condition is invalid	Site name not provided
leadTechnicalManager	ERROR: class com.micros.creations.core.domain.Site.leadTechnicalManager - The condition is invalid	Lead Technical Manager not provided
leadTechnicalManager	HTTP 417 IllegalStateException: Cannot locate keyword for class com.micros.creations.core.domain.User using code:<###>	Lead Technical Manager not found
businessCategories	ERROR: class com.micros.creations.core.domain.Site.businessCategories - The condition is invalid	Business Category not provided
businessCategories	Cannot locate keyword for class com.micros.creations.core.domain.BusinessCategory using code:<###>	Business Category not found
code	Duplicate site code	The site already exists for this supplier

Update Record

Description

Updates an existing Site record. Use this function to update a site's details in Supplier Evaluation based on data sourced from the external system.

Endpoint address: /services/rest/site/{id} **HTTP method:** PUT

Request Details

The body of the request contains a SiteUpdatedDTO to specify the updates to the Site record. Compared to retrieving a site (which uses the SiteFullDTO type), this request is much shorter. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

The request content is similar to that for creating a site, but crucially, the links to other top-level records (SiteContact) are omitted. The omission of those ensures that when updating a site, only the site details need to be specified, and not the details for the related records that may not require updating (and which should be updated with calls to their respective services). After the call, the Site record is updated to match the request

**Note:**

When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the Site record (except for the SiteContacts).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a SiteLink root element containing the site data, and a SupplierLink element, consisting of the parent supplier data. The returned elements are shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Updated Site record's internal ID
recordLink	string	URI to the updated Site record, for use in a GET request
code	string	Site code business key to the updated Site record
name	string	Site's name
localName	string	Site's name in business language (if used)
supplierLink/recordId	long	Updated site's Supplier record internal ID
supplierLink/recordLink	string	URI to the updated site's Supplier record, for use in a GET request
recordLink/code	string	Supplier code business key to the updated site's Supplier record
recordLink/name	string	Supplier's name
recordLink/localName	string	Supplier's name in business language (if used)

Error Responses

If the supplied data does not result in a valid Site (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

ContactRestService

This section describes the API for managing supplier and site contacts. The following functions are available:

- [List of Values](#): retrieves a list of contacts
- [Retrieve Record by ID](#): retrieves a Contact record using its unique identifier
- [Retrieve Supplier Contact Record by Business Key](#): retrieves a Supplier Contact record using its business key

- [Retrieve Site Contact Record by Business Key](#): retrieves a Site Contact record using its business key
- [Check Record Modification Timestamp](#): retrieves the timestamp when a Contact record was last updated
- [Create Record](#): creates a new Contact record
- [Update Record](#): updates an existing Contact record

List of Values

Description

Retrieves a list of contacts in a paged list. Use this function to retrieve a simple list of contact names and IDs, or to locate Contact record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/contact **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
offset	Optional	int	No	No	60
pageSize	Optional	int	No	No	30
name	Optional	string	Yes	Yes	John Smith
supplierContactRoles	Optional	string	Yes	Yes	HEAD OF MANUFACTURING
siteContactRoles	Optional	string	Yes	Yes	HEAD OF MANUFACTURING
country	Optional	string	Yes	Yes	UK
selectedSites	Optional	string	No	No	A0001-0001
supplierCode	Optional	string	Yes	Yes	A0001
supplierName	Optional	string	Yes	Yes	ABC Ltd.
siteStatus	Optional	string	Yes	No	ACTIVE~INACTIVE
supplierActive	Optional	string	No	No	true returns active contacts no value returns both active and inactive contacts
modifiedSince	Optional	string	No	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	No	2015-05-19 13:30:39

Example URLs

.../services/rest/contact/?offset=2&pageSize=20 .../services/rest/contact/?siteStatus=ACTIVE
.../services/rest/contact/?supplierName=API%

Response Details

For a successful response, XML is returned with a ContactLinkList root element containing an entries element for each matched contact. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Contact record's internal ID
recordLink	string	URI to the ContactRestService Retrieve record service for this contact
email	string	Contact's email address
name	string	Contact's name
siteContact	Boolean	Indicates if a contact for the site (true), else false
supplierContact	Boolean	Indicates if a contact for the supplier (true), else false

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

Element	Message	Meaning
supplierContactRoles	HTTP 417 IllegalStateException: The Supplier Contact Role <###> does not exist	Invalid supplier contact role
supplierContactRoles	HTTP 417 IllegalStateException: The Supplier Contact Roles <###~###> do not exist	Invalid supplier contact role
siteContactRoles	HTTP 417 IllegalStateException: The Site Contact Role <###> does not exist	Invalid site contact role
siteContactRoles	HTTP 417 IllegalStateException: The Site Contact Roles <###~###> do not exist	Invalid site contact role
country	HTTP 417 IllegalStateException: The Country <###> does not exist	Invalid country code
country	HTTP 417 IllegalStateException: The Countries <###~###> do not exist	Invalid country code
siteStatus	HTTP 417 IllegalStateException: The Supplier Status <###> does not exist	Invalid supplier status code
pageSize	Error Code: INVALIDRESTSERVICEPAGESIZE INVALIDPAGESIZE	Page Size must be between 1 and 100
offset	Error Code: INVALIDOFFSET	Offset must be a positive integer
supplierContactRole	Error Code: INVALIDCONTACTROLE	Invalid Supplier Contact Role
siteContactRole	Error Code: INVALIDCONTACTROLE	Invalid Site Contact Role
country	Error Code: INVALIDCOUNTRY	Invalid Country
selectedSite	Error Code: INVALIDSITE	Invalid Site
supplierCode	Error Code: INVALIDSUPPLIER	Invalid Supplier

Element	Message	Meaning
siteStatus	Error Code: INVALIDSITESTATUS	Invalid Site Status
supplierActive	Error Code: INVALIDBOOLEAN	Invalid supplierActive value Permitted values (not case-sensitive) are: true; yes; 1; false; no; 0

Retrieve Record by ID

Description

Retrieves a single Contact record's details using the record's unique ID. Use this function to retrieve the full details of an individual supplier or site contact.

Endpoint address: /services/rest/contact/{id} **HTTP method:** GET

Request Details

The URL contains the {id} parameter that determines the record to retrieve.

URI Parameters

Parameter Name	Mandatory/Optional	Value Type	Multiple Value Separator (~) Supported?	Example
locale	Optional	string	Yes	en_GB~fr Locale country code. If used, fields that have language translations have a localeData element returned for the specified locales if data is present.

Example URL

.../services/rest/contact/405

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of a contactAndPersonDTO root element containing the individual attributes of the requested Contact record and the person it associates to.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 IllegalStateException: Cannot find the Contact to return with id:<###>	Invalid {id} - not found

Retrieve Supplier Contact Record by Business Key

Description

Retrieves a single Supplier Contact record's details using its business key (supplier code, contact name, and email address). Use this function to retrieve the full details of an individual supplier contact using the combination of the Supplier Evaluation supplier codes, name of the contact, and the contact's email address.

Endpoint address: /services/rest/contact/byKey/supplier/{supplierCode}/{name}/{email} **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/Optional	Value Type	Multiple Value Separator (~) Supported?	Example
supplierCode	Mandatory	string	No	A0001
name	Mandatory	string	No	John Doe
email	Mandatory	string	No	john.doe@email.com

Example URL

.../services/rest/contact/byKey/supplier/A0001/John Doe/john.doe@email.com

Response Details

If successful, an HTTP 301 redirect response is sent containing a header Location with the URI of the document requested.

The returned URI represents the call to be made to the Retrieve Record by ID function (with the id substituted with the relevant value), which will return a contactAndPersonDTO root element containing the individual attributes of the requested Contact record.



Note:

If your method of calling the API is configured to automatically follow HTTP 301 redirects, the call may automatically return a contactAndPersonDTO root element containing the individual attributes of the requested Contact record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Responses

In the document cannot be located, an HTTP 404 response is sent.

Retrieve Site Contact Record by Business Key

Description

Retrieves a single Site Contact record's details using its business key (supplier code, site code, contact name, and contact email). Use this function to retrieve the full details of an individual site contact using the combination of the Supplier Evaluation supplier and site codes, name of the contact, and the contact's email address.

Endpoint address: /services/rest/contact/byKey/site/{supplierCode}/{siteCode}/{name}/{email}
HTTP method: GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
supplierCode	Mandatory	string	No	A0001
siteCode	Mandatory	string	No	A0001-0001
name	Mandatory	string	No	John Doe
email	Mandatory	string	No	john.doe@email.com

Example URL

.../services/rest/contact/byKey/site/A0001/A0001-0001/ John Doe/john.doe@email.com

Response Details

For a successful response, XML is returned with a contactAndPersonDTO root element containing the individual attributes of the requested Contact record and the person it associates to.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Responses

In the document cannot be located, an HTTP 404 response is sent.

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a Contact record. Use this function to determine when a supplier or site contact's details were last updated.

Endpoint address: /services/rest/contact/{id} **HTTP method:** HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/contact/405

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header to show the last modification date and time of the last update of the requested Contact record.

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 Expectation Failed	Invalid {id} - not found

Create Record

Description

Creates a new Contact record (and the associated Person record). Use this function to create new supplier or site contacts in Supplier Evaluation based on data sourced from the external system.

Dependencies: The Supplier (and Site, if a site contact) must be present in the application and its record ID obtained. Omit the Site element if not a site contact. The User/Person must be present in the application and its record ID obtained. For more information, see [Dependencies](#).

Endpoint address: /services/rest/contact **HTTP method:** POST

Request Details

The body of the request contains a ContactAndPersonDTO to specify the details of the contact to create. Compared to the retrieving a contact (which uses the same ContactAndPersonDTO type), this request is much shorter. Only the attributes that are to be populated on the created Contact record need to be included. As a minimum, this must include the fields shown in the following table.

Contact Mandatory Fields

Field Name	Element Name
Name	contactFullDTO/person/name personFullDTO/name

Field Name	Element Name
User Id	contactFullDTO/person/id
Supplier	contactFullDTO/person/supplier/id contactFullDTO/company/code personFullDTO/supplier/code personFullDTO/supplier/id
Email	contactFullDTO/person/email personFullDTO/email
Phone	contactFullDTO/contactDetails/phoneNumber
Supplier Contact Roles	contactFullDTO/supplierContactRole/code
Contact Type	contactFullDTO/dtype
Site Contact Flag	contactFullDTO/siteContact
Supplier Contact Flag	contactFullDTO/supplierContact
Deleted Record	deleted (Set to false)

Example Request XML

This example shows the minimum requirement to be able to create a contact against a supplier.

```
<ns0:ContactAndPersonDTO xmlns:ns4="http://www.oracle.com/orsecs/service/rest/model"
xmlns:ns0="http://www.micros.com/creations/core/domain/dto/vlp0/full" xmlns:ns1="http://
www.micros.com/creations/core/domain/dto/vlp0/simple">
  <ns0:contactFullDTO>
    <ns0:contactDetails>
      <ns0:phoneNumber>01234 56789</ns0:phoneNumber>
    </ns0:contactDetails>
    <ns0:dtype>SupplierContact</ns0:dtype>
    <ns0:person>
      <ns1:email>jphillips@example.com</ns1:email>
      <ns1:name>Jane Phillips</ns1:name>
      <ns1:supplier>
        <ns1:code>A0001</ns1:code>
      </ns1:supplier>
    </ns0:person>
    <ns0:siteContact>false</ns0:siteContact>
    <ns0:supplierContact>true</ns0:supplierContact>
    <ns0:supplierContactRole>
      <ns1:code>ACCOUNT MANAGER</ns1:code>
    </ns0:supplierContactRole>
    <ns0:company>
      <ns1:code>A0001</ns1:code>
    </ns0:company>
  </ns0:contactFullDTO>
  <ns0:personFullDTO>
    <ns0:email>jphillips@example.com</ns0:email>
    <ns0:name>Jane Phillips</ns0:name>
    <ns0:supplier>
      <ns1:code>A0001</ns1:code>
    </ns0:supplier>
  </ns0:personFullDTO>
  <ns0:deleted>>false</ns0:deleted>
</ns0:ContactAndPersonDTO>
```


Set the deleted attribute to false to ensure the record can be seen in the UI. If omitted or set to true, it will exist, but will not be visible.

If a contact has multiple supplier or site contact roles, they should be separate entries as follows:

```
<ns0:supplierContactRole>
  <ns1:code>ACCOUNT MANAGER</ns1:code>
</ns0:supplierContactRole>
<ns0:supplierContactRole>
  <ns1:code>AUDITS AND VISITS CONTACT</ns1:code>
</ns0:supplierContactRole>
<ns0:supplierContactRole>
  <ns1:code>EMERGENCY CONTACT</ns1:code>
</ns0:supplierContactRole>
```

Where the record is linked to another record, such as the Contact Role in this case, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a ContactLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Contact record's internal ID
recordLink	string	URI to the newly created Contact record, for use in a GET request
email	string	Email address of the newly created contact
name	string	Contact's name
siteContact	Boolean	Indicates if a contact for the site (true), else false
supplierContact	Boolean	Indicates if a contact for the supplier (true), else false

Error Messages

Element	Message	Meaning
phoneNumber	phoneNumber - The condition is invalid	Missing tag
dtype	IllegalArgumentException: Invalid value for dtype:<###>	Invalid value SupplierContact; SiteContact
name & email	HTTP 417 IllegalStateException: Cannot locate Person record with name: <###> and email: <###>	A valid user must be found with matching name and email address
company	HTTP 417 IllegalStateException: Cannot locate keyword for class com.micros.creations.core.domain.Company using code:<###>	Invalid company/supplier code

Update Record

Description

Updates an existing Contact record. Use this function to update a supplier or site contact's details in Supplier Evaluation based on data sourced from the external system.

Endpoint address: /services/rest/contact/{id} **HTTP method:** PUT

Request Details

The body of the request contains a ContactAndPersonDTO element to specify the updates to the Contact record. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

The request content is the same as that for creating a contact. After the call, the Contact record is updated to match the request.



Note:

When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the Contact record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a ContactLink element. The ContactLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Contact record's internal ID
recordLink	string	URI to the Contact record, for use in a GET request
email	string	Contact's email address
name	string	Contact's name
siteContact	Boolean	Indicates if a contact for the site (true), else false
supplierContact	Boolean	Indicates if a contact for the supplier (true), else false

Error Messages

If the supplied data does not result in a valid Contact (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

TaskRestService

This section describes the API for managing user tasks. The following function is available:

- [List of Tasks](#): Retrieves a list of user tasks

List of Tasks

Description

Retrieves a list of tasks for a user in the given language. Use this function to retrieve a list of a specific user's entries in their Supplier Evaluation Task Manager app. Parameters are used to specify the name of the user, and the language of the returned task details.

Endpoint address: /services/rest/task **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
userId	Mandatory	string	No	No	John Smith
language	Mandatory	string	No	No	en_GB

Example URL

.../services/task/?language=en_GB

The userId parameter is the login ID of the user for which the tasks list is to be retrieved. The language parameter is the code of the language record/locale in which to retrieve the task details.

Response Details

For a successful response, XML is returned with a TaskDTOList root element containing a tasks element for each matched task. The root element consists of the elements in the following table.

Returned Elements

Element	Type	Description
message	string	Name of the task, in the language specified
messageId	string	A language-agnostic identifier for the task
myCreationsLink	string	A URI to the Supplier Evaluation system, which will open the list of items for that task
taskItemCount	int	The number of items in the task

Error Messages

Element	Message	Meaning
userId	userId required	User not provided
userId	Cannot find user with login id <###>	User not found
language	language required	Language not provided
language	Invalid locale format: <###>	Language not found

UrgentItemsRestService

This section describes the API for retrieving a count of a user's urgent items. The following function is available:

- [Number of Urgent Items](#): retrieves the number of urgent items pending for a user

Number of Urgent Items

Description

Retrieves a count of the number of Urgent Item tasks for a user. Use this function to determine the number of pending urgent tasks a specific user has in their Supplier Evaluation Urgent Item Manager app. A parameter is passed to specify the name of the user.

Endpoint address: /services/rest/urgentItems **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
userId	Mandatory	string	No	No	John Smith

The userId parameter is the login ID of the user for which the number of pending urgent items are to be retrieved.

Response Details

For a successful response, XML is returned with an UrgentItemsModel root element containing an itemCount element that specifies the number of Urgent Items.

Error Messages

Element	Message	Meaning
userId	User not provided: userId required	User not provided
userId	Cannot find user with login id <###>	User not found

ArtworkRestService

This section describes the API for Artwork integration. The following functions are available:

- **Started Activities:** retrieves a list of Artwork Activities that have started
- **Update Record:** updates existing Artwork Activities

Started Activities

Description

Retrieves a list of Project Activities that have their useMyArtwork flag set, and their status has changed to Started within the specified date range. This function is used for the integration of Supplier Evaluation with an external Artwork Management system.

Endpoint address: /services/rest/artwork/started **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
offset	Optional	int	No	No	60
pageSize	Optional	int	No	No	30
fromDate	Mandatory	string	No	No	2014-12-30 23:59:59
toDate	Mandatory	string	No	No	2015-12-30 23:59:59

Response Details

For a successful response, XML is returned with a CreateArtworkRequestList root element containing an entries element for each matching Artwork project activity. A totalRecords element identifies the number of records returned. Separate activityDetails and project elements group the returned activity and project data. The elements consist of the elements in the following tables.

Returned Elements - activityDetails

Element	Type	Description
activityName	string	Project activity's title
activityRecordId	long	Project activity record's internal ID
actualStartDate	string	Actual start date of the project activity
checkpointOnly	Boolean	Is the project activity a checkpoint only type
criticalPath	Boolean	Is the project activity a critical path type
duration	int	Duration of the project activity

Element	Type	Description
isAutoGate	string	Project's type of auto gate
isGate	Boolean	Is the project activity a gate type
isKey	Boolean	Is the project activity a key type
projectId	long	Project record's internal ID
projectTitle	string	Project's title
proposedEndDate	string	Proposed end date of the project activity
proposedStartDate	string	Proposed start date of the project activity
responsibleUserRoles	string	Role names of the responsible users
sequenceNumber	string	Project activity's sequence number
statusCode	string	Code of the project activity's status
subStatusCode	string	Code of the project activity's sub-status

Returned Elements - projects

Element	Type	Description
brand	string	Name of the project's brand
categories	string	Names of the project's categories (the lowest level of the selected category - may be multiples)
masterProject	Boolean	If the project is a master type
newCategoryComments	string	Category comments
packaging	string	Packaging details
productRangeInformation	string	Project's product range details
projectConcept	string	Project's business key ID
projectId	string	Project's business key ID
projectManager	string	Name of the project's project manager
projectRecordId	long	Project record's internal ID
projectTitle	string	Project's title
projectType	string	Type of project
proposedProducts	string	Project's proposed products
specType	string	Type of project's product specification
status	string	Status of the project
subBrand	string	Name of the project's sub-brand
supplier	string	Name of the supplier associated to the project
site	string	Name of the site associated with the project
targetConsumer	string	Project's target consumer details
teamDetails/role	string	Name of the team role
teamDetails/users	string	Name of the team users
templateFolder	string	Name of the project template folder

Element	Type	Description
templateType	string	Type of the project's template
templateUsed	string	Name of the project's template

Error Messages

Element	Message	Meaning
fromDate	fromDate parameter required	From date not provided
toDate	toDate parameter required	To date not provided
fromDate	fromDate and toDate parameters are required	From and To dates not provided
toDate	From date must be before or equal to To date	From date is after To date
fromDate	HTTP 417 <errorMessage>	Not a valid date, whereby <errorMessage> will identify the issue For example: IllegalInstantException: Cannot parse "2018-03-25 01:00:00": Illegal instant due to time zone offset transition (Europe/London) In this example, due to the clocks In this example, due to the clocks moving forward (in the server's time-zone), the passed in date is invalid
toDate	HTTP 417 <errorMessage>	Not a valid date, whereby <errorMessage> will identify the issue (see example above)

Update Record

Description

Updates the sub-status of existing Artwork Project Activities. This function is used for the integration of Supplier Evaluation with an external Artwork Management system, to update the status of an Artwork activity from the external system.

Endpoint address: /services/rest/artwork/update **HTTP method:** POST

Request Details

The body of the request contains an activityUpdateRequest element consisting of the elements shown in the following table.

Request Elements

Element	Type	Description
activityName	string	Project activity's name
activityRecordId	long	Project activity record's internal ID
projectId	string	ID business key to the Project record
projectRecordId	long	Project record's internal ID

Element	Type	Description
subStatusCode	string	Code business key to the activity's sub-status

After the call, the Project Activity record's sub-status is updated to match the request.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 3 - Process* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing an ArtworkActivityLink element. The ArtworkActivityLink element consists of the elements shown in the following table.

Returned Elements

If successful, an HTTP 200 response is sent.

Element	Type	Description
recordId	long	Project activity's internal ID
recordLink	string	URI to the Product Activity record, for use in a GET request
code	string	Code business key to the Project Activity record
title	string	Project activity's title

Error Messages

In the event that an error occurs, an HTTP 500 response is sent.

The following conditions may prevent a successful update:

- An invalid project record id or activity record id.
- The activity sub-statuses not being configured in Supplier Evaluation (including the general sub-statuses COMPLETED, MESSAGE SENT, MESSAGE ACCEPTED, and MESSAGE FAILED).
- The activity's template not being flagged as an artwork type, or having the permitted activity sub-statuses assigned, prior to the activity being created (a snapshot of the template settings is stored on the Activity record on creation).

BusinessCategoryService

This section describes the API for managing Business Categories. The following functions are available:

- [List of Values](#): retrieves a list of categories
- [Retrieve Record by ID](#): retrieves a Business Category record using its unique identifier
- [Check Record Modification Timestamp](#): retrieves the timestamp when a Business Category record was last updated
- [Create Record](#): creates a new Business Category record
- [Update Record](#): updates an existing Business Category record

- **Delete Record:** deletes an existing Business Category record

List of Values

Description

Retrieves a list of categories in a paged list. Use this function to retrieve a simple list of categories and IDs, or to locate Business Category record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/businessCategory **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
offset	Optional	int	No	No	60
pageSize	Optional	int	No	No	30
code	Optional	string	Yes	Yes	POTATOES_LOOSE
parentCode	Optional	string	Yes	Yes	POTATOES
entityDescription	Optional	string	Yes	Yes	Loose Potatoes
specificationType	Optional	string	Yes	No	FOOD~PRODUCE
topLevelCategory	Optional	Boolean	No	No	TRUE
modifiedSince	Optional	string	No	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	No	2015-05-19 13:30:39

Example URLs

.../services/rest/businessCategory/?offset=2&pageSize=20 .../services/rest/
businessCategory/?code=POTATOES_LOOSE .../services/rest/businessCategory/?
entityDescription=Potato%

The code parameter will attempt to find a single category with a matching code. The parentCode parameter will find any categories whose parent has the given code.

The entityDescription matches any category whose description (irrespective of language) matches the value given; this is just the description of the category in question and does not include those of parents.

The specificationType parameter is a tilde (~) separated list of types.

**Note:**

Including a value in the specificationType filter will exclude any categories for which no specification type has been selected.

The topLevelCategory parameter determines whether only top-level categories (those with no parent) will be included in the results.

Response Details

For a successful response, XML is returned with a BusinessCategoryLinkList root element containing an entries element for each matched category. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Business Category record's internal ID
recordLink	string	URI to the BusinessCategoryService Retrieve record service for this category
code	string	Code business key to the Business Category record
entityDescription	string	Category's name
path	string	Category's full path (including parent levels - separated by '/')

Error Messages

Element	Message	Meaning
specificationType	<###> is not a valid Specification Type	Specification type not found
pageSize	Error Code: INVALIDPAGESIZE	Page Size must be between 1 and 100
offset	Error Code: INVALIDOFFSET	Offset must be a positive integer
code	Error Code: INVALIDBUSINESSCATEGORY	Invalid Business Category
parentCode	Error Code: INVALIDBUSINESSCATEGORY	Invalid Business Category
specificationType	Error Code: INVALIDSPECIFICATIONTYPE	Invalid SpecificationType
topLevelCategory	Error Code: INVALIDBOOLEAN	Invalid topLevelCategory value Permitted values (not case-sensitive) are: true; yes; 1; false; no; 0

Retrieve Record by ID

Description

Retrieves a single Business Category record's details using the record's unique ID. Use this function to retrieve the full details of an individual category.

Endpoint address: /services/rest/businessCategory/{id} **HTTP method:** GET

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/businessCategory/105

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of a BusinessCategoryFullDTO root element containing the individual attributes of the requested Business Category record. If an ID is not specified, a list of all categories is returned (per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not found	Invalid {id} - not numeric
id	Invalid record id	Invalid {id} - not found

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a Business Category record. Use this function to determine when a category's details were last updated.

Endpoint address: /services/rest/businessCategory/{id} **HTTP method:** HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/businessCategory/105

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header showing the date and time of the last update of the requested Business Category record. For example:

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 417 Expectation Failed	Invalid {id} - not found
id	HTTP 404 Not found	Invalid {id} - not numeric

Create Record

Description

Creates a new Business Category record. Use this function to create a new category in Supplier Evaluation based on data sourced from the external system.

Dependencies: If a lower-level category, the parent Business Category must be present in the application and its record ID obtained. For more information, see [Dependencies](#).

Endpoint address: /services/rest/businessCategory **HTTP method:** POST

Request Details

The body of the request contains a BusinessCategoryFullDTO to specify the details of the category to create. Compared to retrieving a user (which uses the same BusinessCategoryFullDTO type), this request is much shorter. Only the attributes that are to be populated on the created Business Category record need to be included. As a minimum, this must include the fields shown in the following table:

Business Category Mandatory Fields

Field Name	Element Name
Code	code
Description	description
Parent Code	parentCode
Parent or Child level	topLevelCategory
Deleted Record	deleted (Set to false)
Path	path (The category's full path descriptions, concatenated and separated by /)



Note:

If creating a top level category, topLevelCategory must be set to true; if creating a lower level category, topLevelCategory must be set to false, and parentCode becomes mandatory (the parent category's code).

Example Request XML

```
<ns0:businessCategoryFullDTO xmlns:ns0="http://www.micros.com/creations/core/domain/dto/vlp0/full" xmlns:ns1="http://www.micros.com/creations/core/domain/dto/vlp0/simple">
  <ns0:code>X1</ns0:code>
  <ns0:localeData>
    <ns0:description>Category X - Level 1</ns0:description>
  </ns0:localeData>
</ns0:businessCategoryFullDTO>
```

```

    <ns0:path>Category X - Level 1</ns0:path>
  </ns0:localeData>
  <ns0:topLevelCategory>true</ns0:topLevelCategory>
  <ns0:deleted>false</ns0:deleted>
</ns0:businessCategoryFullDTO>

```

Set the deleted attribute to false to ensure the record can be seen in the UI. If omitted or set to true, it will exist, but will not be visible.

The following example adds a second level category to the above top-level parent category:

```

<ns0:businessCategoryFullDTO xmlns:ns0="http://www.micros.com/creations/core/domain/dto/vlp0/full" xmlns:ns1="http://www.micros.com/creations/core/domain/dto/vlp0/simple">
  <ns0:code>X2</ns0:code>
  <ns0:localeData>
    <ns0:description>Category X - Level 2</ns0:description>
    <ns0:path>Category X - Level 1 / Category X - Level 2</ns0:path>
  </ns0:localeData>
  <ns0:parentCode>X1</ns0:parentCode>
  <ns0:topLevelCategory>false</ns0:topLevelCategory>
  <ns0:deleted>false</ns0:deleted>
</ns0:businessCategoryFullDTO>

```

The value of the code element must be unique.

The description and path elements are locale-dependent, so are contained within a localeData element, with a locale element supported language code, for example:

```

<ns0:localeData>
  <ns0:description>Cheese</ns0:description>
  <ns0:locale>en_GB</ns0:locale>
</ns0:localeData>

<ns0:localeData>
  <ns0:description>Fromage</ns0:description>
  <ns0:locale>fr</ns0:locale>
</ns0:localeData>

```

The hierarchy of category levels is maintained through the parent code, linking a category to its immediate parent. All levels except the top level must therefore have a valid parent code specified.

The number of levels in the hierarchy is defined in the Supplier Evaluation Admin area.

Categories may be assigned to specific product specifications, or available for use by all; if specified, the specification type must be match those assigned to its parent categories.

Where the record is linked to another record, such as the Locale in this case, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

For a successful response, an HTTP 200 response is sent with a body containing a BusinessCategoryLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Business Category record's internal ID
recordLink	string	URI to the newly created Business Category record, for use in a GET request
code	string	Code ID business key to the newly created Business Category record
entityDescription	string	Category's description

Error Responses

If the supplied data does not result in a valid Business Category (such as a missing mandatory field), an HTTP 417 response is sent with an XML body message stating the validation errors. The request should not be reattempted with the same content.

Error Messages

Element	Message	Meaning
code	Code {0} has already been used	Codes must be unique
parent	The Parent Code cannot be blank when the Top Level Category flag is false	A parent code must be specified unless the category is a top level category
topLevelCategory	The Top Level Category flag cannot be true when the Parent Code has a value	A category cannot be a top level category and have a parent
description	Description must be provided	Descriptions are mandatory
code	The parent Business Category with code {0} cannot be found	The parent category must exist
parent	A new Business Category cannot be added to the Business Category with the code{0} because it will fall outside of the Business Category Hierarchy	Categories can only be added within the bounds of the Business Category Hierarchy as defined in Supplier Evaluation's Admin/System Control/ Business Category Configuration
specificationType	{0} is not a valid Specification Type(s)	The specification type or types must be valid
specificationType	The Specification Types must be a subset of the parent Business Category's Specification Types: {0}	This message is issued when no specification types are specified on a child, but there are specification types assigned to the parent It lists the specification types selected on the parent
specificationType	The Business Category Specification Types {0} are not a subset of the parent Business Category Specification Types {1}	This message is issued when the category's specification types have not all be selected on the parent The child's specification types are listed in {0} and the parent's specification types are listed in {1}

Update Record

Description

Updates an existing Business Category record. Use this function to update a category's details in Supplier Evaluation based on data sourced from the external system, or to move a category to appear beneath a different parent category.

Endpoint address: /services/rest/businessCategory/{id} **HTTP method:** PUT

Request Details

The body of the request contains a BusinessCategoryUpdateDTO to specify the updates to the Business Category record.

The request content is similar to that for creating a category, but does not include the child categories link. If children are to be managed, then a separate call is required for each child. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

After the call, the Business Category record is updated to match the request.



Note:

When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the Business Category record.

The id element is used to locate the category to update.

The hierarchy of category levels is maintained through the parent code, linking a category to its immediate parent. All levels except the top level must therefore have a valid parent code specified.

The number of levels in the hierarchy is defined in the Supplier Evaluation Admin area.

Categories may be assigned to specific product specifications, or available for use by all; if specified, the specification type must be match those assigned to its parent categories.

Categories may be moved to the same level within another part of the hierarchy. When a category is moved, its children are automatically moved.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 1 - Framework* for details of their mapping to the fields within the Supplier Evaluation UI.

The body of the request contains a BusinessCategoryUpdateDTO element to specify how the category should appear after the update.

Response Details

If successful, an HTTP 200 response is sent with a body containing a BusinessCategoryLink element. The BusinessCategoryLink element consists of the returned elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Business Category record's internal ID
recordLink	string	URI to the Business Category record, for use in a GET request
code	string	Code ID business key to the Business Category record
entityDescription	string	Category's description
path	string	Category's full path (including parent levels - separated by '/')

Error Responses

If the supplied data does not result in a valid Business Category (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Error Messages

Element	Message	Meaning
recordId	Invalid record id	The ID used to call the PUT process does not relate to an existing record
parent	The Parent Code cannot be blank when the Top Level Category flag is false	A parent code must be specified unless the category is a top level category
topLevelCategory	The Top Level Category flag cannot be true when the Parent Code has a value	A category cannot be a top level category and have a parent
code	The parent Business Category with code {0} cannot be found	The parent category must exist
code	A new Business Category cannot be added to the Business Category with the code{0} because it will fall outside of the Business Category Hierarchy	Categories can only be added within the bounds of the Business Category Hierarchy as defined in Supplier Evaluation's Admin/System Control/ Business Category Configuration
parent	Business Category cannot be moved from a parent at level {0} to a parent at level {1}	Categories can only be moved between parent categories at the same level in the hierarchy
specificationType	{0} is not a valid Specification Type(s)	The specification type(s) must be valid
specificationType	The Specification Types must be a subset of the parent Business Category's Specification Types: {0}	This message is issued when no specification types are specified on a child but there are specification types assigned to the parent It lists the specification types selected on the parent
specificationType	The Business Category Specification Types {0} are not a subset of the parent Business Category Specification Types {1}	This message is issued when the category's specification types have not all be selected on the parent The child's specification types are listed in {0} and the parent's specification types are listed in {1}

Delete Record

Description

Deletes an existing Business Category record, along with any associated child categories. Use this function to remove a single category or a category and all its lower level categories.

Endpoint address: /services/rest/businessCategory/{id} **HTTP method:** DELETE

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to delete.

Response Details

If the Business Category is deleted successfully, an HTTP 200 response is sent.

Error Responses

If a Business Category cannot be deleted due to it being referenced by another record (or a child of the category being referenced by another record), an HTTP 417 response is sent with an ErrorMessage/Message XML body containing the message: "Could not delete: with a URI link to the category."

Error Messages

Element	Message	Meaning
id	HTTP 417 Could not delete: {URI}	Category is referenced by another record
id	HTTP 404 Not found	Invalid {id} - not numeric
id	Invalid record id for deletion	Invalid {id} - not found

AuditRestService

This section describes the API for managing audit and visits. The following functions are available:

- [List of Values \(Audit\)](#): retrieves a list of audits and visits
- [Retrieve Record by ID \(Audit\)](#): retrieves an Audit or Visit record using its unique identifier
- [Retrieve Record by Business Key \(Audit\)](#): retrieves an Audit or Visit record using its business key
- [Check Record Modification Timestamp \(Audit\)](#): retrieves the timestamp when an Audit or Visit record was last updated
- [Create Record \(Audit\)](#): creates a new Audit or Visit record
- [Update Record \(Audit\)](#): updates an existing Audit or Visit record
- [List of Values \(Checklist\)](#): retrieves a list of checklists
- [Retrieve Record by ID \(Checklist\)](#): retrieves a Checklist record using its unique identifier
- [Retrieve Record by Business Key \(Checklist\)](#): retrieves a Checklist record using its business key

- [Create Record \(Checklist\)](#): creates a new Checklist record
- [Update Record \(Checklist\)](#): updates an existing Checklist record
- [Delete Record \(Checklist\)](#): deletes an existing Checklist record

List of Values (Audit)

Description

Retrieves a list of Audits or Visits in a paged list. Use this function to locate Audit or Visit record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/audit **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
offset	Optional	int	No	No	60
pageSize	Optional	int	No	No	30
modifiedSince	Optional	string	No	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	No	2015-05-19 13:30:39
status	Optional	string	Yes	No	Scheduled~In Progress
softDelete	Optional	string	No	No	true

Example URL

.../services/rest/audit/?offset=2&pageSize=20

Response Details

For a successful response, XML is returned with an AuditVisitLinkList root element containing an entries element for each matched audit or visit. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Audit/Visit record's internal ID
recordLink	string	URI to the AuditRestService Retrieve record service for this audit/visit
code	string	Audit/Visit's code business key
dueDate	string	Audit/Visit's due date
siteCode	string	Audit/Visit's site code
supplierCode	string	Audit/Visit's supplier code

Element	Type	Description
templateCode	string	Audit/Visit's template code

Error Messages

Element	Message	Meaning
pageSize	Error Code: INVALIDPAGESIZE	Page Size must be between 1 and 100

Retrieve Record by ID (Audit)

Description

Retrieves a single Audit or Visit record's details using the record's internal unique ID (which is not visible in the UI). Use this function to retrieve the full details of an individual audit or visit.

Endpoint address: /services/rest/audit/{id} **HTTP method:** GET

Request Details

The URL contains the {id} parameter that determines the record to retrieve.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
locale	Optional	string	Yes	en_GB~fr Locale country code. If used, fields that have language translations have a localeData element returned for the specified locales if data is present.

Example URL

.../services/rest/audit/105

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of an AuditFullDTO root element containing the individual attributes of the requested Audit or Visit record. If an ID is not specified, a list of all audits/visits is returned (per the List of Values function).

File Structure:

The main elements of the auditFullDTO root element:

- auditTemplateSnapshot and Template - the settings from the associated template.
- nonConformances - If the audit/visit has any non conformances/issues raised against it, they appear within a nonConformances element - a separate element for each.
- supplier and site - details of the supplier and site that the audit/visit is related to.

- `udfData` - the contents of any user-defined custom fields associated to the Audit/Visit record. `nonConformances` may also contain `udfData` elements, if the Non Conformance record has user-defined custom fields configured.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 IllegalStateException: Cannot find the Audit to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key (Audit)

Description

Retrieves a single Audit or Visit record's details using its business key (code). Use this function to retrieve the full details of an individual audit or visit using its key code.

Endpoint address: `/services/rest/audit/byKey/{code}` **HTTP method:** GET

Request Details

There are no request parameters, but the URL contains the `{code}` parameter that determines the record to retrieve.

Example URL

`.../services/rest/audit/byKey/110`

Response Details

If successful, an HTTP 301 redirect response is sent containing a header `Location` with the URI of the document requested.

The returned URI represents the call to be made to the Retrieve Record by ID function (with the `id` substituted with the relevant value), which will return an `AuditFullDTO` Element Message containing the individual attributes of the requested Audit or Visit record.



Note:

If your method of calling the API is configured to automatically follow HTTP 301 redirects, the call may automatically return a `AuditFullDTO` root element containing the individual attributes of the requested Audit or Visit record.

File Structure:

The main elements of the `auditFullDTO` root element:

- `auditTemplateSnapshot` and `Template` - the settings from the associated template.

- **nonConformances** - If the audit/visit has any non conformances/issues raised against it, they appear within a nonConformances element - a separate element for each.
- **supplier and site** - details of the supplier and site that the audit/visit is related to.
- **udfData** - the contents of any user-defined custom fields associated to the Audit/Visit record. nonConformances may also contain udfData elements, if the Non Conformance record has user-defined custom fields configured.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
code	HTTP 404 Not Found	Invalid {code} - blank or not found
code	HTTP 404 Not Found	Invalid {code} - not found

Check Record Modification Timestamp (Audit)

Description

Retrieves the last modification date and time of an Audit or Visit record. Use this function to determine when the audit or visit was last updated.

Endpoint address: /services/rest/audit/{id} **HTTP method:** HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/audit/105

Response

If successful, an HTTP 200 response is sent containing the Last-Modified header to show the last modification date and time of the last update of the requested Audit or Visit record.

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 417 Expectation Failed	Invalid {id} - not found
id	HTTP 404 Not Found	Invalid {id} - not numeric

Create Record (Audit)

Description

Creates a new Audit or Visit record. Use this function to create a new audit or visit in Supplier Evaluation based on data sourced from the external system.

Dependencies: The Supplier and Site must be present in the application and its record ID obtained. If assigning a Business Category, the ID of the Business Category record must be present in the application and its record ID obtained. If assigning a Technologist or People Present, the users must be present in the application and the record IDs of the User record obtained. If non conformances/issues are to be included, and they are associated to a specific user of the application, the record IDs of the User record must be obtained. An audit/visit must be associated to an audit/visit template. For more information, see [Dependencies](#).

Endpoint address: /services/rest/audit **HTTP method:** POST

Request Details

The body of the request contains an AuditFullDTO element to specify the detail of the audit or visit to create, which is based on the AuditFullDTO element returned when retrieving an audit/visit. As a minimum, the fields shown in the following table must be populated.

Audit/Visit Mandatory Fields

Field Name	Element Name
Audit/Visit Template	template / id (or code) auditTemplateSnapshot
Audit/Visit Code	code
Lead Technologist	leadTechnicalManager / code
Supplier	supplier / id
Site	site / id
Due Date	dueDate
Record Type	recordType
Status	status / id
From Date	fromDate
To Date	toDate
People Present	peoplePresent / name
Deleted Record	deleted (Set to false)



Note:

If the status is Scheduled, fromDate, toDate, and peoplePresent are not mandatory.

To ensure a valid unique business key code is automatically assigned to the record being created, a value of zero must be passed in the <code> element of the XML when submitting the POST request.

People Present Table:

This table is used to identify the people present during the audit or visit. The options for completing the table depend on the settings in the audit/visit's template:

- The table must contain at least one entry (whether an audit or a visit).
- If a visit or an internal audit, the entry should be a user of the system who has a Technologist role and is set as an approved auditor. The template may have been set to just allow specific auditors.
- If a third-party audit, the entry should be a Certification Body. The template may have been set to just allow specific auditors.
- Rows of free text may be added to the table.

File Structure:

The main elements of the AuditFullDTO root element are shown in the following table:

Summary of AuditFullDTO Elements

Element Name	Description
announced costChargedToSupplier auditTemplateSnapshot	Values from the audit/visit template, required to control the record's behavior when it is progressed within Supplier Evaluation.
code	The audit/visit's business key.
dueDate fromDate toDate leadTechnical Manager recordType	Values that may be mandatory for the audit/visit to be created.
nonConformances	Container for the non conformance data - a separate element for each non conformance/issue.
supplier	ID of the supplier that the audit/visit is associated to.
site	ID of the site that the audit/visit is associated to.
status	The audit/visit's status.
template	ID of the audit/visit template that the audit/visit is associated to.

Example Request XML

This example shows the minimum requirement to be able to create an In Progress audit or visit.

```
<ns0:auditFullDTO xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full"
xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple">
  <ns0:announced>true</ns0:announced>
  <ns0:costChargedToSupplier>false</ns0:costChargedToSupplier>
  <ns0:auditTemplateSnapshot>
    <ns0:allowWorkingOffline>false</ns0:allowWorkingOffline>
    <ns0:allowableScoreOptions>
      <ns1:code>AMBER</ns1:code>
    </ns0:allowableScoreOptions>
    <ns0:allowableScoreOptions>
      <ns1:code>GREEN</ns1:code>
    </ns0:allowableScoreOptions>
    <ns0:auditType>INTERNAL</ns0:auditType>
  </ns0:auditTemplateSnapshot>
</ns0:auditFullDTO>
```

```

<ns0:auditVisibility>OPTIONAL</ns0:auditVisibility>
<ns0:autoSchedule>>false</ns0:autoSchedule>
<ns0:buyerInformationRequired>>true</ns0:buyerInformationRequired>
<ns0:categoriesRequired>2</ns0:categoriesRequired>
<ns0:categoryRequired>>false</ns0:categoryRequired>
<ns0:certificationBodiesSelectable>ALL</ns0:certificationBodiesSelectable>
<ns0:closureDeadline>56</ns0:closureDeadline>
<ns0:closureDeadlineNoOfDays>0</ns0:closureDeadlineNoOfDays>
  <ns0:commentsNotVisibleToOrganisationRequired>true
</ns0:commentsNotVisibleToOrganisationRequired>
<ns0:costChargedToSupplier>>false</ns0:costChargedToSupplier>
<ns0:defaultFrequency>12</ns0:defaultFrequency>
<ns0:defaultIssueCompletion>28</ns0:defaultIssueCompletion>
<ns0:extendedDeadline>56</ns0:extendedDeadline>
<ns0:extendedDeadlineNoOfDays>0</ns0:extendedDeadlineNoOfDays>
<ns0:internalAuditorsSelectable>ALL</ns0:internalAuditorsSelectable>
<ns0:isActive>true</ns0:isActive>
<ns0:localeData>
  <ns0:name>Checklist Audit</ns0:name>
  <ns0:id>49</ns0:id>
</ns0:localeData>
<ns0:mayExternalManagersEditConfig>MAY_NOT_EDIT
</ns0:mayExternalManagersEditConfig>
<ns0:nonConformanceCreationMethod>VIA_CHECKLISTS
</ns0:nonConformanceCreationMethod>
<ns0:organisationType>SITE</ns0:organisationType>
<ns0:recordType>AUDIT</ns0:recordType>
<ns0:scoreAudit>true</ns0:scoreAudit>
<ns0:scoreRequired>true</ns0:scoreRequired>
<ns0:stampOrgWithLastAuditDate>true</ns0:stampOrgWithLastAuditDate>
<ns0:supplierGenerated>>false</ns0:supplierGenerated>
<ns0:useAuditVisitStandards>>false</ns0:useAuditVisitStandards>
  <ns0:id>6</ns0:id>
</ns0:auditTemplateSnapshot>
<ns0:dueDate>2017-04-27</ns0:dueDate>
<ns0:fromDate>2017-04-27</ns0:fromDate>
<ns0:leadTechnicalManager>
  <ns1:code>PROD-TECH</ns1:code>
  <ns1:disabled>>false</ns1:disabled>
  <ns1:externalAuthenticationUser>>false</ns1:externalAuthenticationUser>
  <ns1:timeZone>Europe/London</ns1:timeZone>
  <ns1:userType>RETAILER</ns1:userType>
  <ns1:id>12</ns1:id>
</ns0:leadTechnicalManager>
<ns0:nonConformances>
  <ns0:description><p>Test</p></ns0:description>
  <ns0:reference>AR01</ns0:reference>
  <ns0:status>
    <ns1:status>OPEN</ns1:status>
  </ns0:status>
  <ns0:type>
    <ns1:code>MAJOR</ns1:code>
  </ns0:type>
  <ns0:code>AR01-11</ns0:code>
</ns0:nonConformances>
<ns0:recordType>AUDIT</ns0:recordType>
<ns0:site>
  <ns1:id>27</ns1:id>
</ns0:site>
<ns0:status>
  <ns1:status>IN PROGRESS</ns1:status>
</ns0:status>

```



```

<ns0:supplier>
  <ns1:id>21</ns1:id>
</ns0:supplier>
<ns0:template>
  <ns1:id>8</ns1:id>
</ns0:template>
<ns0:toDate>2017-04-27</ns0:toDate>
<ns0:code>AR1</ns0:code>
<ns0:deleted>false</ns0:deleted>
</ns0:auditFullDTO>

```

Set the deleted attribute to false to ensure the record can be seen in the UI. If omitted or set to true, it will exist, but will not be visible.

Where the record is linked to another record, such as the Template, the business key or ID must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record. The supplier and site elements can contain just the ID of the respective record links.

A unique code for the audit/visit must be assigned.

Templates:

The audit/visit is linked to its template using the template and auditTemplateSnapshot elements. The template element can contain just the code or ID of the associated Template record.

The auditTemplateSnapshot element must be completed to provide the parameters that control the audit/visits' behavior as it progresses through its Supplier Evaluation workflow. If scoring is used, the allowable options can be specified using the code or ID of the glossary record; if none are specified, all options will be available when the user selects the score.

Non Conformances:

If the audit/visit is to be created with non conformances/issues already raised against it, they can be added within a nonConformances element - a separate element for each. A unique code must be assigned to each. The status can be set using its code or ID.

For further information about the process for creating audit, non conformances and checklists, see [Dependencies](#).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a AuditVisitLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Audit/Visit record's internal ID
recordLink	string	URI to the newly created Audit/Visit record, for use in a GET request
code	string	Audit/Visit's code business key
dueDate	string	Audit/Visit's due date
siteCode	string	Audit/Visit's site code

Element	Type	Description
supplierCode	string	Audit/Visit's supplier code
templateCode	string	Audit/Visit's template code

Error Responses

If the supplied data does not result in a valid Audit or Visit (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Error Messages

Element	Message	Meaning
peoplePresent	WARNING: class com.micros.creations.core.domain.Audit.peoplePresent - There must be at least one entry in the People Present table	People Present not provided
recordType	IllegalArgumentException: No enum constant com.micros.creations.core.type.AuditRecordType. <###>	Record Type not found
supplier/id code site/id code supplierId supplierCode	HTTP 417 IllegalStateException: Cannot locate keyword for class com.micros.creations.core.domain.Site using id: <###>, code:<###>, supplierCode:<###> and supplierId:<###>	Site Id not found Site Code not found Supplier Id not found Supplier Code not found
template	EntityNotFoundException: Unable to find com.micros.creations.core.domain.AuditTemplate with id <###>	Template not found
fromDate	ERROR: class com.micros.creations.core.domain.Audit.fromDate - The condition is invalid	From Date not provided
code	ERROR: class com.micros.creations.core.domain.Audit.code - The condition is invalid	Audit/Visit code not provided
code	ERROR: class com.micros.creations.core.domain.Audit.code - Another object exists with this value	Audit/Visit code already exists

Update Record (Audit)

Description

Updates an existing Audit or Visit record. Use this function to update an audit or visit's details in Supplier Evaluation based on data sourced from the external system.

Endpoint address: /services/rest/audit/{id} **HTTP method:** PUT

Request Details

The body of the request contains an AuditFullDTO element to specify the updates to the Audit or Visit record, which is based on the AuditFullDTO type returned when retrieving an audit/visit.

Note:

The request content is similar to that for creating an audit or visit.

As a minimum, the values specified as mandatory for the Create Record function (see above) must be included. However, when updating records, all values must be included.

If a value or element is omitted from the request, the field contents will be cleared on the Audit/Visit record.

If the audit/visit has any non conformances/issues raised against it, they appear within a nonConformances element - a separate element for each. The non conformances/issues can be updated or created when updating the audit/visit. If creating a new Non Conformance record, assign a unique code and omit the id element.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing an AuditVisitLink element. The AuditVisitLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Audit/Visit record's internal ID
recordLink	string	URI to the Audit/Visit record, for use in a GET request
code	string	Audit/Visit's code business key
dueDate	string	Audit/Visit's due date
siteCode	string	Audit/Visit's site code
supplierCode	string	Audit/Visit's supplier code
templateCode	string	Audit/Visit's template code

Error Responses

If the supplied data does not result in a valid Audit or Visit (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

List of Values (Checklist)

Description

Retrieves a list of Checklists in a paged list. Use this function to locate Checklist record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/audit/{auditId}/checklist **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
auditId	Optional	long	No	No	105
offset	Optional	int	No	No	60
pageSize	Optional	int	No	No	30

Example URL

.../services/rest/audit/105/checklist/?offset=2&pageSize=20

Response Details

For a successful response, XML is returned with an AuditChecklistLinkList root element containing an entries element for each matched audit or visit. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Checklist record's internal ID
recordLink	string	URI to the AuditRestService Retrieve record service for this checklist
parentId	string	The parent Audit/Visit's internal ID
auditCode	string	The parent Audit/Visit's code business key
globalId	string	Checklist's code business key

Error Messages

In the event that an error occurs, an HTTP 500 response is sent.

Retrieve Record by ID (Checklist)

Description

Retrieves a single Checklist record's details using the record's internal unique ID (which is not visible in the UI) and its parent audit/visit's internal unique Id. Use this function to retrieve the full details of an individual audit or visit.

Endpoint address: /services/rest/audit/{auditId}/checklist/{id} **HTTP method:** GET

Request Details

The URL contains the {auditId} parameter that identifies the parent Audit/Visit record, and the {id} parameter that determines the record to retrieve.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
locale	Optional	string	Yes	en_GB~fr Locale country code. If used, fields that have language translations have a localeData element returned for the specified locales if data is present.

Example URL

.../services/rest/audit/105/checklist/7

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of an auditChecklistFullDTO root element containing the individual attributes of the requested Checklist record. If a Checklist ID is not specified, a list of all the audit/visit's checklists is returned (as per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
auditId	HTTP 404 Not found	Invalid {auditId} - not numeric
auditId	HTTP 417 IllegalStateException: Cannot find the Audit to return with id:<###>	Invalid {auditId} - not found
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 IllegalStateException: Cannot find the Audit to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key (Checklist)

Description

Retrieves a single Checklist record's details using its business key (code) and its parent audit/visit's code. Use this function to retrieve the full details of an individual checklist using its key code.

Endpoint address: /services/rest/audit/byKey/{auditCode}/checklist/{checklistCode} **HTTP method:** GET

Request Details

There are no request parameters, but the URL contains the {auditCode} parameter that identifies the parent Audit/Visit record, and the {checklistCode} parameter that determines the record to retrieve.

Example URL

.../services/rest/audit/byKey/110/checklist/7374be1e-2a6c-11e7-8fda-0021f60dd918

Response Details

If successful, an HTTP 301 redirect response is sent containing a header Location with the URI of the document requested.

The returned URI represents the call to be made to the Retrieve Record by ID function (with the id substituted with the relevant value), which will return an auditChecklistFullIDTO root element containing the individual attributes of the requested Checklist record.



Note:

If your method of calling the API is configured to automatically follow HTTP 301 redirects, the call may automatically return an auditChecklistFullIDTO root element containing the individual attributes of the requested Checklist record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
auditCode	HTTP 404 Not Found	Invalid {auditCode} - blank or not found
checklistCode	HTTP 404 Not Found	Invalid {checklistCode} - blank or not found

Create Record (Checklist)

Description

Creates a new Checklist record. Use this function to create a new checklist in Supplier Evaluation based on data sourced from the external system.

Dependencies: To create an audit with a checklist, the parent Audit record must be created using the Create Record (Audit/Visit) function, before creating the associated Checklist records.

Therefore, when creating the checklist, the parent Audit/Visit record must be present in the application and its record ID obtained. A checklist's questions and the permitted answers are determined by the checklist template that has been assigned to the audit/visit's template. The IDs of the Checklist Template record must also be provided when creating a template. For more information, see [Dependencies](#).

Endpoint address: /services/rest/audit{auditId}/checklist **HTTP method:** POST

Request Details

The body of the request contains an auditChecklistFullDTO element to specify the detail of the checklist to create, which is based on the auditChecklistFullDTO element returned when retrieving a checklist. As a minimum, the fields shown in the following table must be populated.

Checklist Mandatory Fields

Field Name	Element Name
Audit/Visit Id	audit / id See note 1 .
Audit/Visit Code	audit /code See note 1 .
Checklist Template	localeData / id
Checklist Comments Allowed	commentsAllowed
Questionnaire Questions	questionnaire / questionnaireQuestion / questionDetails
Question Answers	questionnaire / questionnaireQuestion / questionnaireAnswer questionnaire / questionnaireQuestion / quesAvailAnsFull
Deleted Record	deleted (Set to false)

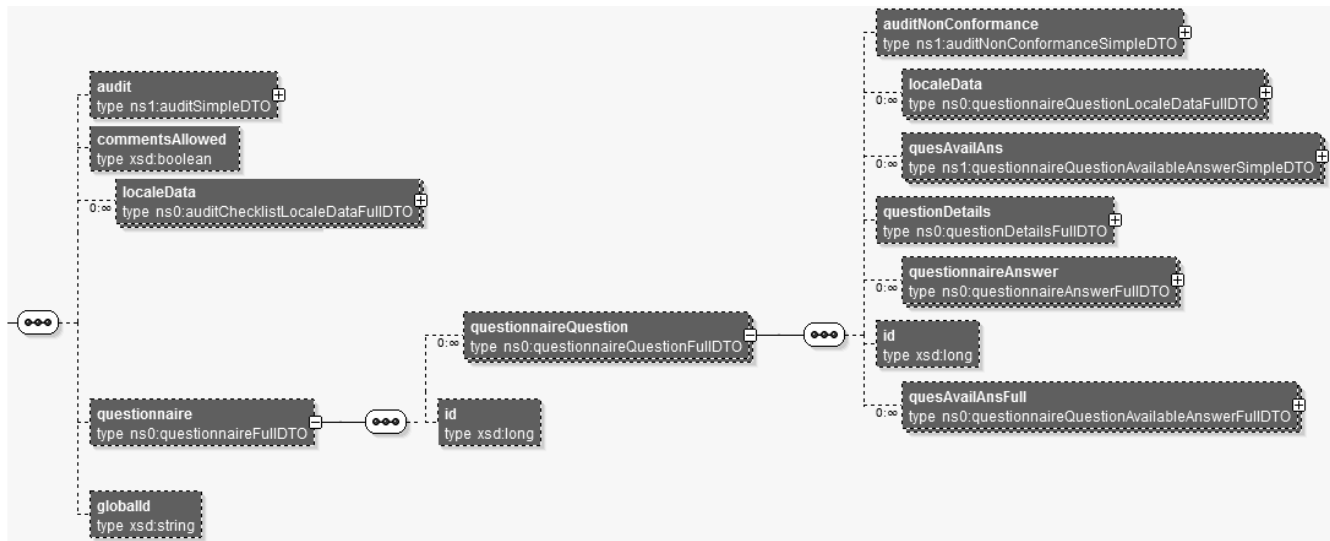
Notes:

1. Either code or ID may be specified to identify the audit/visit.

File Structure:

The following diagram shows the structure of the auditChecklistFullDTO request file structure:

Figure 4-3 RESTful Checklist Request



Summary of AuditChecklistFullIDTO Elements

Element Name	Description
audit	Identifies the parent audit record by code or id.
commentsAllowed	A value from the checklist template, required to control the record's behavior when it is progressed within Supplier Evaluation.
localeName	Identifies the checklist template by name and id.
questionnaire	Groups the questions and their answers.
questionnaireQuestion	Groups the checklist questions and their answers - repeated for each question in the checklist. reference is the question reference.
questionDetails	Individual questions. question is the question text. answerComment is the answer's comments (if configured). answerFieldType is code of answer type (MULTI_SELECTION, TEXT, or SINGLE_SELECTION). commentAllowed is whether comments are allowed.
questionnaireAnswer	Groups the answer details.
questionnaireAvailableAnswer	Answers for the individual questions. code is the question's unique key. commentMandatory is whether comments are mandatory or not. generateNonConformance is whether an answer will generate an issue.
quesAvailAnsFull	The available answers - repeated for each answer. code is the question's unique key. commentMandatory is whether comments are mandatory or not. id and keywordTypeEnum are the keys to the Issues glossary.

Process for Creating Checklist Audits

The general process for creating checklist audits:

1. Create the Audit/Visit. While it is possible to create a checklist audit using the Create Audit function, it is recommended that they be created within the Supplier Evaluation application.

A method of creating a new audit:

- a. Retrieve an existing audit using a GET function.
- b. Omit the id and statusChangeHistory elements from the retrieved file.
- c. Omit localData elements from status and allowableAuditStandards.
- d. Assign a new unique code.
- e. Link to the Supplier, Site, and Audit Template records using the relevant IDs.
- f. Change other content to reflect the new audit.
- g. Submit using the POST function.
- h. If non conformances are to be created at the time of creating the audit/visit, include them as nonConformances elements.

Note:

The API does not automatically create a non conformance based on the answer of a checklist question. Non conformances must be assigned when either creating or updating the Audit/Visit record.

2. Retrieve the code or ID of the newly created Audit, and its template detail using a GET function.
3. Create the checklist using the POST function (repeat for each checklist):
 - Identify the parent audit/visit by setting the code or id within audit element.
 - Can retrieve an existing checklist using a GET function, in which case the id and questionnaireQuestionAvailableAnswer elements must be omitted. Also omit any id elements within the questionnaire element, except those that relate to the auditNonConformance element.

Example Request XML

This example shows the minimum requirement to be able to create an audit checklist. This example contains three questions, each with the available answers of Yes, No, or Not Applicable.

Figure 4-4 Checklist Questions

Details Attachments Change History		
Questions		
Ref1	Question 1	
	Yes	Yes comment
Ref2	Question 2	
	No	No comment
	Assigned to: Assignee	Deadline: 20/05/17
Ref3	Question 3	
	Not Applicable	N/A comment

```

<ns0:auditChecklistFullDTO xmlns:ns1="http://www.micros.com/creations/core/domain/dto/
v1p0/simple" xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full">
  <ns0:audit>
    <ns1:code>AUD09</ns1:code>
    <ns1:id>132</ns1:id>
  </ns0:audit>
  <ns0:commentsAllowed>false</ns0:commentsAllowed>
  <ns0:localeData>
    <ns0:name>
      <![CDATA[ API Test Template ]]>
    </ns0:name>
    <ns0:id>563</ns0:id>
  </ns0:localeData>
  <ns0:questionnaire>
    <ns0:questionnaireQuestion>
      <ns0:localeData>
        <ns0:reference>
          <![CDATA[Ref1]]>
        </ns0:reference>
      </ns0:localeData>
      <ns0:questionDetails>
        <ns0:answerComment>
          <![CDATA[Yes comment]]>
        </ns0:answerComment>
        <ns0:answerFieldType>SINGLE_SELECTION</ns0:answerFieldType>
        <ns0:commentAllowed>true</ns0:commentAllowed>
        <ns0:localeData>
          <ns0:question>
            <![CDATA[Question 1]]>
          </ns0:question>
        </ns0:localeData>
      </ns0:questionDetails>
      <ns0:questionnaireAnswer>
        <ns0:questionnaireAvailableAnswer>
          <ns1:code>QQAA7236</ns1:code>
          <ns1:commentMandatory>false</ns1:commentMandatory>
          <ns1:generateNonConformance>false</ns1:generateNonConformance>
          <ns1:localeData>
            <ns0:answer>Yes</ns0:answer>
          </ns1:localeData>
        </ns0:questionnaireAvailableAnswer>
        <ns0:questionnaireQuestion>
          <ns1:localeData>
            <ns0:reference>
              <![CDATA[Ref1]]>
            </ns0:reference>
          </ns1:localeData>
        </ns0:questionnaireQuestion>
      </ns0:questionnaireAnswer>
      <ns0:quesAvailAnsFull>
        <ns0:code>QQAA7236</ns0:code>
        <ns0:commentMandatory>false</ns0:commentMandatory>
        <ns0:generateNonConformance>false</ns0:generateNonConformance>
        <ns0:localeData>
          <ns0:answer>Yes</ns0:answer>
        </ns0:localeData>
      </ns0:quesAvailAnsFull>
      <ns0:quesAvailAnsFull>
        <ns0:code>QQAA7237</ns0:code>
        <ns0:commentMandatory>true</ns0:commentMandatory>
        <ns0:generateNonConformance>true</ns0:generateNonConformance>

```

```

        <ns0:localeData>
          <ns0:answer>No</ns0:answer>
        </ns0:localeData>
      </ns0:quesAvailAnsFull>
    <ns0:quesAvailAnsFull>
      <ns0:code>QQAA7238</ns0:code>
      <ns0:commentMandatory>false</ns0:commentMandatory>
      <ns0:generateNonConformance>false</ns0:generateNonConformance>
      <ns0:localeData>
        <ns0:answer>Not Applicable</ns0:answer>
      </ns0:localeData>
    </ns0:quesAvailAnsFull>
  </ns0:questionnaireQuestion>
  <ns0:questionnaireQuestion>
    <ns0:localeData>
      <ns0:reference>
        <![CDATA[Ref2]]>
      </ns0:reference>
    </ns0:localeData>
    <ns0:questionDetails>
      <ns0:answerComment>
        <![CDATA[No comment]]>
      </ns0:answerComment>
      <ns0:answerFieldType>SINGLE_SELECTION</ns0:answerFieldType>
      <ns0:commentAllowed>true</ns0:commentAllowed>
      <ns0:localeData>
        <ns0:question>
          <![CDATA[Question 2]]>
        </ns0:question>
      </ns0:localeData>
    </ns0:questionDetails>
    <ns0:questionnaireAnswer>
      <ns0:questionnaireAvailableAnswer>
        <ns1:code>QQAA7240</ns1:code>
        <ns1:commentMandatory>true</ns1:commentMandatory>
        <ns1:generateNonConformance>true</ns1:generateNonConformance>
        <ns1:localeData>
          <ns0:answer>No</ns0:answer>
        </ns1:localeData>
      </ns0:questionnaireAvailableAnswer>
    </ns0:questionnaireQuestion>
    <ns1:localeData>
      <ns0:reference>
        <![CDATA[Ref2]]>
      </ns0:reference>
    </ns1:localeData>
  </ns0:questionnaireQuestion>
</ns0:questionnaireAnswer>
<ns0:quesAvailAnsFull>
  <ns0:code>QQAA7239</ns0:code>
  <ns0:commentMandatory>false</ns0:commentMandatory>
  <ns0:generateNonConformance>false</ns0:generateNonConformance>
  <ns0:localeData>
    <ns0:answer>Yes</ns0:answer>
  </ns0:localeData>
</ns0:quesAvailAnsFull>
<ns0:quesAvailAnsFull>
  <ns0:code>QQAA7240</ns0:code>
  <ns0:commentMandatory>true</ns0:commentMandatory>
  <ns0:generateNonConformance>true</ns0:generateNonConformance>
  <ns0:localeData>
    <ns0:answer>No</ns0:answer>
  </ns0:localeData>
</ns0:quesAvailAnsFull>

```

```

        </ns0:localeData>
    </ns0:quesAvailAnsFull>
    <ns0:quesAvailAnsFull>
        <ns0:code>QQAA7241</ns0:code>
        <ns0:commentMandatory>false</ns0:commentMandatory>
        <ns0:generateNonConformance>false</ns0:generateNonConformance>
        <ns0:localeData>
            <ns0:answer>Not Applicable</ns0:answer>
        </ns0:localeData>
    </ns0:quesAvailAnsFull>
</ns0:questionnaireQuestion>
<ns0:questionnaireQuestion>
    <ns0:localeData>
        <ns0:reference>
            <![CDATA[Ref3]]>
        </ns0:reference>
    </ns0:localeData>
    <ns0:questionDetails>
        <ns0:answerComment>
            <![CDATA[N/A comment]]>
        </ns0:answerComment>
        <ns0:answerFieldType>SINGLE_SELECTION</ns0:answerFieldType>
        <ns0:commentAllowed>true</ns0:commentAllowed>
        <ns0:localeData>
            <ns0:question>
                <![CDATA[Question 3]]>
            </ns0:question>
        </ns0:localeData>
    </ns0:questionDetails>
    <ns0:questionnaireAnswer>
        <ns0:questionnaireAvailableAnswer>
            <ns1:code>QQAA7244</ns1:code>
            <ns1:commentMandatory>false</ns1:commentMandatory>
            <ns1:generateNonConformance>false</ns1:generateNonConformance>
            <ns1:localeData>
                <ns0:answer>Not Applicable</ns0:answer>
            </ns1:localeData>
        </ns0:questionnaireAvailableAnswer>
        <ns0:questionnaireQuestion>
            <ns1:localeData>
                <ns0:reference>
                    <![CDATA[Ref3]]>
                </ns0:reference>
            </ns1:localeData>
        </ns0:questionnaireQuestion>
    </ns0:questionnaireAnswer>
</ns0:quesAvailAnsFull>
<ns0:quesAvailAnsFull>
    <ns0:code>QQAA7242</ns0:code>
    <ns0:commentMandatory>false</ns0:commentMandatory>
    <ns0:generateNonConformance>false</ns0:generateNonConformance>
    <ns0:localeData>
        <ns0:answer>Yes</ns0:answer>
    </ns0:localeData>
</ns0:quesAvailAnsFull>
<ns0:quesAvailAnsFull>
    <ns0:code>QQAA7243</ns0:code>
    <ns0:commentMandatory>true</ns0:commentMandatory>
    <ns0:generateNonConformance>true</ns0:generateNonConformance>
    <ns0:localeData>
        <ns0:answer>No</ns0:answer>
    </ns0:localeData>
</ns0:quesAvailAnsFull>

```

```

    <ns0:quesAvailAnsFull>
      <ns0:code>QQA7244</ns0:code>
      <ns0:commentMandatory>>false</ns0:commentMandatory>
      <ns0:generateNonConformance>>false</ns0:generateNonConformance>
      <ns0:localeData>
        <ns0:answer>Not Applicable</ns0:answer>
      </ns0:localeData>
    </ns0:quesAvailAnsFull>
  </ns0:questionnaireQuestion>
</ns0:questionnaire>
<ns0:globalId>7470e250-36cc-4dea-a0f1-eb6040185865</ns0:globalId>
<ns0:deleted>>false</ns0:deleted>
</ns0:auditChecklistFullDTO>

```

Set the deleted attribute to false to ensure the record can be seen in the UI. If omitted or set to true, it will exist, but will not be visible.

For further information about the process for creating audit, non conformances, and checklists, see [Dependencies](#).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing an AuditChecklistLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Checklist record's internal ID
recordLink	string	URI to the newly created Checklist record, for use in a GET request
parentId	long	The parent Audit/Visit's internal ID
auditCode	string	The parent Audit/Visit's code business key
globalId	string	Checklist's code business key

Error Responses

If the supplied data does not result in a valid Checklist (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Error Messages

Element	Message	Meaning
audit / id audit / code	HTTP 417 IllegalStateException: The checklists audit cannot be located	Audit/Visit id/code not found
id	HTTP 417 IllegalStateException: Cannot find matching relation to update on <###> looking for <###>	Invalid link to related record

Update Record (Checklist)

Description

Updates an existing Checklist record. Use this function to update a checklist's details in Supplier Evaluation based on data sourced from the external system.

Endpoint address: /services/rest/audit/{auditId}/checklist/{id} **HTTP method:** PUT

Request Details

The body of the request contains an auditChecklistFullDTO element to specify the updates to the Audit or Visit record, which is based on the auditChecklistFullDTO type which is returned when retrieving an audit/visit.



Note:

The request content is similar to that for creating a checklist.

As a minimum, the values specified as mandatory for the Create Record function (see above) must be included. However, when updating records, all values must be included.

If a value or element is omitted from the request, the field contents will be cleared on the Checklist record.

The link to the parent Audit/Visit record is defined by the code or ID within the audit element.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing an auditChecklistLink element. The auditChecklistLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Checklist record's internal ID
recordLink	string	URI to the Checklist record, for use in a GET request
parentId	long	The parent Audit/Visit's internal ID
auditCode	string	The parent Audit/Visit's code business key
globalId	string	Checklist's code business key

Error Messages

If the supplied data does not result in a valid Audit or Visit (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Delete Record (Checklist)

Deletes an existing Checklist record. Use this function to delete a checklist that has been created in error from the external system

Endpoint address: /services/rest/audit/{auditId}/checklist/{id} **HTTP method:** DELETE

Request Details

There are no request parameters, but the URL contains the {auditId} parameter that identifies the parent Audit/Visit record, and the {id} parameter that determines the record to delete.



Note:

It is not possible to manually delete checklists from an audit/visit within the application; this delete function has been provided solely for the deletion of checklists that have been created in error by the API.

To delete a checklist, the external system must be granted access to the AUDIT_CHECKLIST_DELETE endpoint.

Example URL

.../services/rest/audit/105/checklist/7

Response Details

If successful, an HTTP 200 response is sent.

Error Messages

Element	Message	Meaning
auditId	HTTP 404 Not Found	Invalid {auditId} - not numeric
auditId	HTTP 417 IllegalStateException: Cannot find the Audit to return with id:<###>	Invalid {auditId} - not found
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 IllegalStateException: Cannot find the Audit Checklist to return with id:<###>	Invalid {id} - not found

DataPrivacyService

This section describes the API for executing Data Privacy requests. The following functions are available:

- **Right to Access:** retrieves personal data from the system relating to the name of an individual
- **Right to be Forgotten:** erases personal data from the system for an individual

Although minimal personal data is held in Supplier Evaluation, it does include the names and contact details of retailer/portal owner, supplier, and third-party users. The Data Privacy API provides a means for the retailer/portal owner to:

1. Request details of personal data relating to an individual be held within the system.
2. Request that the personal data relating to the individual be removed from the system.

For additional detail on using the Data Privacy API, including examples, see [Appendix: Using the Data Privacy API](#).

**Note:**

The Supplier Evaluation and Supply Chain Hub modules have separate APIs for managing data privacy requests, such as GDPR right to access and right to be forgotten requests. For more details, see the *Oracle Retail Supply Chain Collaboration Cloud Service Integration User Guide*.

Right to Access

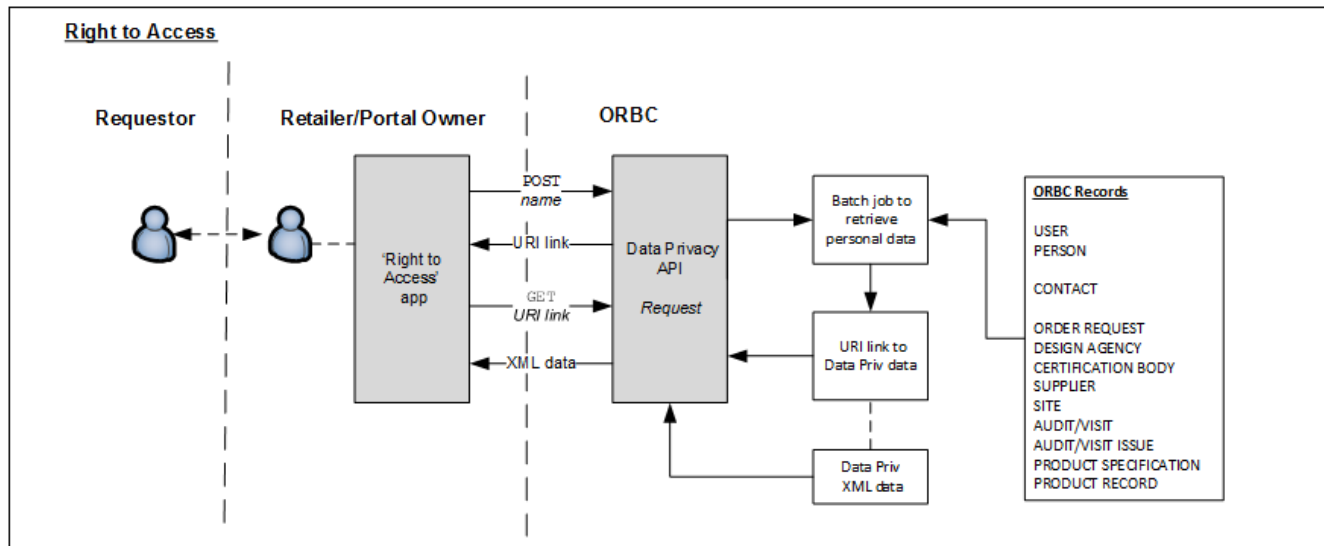
Description

Retrieves a set of personal data that relates to the name of an individual. Use this function to perform Right to Access requests for the following scenarios:

- An employee of the retailer/portal owner requests an electronic copy of their personal data that is held within the system.
- An employee of one of the retailer/portal owner's suppliers requests an electronic copy of their personal data that is held within the system.
- An employee of the retailer/portal owner's partner organizations requests an electronic copy of their personal data that is held within the system.

The API's access function allows the name of the individual to be passed, with an XML message being returned containing any personal data found for that name. The XML message is a structured text format, which is both machine and human readable.

Figure 4-5 Right to Access Process



The steps for completing a Right to Access request are as follows (for further details, see [Appendix: Using the Data Privacy API](#)):

1. Submit a POST call, passing the name of the individual.

A batch job is submitted within Supplier Evaluation to search the system for personal data for the presence of the name (for which records and fields are searched, see [Appendix: Using the Data Privacy API](#)). On completion, a Data Privacy record containing the retrieved personal data is created within Supplier Evaluation.

A link to the Data Privacy record is returned as the response to the POST call.

2. On receipt of the response to the POST call, submit a GET call, passing the returned URI link.

Note:

If the GET call is submitted before the batch job has finished compiling the personal data, just a header will be returned, and the GET call should be retried.

A message containing the personal data is returned. The XML data is structured within a `dataprivRequestDTO` root element (for details of the message structure, see [Appendix: Using the Data Privacy API](#)).

Just the personal data fields and values to identify the record are returned, not the full record contents.

Addresses may contain multiple `localeData` elements for the language translation of countries.

XML elements will only be present if they contain data.

3. Assess whether the returned data relates to the individual who made the request, editing and/or reformatting it if necessary, before passing it to the requestor.

Endpoint address: `/services/rest/datapriv/access` **HTTP methods:** POST, GET

Request Details

For the POST method, a name parameter is passed in the payload.

For the GET method, an id parameter is passed as a URI parameter.

URI Parameters

Example URL (POST method): .../services/rest/access/

Example URL (GET method): .../services/rest/access/?id=16

Response Details

For a successful response to the POST method, a link to the Data Privacy record is returned.

For a successful response to the GET method, XML is returned with a dataprivRequestDTO root element containing an element for each record where the personal data has been found.

Error Responses

In the event that an error occurs, an HTTP 500 response is sent.

Right to be Forgotten

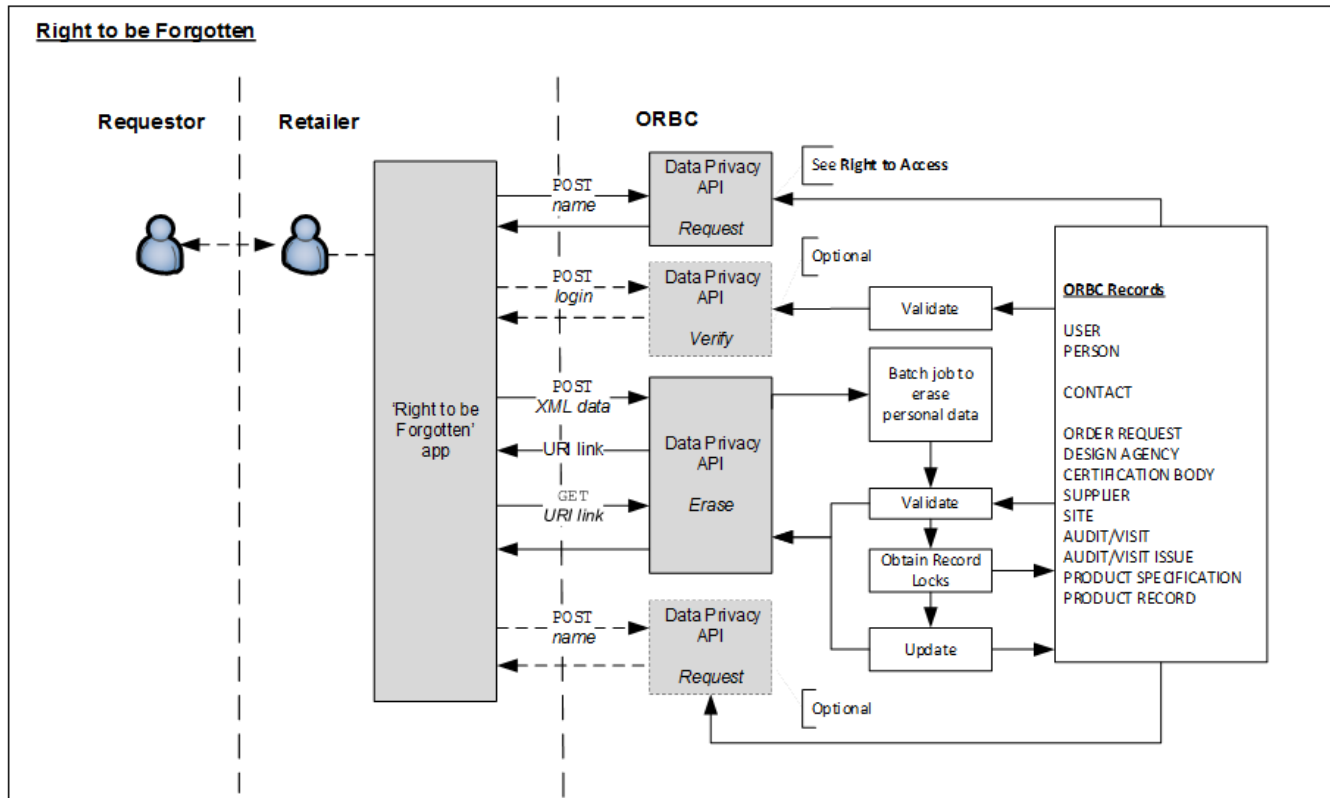
Description

Erases a set of personal data that relates to an individual. Use this function to perform Right to be Forgotten requests for the following scenarios:

- An employee of the retailer/portal owner requests their personal data be erased from the system.
- An employee of one of the retailer/portal owner's suppliers requests their personal data be erased from the system.
- An employee of a partner organizations requests their personal data be erased from the system.
- The retailer/portal owner initiates the purging of an inactive user's personal data from the system.

An erase function allows the XML message returned from a Right to Access request to be used to erase the data.

Figure 4-6 Right to be Forgotten Process



The steps for completing a Right to be Forgotten request are as follows:

1. The Right to Access service must be used to retrieve the personal data for the individual.
For details, see [Right to Access](#)

2. View, and if necessary, edit the XML returned from the Right to Access request.

As there may be any number of users with the same name, it is necessary to decide which data represents the individual who made the request, removing any that is not relevant (for further information on editing the XML message, see [Appendix: Using the Data Privacy API](#)).

3. Optionally submit a POST call to the verify function, passing the login id of the individual (obtained from the XML message returned in Step 2).

This step is used to check whether the erasure of data for the individual will be permitted, before actually submitting the request. The erase will not be permitted if the individual is responsible for the completion of a supplier site's Assessment.

XML is returned with a `dataprivVerifyDTO` root element containing a status and a message element. If the call is successful the message will contain *Success*, otherwise it will contain *Failure* and the reason for the failure will be present in the message element.

4. Submit a POST call to the erase function, passing the XML retrieved from the Right to Access request (with any edits having been applied).

A batch job is submitted to validate the request, verify that the erasure is permitted (that is, the individual is not responsible for the completion of an Assessment), and obtain locks on the records to be updated.

5. If the validation fails, or a lock cannot be obtained, the process is terminated and an XML is returned with a `dataprivErasureDTO` root element containing a message element that contains the reason for the failure, and the status element set to *Failure*.

If no exceptions are encountered, a Data Privacy record containing the personal data to erase is created and a link to the record is returned as the response to the POST call.

6. On receipt of the response to the POST call, submit a GET call to the erase function, using the returned URI link.

The update processing replaces the data passed in the XML message with 30 characters of randomly-generated text. Email addresses are changed to `x@example.com`, where `x` is 15 characters of randomly-generated text*. GPS coordinate numbers and country selections are blanked.

Some fields have a specific length, in which case the 30 characters of obfuscated text is overridden:

- Name 50 characters
- Login 20 characters
- Address lines 20 characters
- Post Code 5 characters

* Email addresses that are stored as free text are fully obfuscated, rather than set to `x@example.com`.

Only certain fields are permitted to be erased (for details, see [Appendix: Using the Data Privacy API](#)).

When updating a user account, the account is set to deactivated, and the password is set to expired.

 Note:

If the GET call is submitted before the batch job has finished compiling the personal data, just a header will be returned, and the GET call should be retried.

The erasure of data is not reversible.

7. Optionally submit another Right to Access request to check that the data has been erased as expected.
8. Inform the requestor of the outcome of their request.

The Right to be Forgotten facility may also be used for purging user accounts that are no longer active.

Endpoint address: `/services/rest/datapriv/erase` **HTTP methods:** POST (verify and erase), GET (erase)

Request Details

For the verify POST method, a `loginId` parameter is passed in the payload.

For the erase POST method, an XML message is passed in the payload.

For the erase GET method, an `id` parameter is passed as a URI parameter.

URI Parameters

Example URL (verify POST method): .../services/rest/verify/

Example URL (erase POST method): .../services/rest/erase/

Example URL (erase GET method): .../services/rest/erase/?id=32

Response Details

For a successful response to the verify POST method, XML is returned with a `dataprivVerifyDTO` root element containing a status of *Success*.

For a successful response to the erase POST method, a link to the Data Privacy record is returned.

For a successful response to the erase GET method, an HTTP 200 response is sent with a `dataprivErasureDTO` root element containing a status of *Success*.

Error Responses

For a failure response to the verify method, XML is returned with a `dataprivVerifyDTO` root element containing a status of *Failure*, with the reason in the message element.

For a failure response to the erase method, XML is returned with a `dataprivErasureDTO` root element containing a status of *Failure*, with the reason in the message element.

Error Messages

Element	Message	Meaning
user	Only one User can be erased at a time	More than one User was passed in to the erase function. The XML must only relate to a single user.
person	Only one Person can be erased at a time	More than one Person was passed in to the erase function. The XML must only relate to a single person.
id	The User is linked to Person with ID: %id but the Person included has ID: %id	The wrong Person has been included. %id is the id taken from the XML.
id	The Contacts are linked to Person with ID: %id but the Person included has ID: %id	The wrong Person has been included. %id is the id taken from the XML.
id	The Contact with ID %id is linked to a different User than the one being erased	The Contacts passed do not match the user being erased. %id is replaced with the id from the XML.
person	All Contacts being erased must be linked to the same Person	Where no User is being erased, all Contacts being erased must be for the same person.
person	Please include Person with ID: %id	If User or Contacts are passed, the matching Person must also be included.
id	Cannot find entity of type: %rec and ID: %id	An id is passed that is not found. %rec is the internal name of the record, such as ProductRecord. %id is the id taken from the XML.

Element	Message	Meaning
id	Entity of type: %rec and ID: %id is currently locked, please try again later	One of the records to be erased is currently locked. %rec is the internal name of the record, such as ProductRecord. %id is the id taken from the XML. Retry the request until the record becomes available.
id	Entity of type: %rec with id: %id has been edited during the erase process, please verify data	One of the records updated has been edited between up front validation and the processing of that particular record. %rec is the internal name of the record, such as ProductRecord. %id is the id taken from the XML.
user	The user cannot be erased because they are responsible for the approval of an Assessment	The user is named as being responsible for completing an assessment, so they cannot be erased.
login id	Cannot find User with login ID: %id	No User account found for the passed login id. %id is the login id.
	The erase has failed due to unexpected internal error	Other unexpected condition has occurred.

AttachmentRestService

This section describes the API for retrieving the file attachments from certain records. The following function is available:

- [Retrieve Report Attachment](#): retrieves attachment files that have been generated by the system as report outputs
- [Retrieve Attachment](#): retrieves files that have been attached to records

Record types that have an attachments element contain a globalId element for each of the record's file attachments. The Retrieve Attachment service can be used to retrieve the actual attachment files by passing globalId as a parameter.

The record types that are capable of returning attachments data are:

- Users
- Suppliers
- Sites
- Audits
- Audit Issues (Non Conformances)
- Scorecards
- Projects
- Activities

The following attributes are included in the attachment element of the XML:

- ID

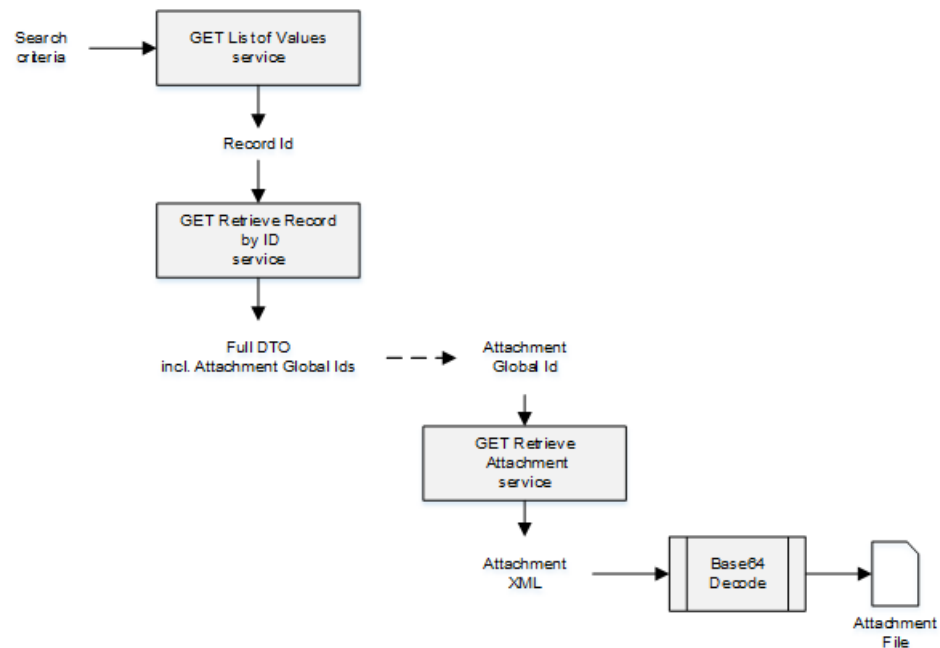
- Record Link (global Id for the attachment file)
- File Name/URL
- Description
- Bytes
- Attached by ID
- Attached On

The Attachments API is GET (retrieve) only, there is no PUT/POST (update/create) capability.

Retrieving a file returns a fileName element containing the file name, plus the encoded attachment. Retrieving a URL returns a uri element containing the URL.

The following describes the steps for retrieving file attachments:

Figure 4-7 Steps for Retrieving Attachments



- Call the List of Values service for the record with appropriate filters to locate the records for which attachments are to be retrieved. Record ids are returned for any records that match search criteria.
- Call the Retrieve Record by ID service for each record id returned from the List of Values call. The record's full DTO element is returned, containing the record's data, including an attachments element which has a globalId element for each of the record's file attachments.
- Call the Retrieve Attachment service passing the globalId as a parameter. The attachment file's data is returned as a Base 64 text string. Use a suitable decoder to generate the actual attachment file.

Retrieve Report Attachment

Description

Retrieves an attachment file that has been generated as the output of a report. The file types will be PDF, Excel, RTF, or HTML. When a report is generated, the outputs are stored in the Report Outputs area, grouped within a hierarchical folder structure.

The API provides a means of automatically locating reports for further analysis of the data within an external system. The ID of the attachment to be retrieved is sent to the external system's designated email address.

For additional detail on using the Report Attachment API, see [Appendix: Using the Report Attachment API](#).

Endpoint address: /services/rest/attachment/report/{globalId} **HTTP method:** GET

Request Details

There are no request parameters, but the URL contains the {globalId} parameter that determines the record to retrieve.

Example URL

.../services/rest/attachment/report/f25eebdd-e7c5-4bb2-b61c-f35171438386

Response Details

For a successful response, XML is returned with a data element which contains the actual attachment as a Base64 encoded text string. The string must then be processed by a Base64 decoder to convert it to the actual report file.

Error Messages

Element	Message	Meaning
globalId	HTTP 404 Not found	Invalid {globalId} - not numeric
globalId	HTTP 417 IllegalStateException: Cannot find the Attachment to return with id:<###>	Invalid {globalId} - not found

Retrieve Attachment

Description

Retrieves a file that has been attached to a record. The files types will typically be documents or images.

Endpoint address: /services/rest/attachment/attachment/{globalId} **HTTP method:** GET

Request Details

There are no request parameters, but the URL contains the {globalId} parameter that determines the record to retrieve.

Example URL

.../services/rest/attachment/attachment/RTxy92Ha4lsQE6Bjc-18t1wJ7smbYoCH

Response Details

For a successful response, XML is returned with a data element which contains the actual attachment as a Base64 encoded text string. The string must then be processed by a Base64 decoder to convert it to the actual attachment file.

Error Messages

Element	Message	Meaning
globalId	HTTP 404 Not found	Invalid {globalId} - not numeric
globalId	HTTP 417 IllegalStateException: Cannot find the Attachment to return with id:<###>	Invalid {globalId} - not found

ScorecardRestService

This section describes the API for managing assessments. The following functions are available:

- [List of Values](#): retrieves a list of assessments
- [Retrieve Record by ID](#): retrieves an Assessment record using its unique identifier
- [Retrieve Record by Business Key](#): retrieves an Assessment record using its business key
- [Check Record Modification Timestamp](#): retrieves the timestamp when an Assessment record was last updated
- [Update Record](#): updates an existing Assessment record
- [Update Sub Status](#): updates the sub status of an existing Assessment record

List of Values

Description

Retrieves a list of assessments in a paged list. Use this function to locate Assessment record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/scorecard **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/Optional	Value Type	Multiple Value Separator (-) Supported?	Wildcard Search Supported?	Example
offset	Optional	int	No	No	60

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
pageSize	Optional	int	No	No	30
status	Optional	string	Yes	Yes	IN PROGRESS~COM PLETE
siteStatus	Optional	string	Yes	Yes	AWAITING REGISTRATION
siteType	Optional	string	Yes	Yes	PRIMARY
siteCode	Optional	string	Yes	Yes	A0001-0001
siteName	Optional	string	Yes	Yes	West Road Site
siteNameBusiness Language	Optional	string	Yes	Yes	West Road Site
supplierCode	Optional	string	Yes	Yes	A0001
supplierName	Optional	string	Yes	Yes	ABC Ltd
supplierNameBusin essLanguage	Optional	string	Yes	Yes	ABC Ltd
businessCategory	Optional	string	Yes	Yes	POTATOES_LOOS E
templateCode	Optional	string	Yes	Yes	ETHICAL_ASSES SMENT
dueDate	Optional	string	Yes	No	2015-05-19
modifiedSince	Optional	string	No	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	No	2015-05-19 13:30:39

Example URLs

.../services/rest/scorecard/?offset=2&pageSize=20 .../services/rest/scorecard/?status=IN
PROGRESS~COMPLETE .../services/rest/scorecard/?
businessCategory=POTATOES_LOOSE

Response Details

For a successful response, XML is returned with a ScorecardLinkList root element containing an entries element for each matched assessment. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Assessment record's internal ID
recordLink	string	URI to the ScorecardRestService Retrieve record service for this assessment
scorecardCode	string	Assessment's code business key
siteCode	string	Assessment's site code
supplierCode	string	Assessment's supplier code
templateCode	string	Assessment's template code

The returned XML also contains a `totalRecords` element, which states the total number of retrievable records that match the filter parameters.

Error Messages

In the event that an error occurs, an HTTP 500 response is sent.

Retrieve Record by ID

Description

Retrieves a single Assessment record's details using the record's internal unique ID (which is not visible in the UI). Use this function to retrieve the full details of an individual assessment.

Endpoint address: `/services/rest/scorecard/{id}` **HTTP method:** GET

Request Details

The URL contains the `{id}` parameter that determines the record to retrieve.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
locale	Optional	string	Yes	en_GB~fr Locale country code. If used, fields that have language translations have a <code>localeData</code> element returned for the specified locales if data is present.

Example URL

.../services/rest/scorecard/54

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of a `scorecardFullDTO` root element containing the individual attributes of the requested Assessment record. If an ID is not specified, a list of all assessments is returned (per the List of Values function).

File Structure

The main elements of the `scorecardFullDTO` root element:

- `scorecardTemplateSnapshot` and `Template` - the settings from the associated template.
- `scorecardQuestions` - the assessment's questions and their answers - a separate element for each.
- `supplier` and `site` - details of the supplier and site that the assessment is related to.
- `udfData` - the contents of any user-defined custom fields associated to the Assessment record.

- attachments - a child element containing the ids of any file attachments the assessment has.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not found	Invalid {id} - not numeric
id	HTTP 417 IllegalStateException: Cannot find the Assessment to return with id:<####>	Invalid {id} - not found

Retrieve Record by Business Key

Description

Retrieves a single Assessment record's details using its business key (code). Use this function to retrieve the full details of an individual assessment using its key code.

Endpoint address: /services/rest/scorecard/byKey/{code} **HTTP method:** GET

Request Details

There are no request parameters, but the URL contains the {code} parameter that determines the record to retrieve.

Example URL

.../services/rest/scorecard/byKey/184

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of a scorecardFullDTO root element containing the individual attributes of the requested Assessment record.

File Structure

The main elements of the scorecardFullDTO root element:

- scorecardTemplateSnapshot and Template - the settings from the associated template.
- scorecardQuestions - the assessment's questions and their answers - a separate element for each.
- supplier and site - details of the supplier and site that the assessment is related to.
- udfData - the contents of any user-defined custom fields associated to the Assessment record.
- attachments - a child element containing the ids of any file attachments the assessment has.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
code	HTTP 404 Not found	Invalid {code} - blank or not found

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of an Assessment record. Use this function to determine when a assessment was last updated.

Endpoint address: /services/rest/scorecard/{id} **HTTP method:** HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/scorecard/54

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header showing the date and time of the last update of the requested Assessment record. For example:

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 417 Expectation Failed	Invalid {id} - not found
id	HTTP 404 Not found	Invalid {id} - not numeric

Update Record

Description

Updates an existing Assessment record. Use this function to update an Assessment's details in Supplier Evaluation based on data sourced from the external system..

Endpoint address: /services/rest/scorecard/{id} **HTTP method:** PUT

Request Details

The body of the request contains a ScorecardFullDTO to specify the updates to the Assessment record, which is based on the ScorecardFullDTO type returned when retrieving an Assessment.

**Note:**

When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the Assessment record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 2 - Supplier* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a ScorecardRecordLink element. The ScorecardRecordLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Assessment record's internal ID
recordLink	string	URI to the Assessment record, for use in a GET request
scorecardCode	string	Assessment's code business key
siteCode	string	Assessment's site code
supplierCode	string	Assessment's supplier code
templateCode	string	Assessment's template code

Error Responses

If the supplied data does not result in a valid Assessment (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Assessment Questions & Answers

This section explains the basic structure of the Scorecard API's payload XML, focussing on the elements which identify the Assessment and contain its Questions and Answers.

The key elements in the parent scorecardFullDTO node which are of relevance when responding to Assessments are as follows.

```
<scorecardFullDTO>
  <autoGenerationEnabled>
  <businessCategories>
  <calculatedScore>
    <code>           The Assessment's business key.
    <dueDate>        The date by which the Assessment is due to be completed.
  <extraScorecard>
  <overdue>
  <period>
  <personResponsible>
  <quarterEndDate>
  <quarterStartDate>
  <scheduled>
  <scoreTemplate>   The Assessment's questions & answers - see Questions and Answers
                    below.
```

```

<scorecardScore>
<scorecardValue>
<status>           The status of the Assessment - see Status and Sub Status below.
<statusHistory>
<supplierGenerated>
<udfData>
<week>
<year>
<id>               The Assessment's record Id.
<createdOn>        When the Assessment was created.
<updatedOn>        When the Assessment was last updated.
<due>
<scorecardTarget> The Assessment's target supplier & site - see Supplier and Site below.
<subStatusText>    The sub status of the Assessment.
<automaticDisqualificationCount>
<scorecardTargetSites>
<scorecardTargetSupplier>
<createdFromAssessmentRequestWizard>

```

Supplier and Site

The scorecardTarget element identifies the target supplier and site.

```

<ns1:scorecardTarget>
  <ns1:site>
    <ns2:code>A0072-00020</ns2:code>
    <ns2:localName>BS Site 123</ns2:localName>
    <ns2:name>BS Site 123</ns2:name>
    <ns2:id>813</ns2:id>
    <ns2:supplierId>490</ns2:supplierId>
    <ns2:supplierCode>A0072</ns2:supplierCode>
  </ns1:site>
</ns1:scorecardTarget>

```

Additional supplier name and contact information is included in the scorecardTargetSupplier element.

Status and Sub Status

The status element contains the Assessment's current status, within the status child element, for example:

```

<ns1:status>
  <ns2:applyTemplateChanges>false</ns2:applyTemplateChanges>
  <ns2:localeData>
    <ns1:description>En cours</ns1:description>
    <ns1:locale>fr</ns1:locale>
    <ns1:id>15</ns1:id>
  </ns2:localeData>
  <ns2:localeData>
    <ns1:description>In behandeling</ns1:description>
    <ns1:locale>nl</ns1:locale>
    <ns1:id>21</ns1:id>
  </ns2:localeData>
  <ns2:localeData>
    <ns1:description>In Progress</ns1:description>
    <ns1:id>14</ns1:id>
  </ns2:localeData>
  <ns2:status>IN PROGRESS</ns2:status>
  <ns2:id>2</ns2:id>
</ns1:status>

```

Multiple localeData nodes represent the language translations of the status description. The base language (English) description has no locale node. The status child element contains the status code.

The subStatusText node contains the sub status of the Assessment.

Questions and Answers

The scoreTemplate element contains the main body of the Assessment, including its questions and answers.

The multiple localeData nodes represent the language translations of the Assessment description, based on its template. The base language (English) description has no locale node. The id node contains the record id of the Assessment's template.

An Assessment's questions are represented by universalQuestions, generalQuestions or adHocQuestions elements. The structure of each is the same, the difference being how they were configured within the Assessment's template: Universal questions appear by default in all templates; General questions are added to the template from a glossary of standard questions; Ad Hoc questions are specific to the template, for that type of Assessment.

A question and its answer are structured as follows:

- Multiple localeData nodes represent the language translations of the Question text. The base language (English) description has no locale node. The questionId node contains the Question's unique id.
- scorecardAnswerType indicates the type of question. The possible types are TEXT, NUMERIC, DATE, URL, RADIO_BUTTON, CHECKBOX, SINGLE_PICKER, MULTI_PICKER and TABULAR.
 - For Text, Numeric, Date and URL answer types, a single scorecardQuestionAnswers element contains the response, within an answerText node.

```
<ns1:scorecardAnswerType>TEXT</ns1:scorecardAnswerType>
<ns1:scorecardQuestionAnswers>
  <ns1:answerText>text response</ns1:answerText>
  <ns1:id>25702</ns1:id>
  <ns1:sequence>0</ns1:sequence>
</ns1:scorecardQuestionAnswers>
```

- For Radio Button answer types, a single scorecardQuestionAnswers element contains the response, within an answerText node. Multiple scorecardAvailableAnswers elements represent the available answer options. Also copy the relevant scorecardAvailableAnswers element to scorecardQuestionAnswers

```
<ns1:scorecardAnswerType>RADIO_BUTTON</ns1:scorecardAnswerType>
<ns1:scorecardQuestionAnswers>
  <ns1:answerText>Yes</ns1:answerText>
  <ns1:scorecardAvailableAnswer>
    <ns1:id>25706</ns1:id>
    <ns1:sequence>0</ns1:sequence>
  </ns1:scorecardQuestionAnswers>
  ...
<ns1:scorecardAvailableAnswers>
<ns1:scorecardAvailableAnswers>
```

- For Checkbox answer types, one or more scorecardQuestionAnswers elements contain the responses, each within an answerText node. Multiple scorecardAvailableAnswers elements represent the available answer options. Also copy the relevant scorecardAvailableAnswers elements to scorecardQuestionAnswers.


```

<ns1:scorecardAnswerType>CHECKBOX</ns1:scorecardAnswerType>
  <ns1:scorecardQuestionAnswers>
    <ns1:answerText>Fresh</ns1:answerText>
    <ns1:comment>Locally sourced only</ns1:comment>
    <ns1:scorecardAvailableAnswer>
      <ns1:id>25703</ns1:id>
      <ns1:sequence>0</ns1:sequence>
    </ns1:scorecardQuestionAnswers>
  </ns1:scorecardQuestionAnswers>
  <ns1:scorecardQuestionAnswers>
    <ns1:answerText>Chilled</ns1:answerText>
    <ns1:scorecardAvailableAnswer>
      <ns1:id>25704</ns1:id>
      <ns1:sequence>0</ns1:sequence>
    </ns1:scorecardQuestionAnswers>
  </ns1:scorecardQuestionAnswers>
  ...
  <ns1:scorecardAvailableAnswers>
  <ns1:scorecardAvailableAnswers>
  <ns1:scorecardAvailableAnswers>

```

- For Single-select Picker answer types, a single scorecardQuestionAnswers element contains the response, within a single scorecardPicklistSingle child element. Multiple scorecardAnswerOptionDetail elements within scorecardPicklistSnapshot represent the available answer options. Copy the contents of the relevant scorecardAnswerOptionDetail element to scorecardPicklistSingle to populate the response.

```

<ns1:scorecardAnswerType>SINGLE_PICKER</ns1:scorecardAnswerType>
  <ns1:scorecardQuestionAnswers>
    <ns1:id>25707</ns1:id>
    <ns1:scorecardPicklistSingle>
      <ns2:mandatoryComments>false</ns2:mandatoryComments>
      <ns2:localeData>
        <ns1:description>Frozen</ns1:description>
        <ns1:id>105522</ns1:id>
      </ns2:localeData>
      <ns2:code>FROZEN</ns2:code>
      ...
    </ns1:scorecardPicklistSingle>
    <ns1:sequence>0</ns1:sequence>
  </ns1:scorecardQuestionAnswers>
  ...
  <ns1:scorecardPicklistSnapshot>
    ...
    <ns1:scorecardAnswerOptionDetail>
    <ns1:scorecardAnswerOptionDetail>
    <ns1:scorecardAnswerOptionDetail>
  </ns1:scorecardPicklistSnapshot>

```

- For Multi-select Picker answer types, one or more scorecardPicklistMulti elements contains the responses, within a single scorecardQuestionAnswers element. Multiple scorecardAnswerOptionDetail elements within scorecardPicklistSnapshot represent the available answer options. Copy the contents of the relevant scorecardAnswerOptionDetail elements to scorecardPicklistMulti to populate the responses.

```

<ns1:scorecardAnswerType>MULTI_PICKER</ns1:scorecardAnswerType>
<ns1:scorecardQuestionAnswers>
  <ns1:id>25708</ns1:id>
  <ns1:scorecardPicklistMulti>
    <ns2:mandatoryComments>false</ns2:mandatoryComments>
    <ns2:localeData>
      <ns1:description>Frozen</ns1:description>
      <ns1:id>105524</ns1:id>

```

```

</ns2:localeData>
<ns2:code>FROZEN</ns2:code>
...
</ns1:scorecardPicklistMulti>
<ns1:scorecardPicklistMulti>
  <ns2:mandatoryComments>false</ns2:mandatoryComments>
  <ns2:localeData>
    <ns1:description>Chilled</ns1:description>
    <ns1:id>105525</ns1:id>
  </ns2:localeData>
  <ns2:code>CHILLED</ns2:code>
  ...
</ns1:scorecardPicklistMulti>
<ns1:sequence>0</ns1:sequence>
</ns1:scorecardQuestionAnswers>
...
<ns1:scorecardPicklistSnapshot>
  ...
  <ns1:scorecardAnswerOptionDetail>
  <ns1:scorecardAnswerOptionDetail>
  <ns1:scorecardAnswerOptionDetail>
</ns1:scorecardPicklistSnapshot>

```

- For Tabular answer types, a single scorecardQuestionAnswers defines the answer type of each column of the table, and contains the responses.

The column types may be text, numeric, URL or single-select picker, defined in columnxAnswerType nodes, where x is the column number.

The responses are contained within nodes which identify the column and answer type:

- * Text answers in a cellAnswerx node, where x is the column number.
- * Numeric answers in a numberAnswerx node, where x is the column number.
- * URL answers in a urlAnswerx node, where x is the column number.
- * Single-select picker answers within a scorecardPicklistx element, where x is the column number. Multiple scorecardAnswerOptionDetail elements within a corresponding picklistSnapshotx element represent the available answer options. Copy the contents of the relevant scorecardAnswerOptionDetail element to scorecardPicklistx to populate the response.

The following example is a table of four columns (text, numeric, URL, single-select picker), populated with one row:

Site Certifications	Certification Body	Certification Year	Product Category
Red Tractor	rt@example.com	2020	Poultry

```

<ns1:scorecardAnswerType>TABULAR</ns1:scorecardAnswerType>
<ns1:scorecardQuestionAnswers>
  <ns1:id>25709</ns1:id>
  <ns1:column1AnswerType>TEXT</ns1:column1AnswerType>
  <ns1:column2AnswerType>URL</ns1:column2AnswerType>
  <ns1:column3AnswerType>NUMERIC</ns1:column3AnswerType>
  <ns1:column4AnswerType>SINGLE_PICKER</ns1:column4AnswerType>
  <ns1:cellAnswer1>Red Tractor</ns1:cellAnswer1>
  <ns1:numberAnswer3>2020</ns1:numberAnswer3>
  <ns1:scorecardPicklist4>
    <ns2:mandatoryComments>false</ns2:mandatoryComments>
    <ns2:localeData>
      <ns1:description>Poultry</ns1:description>
      <ns1:id>105527</ns1:id>
    </ns2:localeData>
  </ns1:scorecardPicklist4>

```

```

        </ns2:localeData>
        <ns2:code>POULTRY</ns2:code>
        ...
    </ns1:scorecardPicklist4>
    <ns1:urlAnswer2>https://rt@example.com</ns1:urlAnswer2>
    <ns1:sequence>0</ns1:sequence>
</ns1:scorecardQuestionAnswers>
...
<ns1:picklistSnapshot4>
    <ns2:code>CATEGORIES</ns2:code>
    <ns2:localeData>
        <ns1:description>Product Categories</ns1:description>
        <ns1:id>58276</ns1:id>
    </ns2:localeData>
    <ns2:id>2095</ns2:id>
    <ns1:scorecardAnswerOptionDetail>
        <ns2:mandatoryComments>false</ns2:mandatoryComments>
        <ns2:localeData>
            <ns1:description>Dairy</ns1:description>
            <ns1:id>105526</ns1:id>
        </ns2:localeData>
        <ns2:code>DAIRY</ns2:code>
        ...
    </ns1:scorecardAnswerOptionDetail>
    <ns1:scorecardAnswerOptionDetail>
        <ns2:mandatoryComments>false</ns2:mandatoryComments>
        <ns2:localeData>
            <ns1:description>Poultry</ns1:description>
            <ns1:id>105527</ns1:id>
        </ns2:localeData>
        <ns2:code>POULTRY</ns2:code>
        ...
    </ns1:scorecardAnswerOptionDetail>
</ns1:picklistSnapshot4>

```

- A comment node can be included with the answer for any supporting information. If commentsMandatory is true, a comment is mandatory.
- When an answer has an alternative Summary Attribute label, a summaryLabel node is present with the answer's description node.

Update Sub Status

Description

Updates the sub status of an existing Assessment record. Use this function to update an Assessment's sub status, such as when being progressed by an Accredited Body as part of third-party integration for the sourcing of data from an external system.

Two alternative endpoints are available, to locate the Assessment either by its record id or by its Code business key. They both return a record link to the Assessment.

Endpoint address: /services/rest/scorecard/{id}/substatus and /services/rest/scorecard/{byKey}/substatus

HTTP method: PUT

Request Details

The body of the request of the request is an assessmentSubStatusDTO message containing the sub status of the Assessment and any comments.

Example Request XML

```
<ns1:assessmentSubStatusDTO
  xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/simple"
  xmlns:ns0="http://www.micros.com/creations/core/domain/dto/v1p0/full">
  <ns1:substatus>RETRIEVED</ns1:substatus>
  <ns1:comments></ns1:comments>
</ns1:assessmentSubStatusDTO>
```

An Assessment's sub status is initially ISSUED; once assigned to a third party Accredited Body it becomes PENDING, making it available for the designated third party to retrieve. The workflow requires that before the third party submits their response data they first acknowledge they have retrieved the Assessment by setting the sub status to RETRIEVED.

As the regular Update (PUT) endpoint requires the full Assessment payload to be included, the Update Sub Status endpoint provides a simpler method of just returning the sub status value, locating the Assessment by its record id or business key.

Subsequently, the external system will submit the full Assessment payload with their responses using the existing regular Update (PUT) endpoint, setting the sub status RESPONDED, indicating their response is complete.

When called by the external system of a registered Accredited Body, only Assessments that are assigned to that Accredited Body can be updated.

For additional detail on using the Scorecard API for third party integration, see [Appendix: LCA Third-Party Integration](#)

Response Details

If successful, an HTTP 200 response is sent with a body containing a ScorecardRecordLink element. The ScorecardRecordLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Assessment record's internal ID
recordLink	string	URI to the Assessment record, for use in a GET request
scorecardCode	string	Assessment's code business key
siteCode	string	Assessment's site code
supplierCode	string	Assessment's supplier code
templateCode	string	Assessment's template code

Error Responses

If the supplied data does not result in a valid Assessment (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

ProjectRestService

This section describes the API for managing processes. The following functions are available:

- [List of Values](#): retrieves a list of processes

- **Retrieve Record by ID:** retrieves a Process record using its unique identifier
- **Retrieve Record by Business Key:** retrieves a Process record using its business key
- **Check Record Modification Timestamp:** retrieves the timestamp when a Process record was last updated
- **Update Record:** updates an existing Process record

List of Values

Description

Retrieves a list of processes in a paged list. Use this function to locate Process record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/project **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
offset	Optional	int	No	No	60
pageSize	Optional	int	No	No	30
projectStatus	Optional	string	Yes	Yes	IN PROGRESS~COM PLETED
templateCode	Optional	string	Yes	Yes	NEW_BAKERY_R ANGE
masterProject	Optional	Boolean	No	No	TRUE
parentProject	Optional	string	Yes	Yes	NEW_LINE_DEVE LOPMENT
projectType	Optional	string	Yes	Yes	CHANGE OF SUPPLIER
projectName	Optional	string	Yes	Yes	New Potato Product Launch
projectManager	Optional	string	Yes	Yes	John Smith
businessCategory	Optional	string	Yes	Yes	POTATOES_LOOS E
specificationType	Optional	string	Yes	Yes	FOOD
siteStatus	Optional	string	Yes	Yes	AWAITING REGISTRATION
siteType	Optional	string	Yes	Yes	PRIMARY
siteCode	Optional	string	Yes	Yes	A0001-0001
siteName	Optional	string	Yes	Yes	West Road Site
siteNameBusiness Language	Optional	string	Yes	Yes	West Road Site
supplierCode	Optional	string	Yes	Yes	A0001
supplierName	Optional	string	Yes	Yes	ABC Ltd

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
supplierNameBusinessLanguage	Optional	string	Yes	Yes	ABC Ltd
modifiedSince	Optional	string	No	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	No	2015-05-19 13:30:39

Example URLs

.../services/rest/user/?offset=2&pageSize=20 .../services/rest/user/?projectStatus=IN
PROGRESS~COMPLETED .../services/rest/user/?
templateCode=NEW_LINE_DEVELOPMENT

Response Details

For a successful response, XML is returned with a ProjectLinkList root element containing an entries element for each matched process. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Process record's internal ID
recordLink	string	URI to the ProjectRestService Retrieve record service for this process
projectId	string	Process's business key
projectName	string	Process's name
parentProjectId	string	Parent process's business key
parentProjectName	string	Parent process's name

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

In the event that an error occurs, an HTTP 500 response is sent.

Retrieve Record by ID

Description

Retrieves a single Process record's details using the record's internal unique ID (which is not visible in the UI). Use this function to retrieve the full details of an individual process.

Endpoint address: /services/rest/project/{id} **HTTP method:** GET

Request Details

The URL contains the {id} parameter that determines the record to retrieve.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
locale	Optional	string	Yes	en_GB~fr Locale country code. If used, fields that have language translations have a localeData element returned for the specified locales if data is present.

Example URL

.../services/rest/project/20

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of a projectFullIDTO root element containing the individual attributes of the requested Process record. If an ID is not specified, a list of all processes is returned (per the List of Values function).

File Structure

The main elements of the projectFullIDTO root element:

- projectTemplateSnapshot and Template - the settings from the associated template.
- briefs - the associated process briefs; a separate element for each.
- teams - the users and their roles associated to the process.
- linkedRecords - the associated linked records; a separate element for each.
- udfData - the contents of any user-defined custom fields associated to the Process record.
- attachments - a child element containing the ids of any file attachments the process has.

Processes and their activities can be nested within a hierarchy. In the absence of a function to extract the full hierarchy with a single GET call, to retrieve the hierarchy of a process a number of successive calls is required. The Parent Project field is a Boolean value which will be blank for processes that have no hierarchy, or are at the top level of the hierarchy. Use the id or code of the Parent Project to traverse the levels. Use the Activities endpoints to retrieve the process's activity details. Master processes will not have a parent project id or a process name.

If a process is linked to a record such as an Audit, the linkedRecords element contains a list of the linked records, containing the id and business key of the record (audit code, supplier code, or site code), its type, name and status, and the date it was created.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 3 - Process* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not found	Invalid {id} - not numeric
id	HTTP 417 IllegalStateException: Cannot find the Process to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key

Description

Retrieves a single Process record's details using its business key (code). Use this function to retrieve the full details of an individual process using its key code.

Endpoint address: /services/rest/project/byKey/{code} **HTTP method:** GET

Request Details

There are no request parameters, but the URL contains the {code} parameter that determines the record to retrieve.

Example URL

.../services/rest/project/byKey/943

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of a projectFullDTO root element containing the individual attributes of the requested Process record.

File Structure

The main elements of the projectFullDTO root element:

- projectTemplateSnapshot and Template - the settings from the associated template.
- teams - the users and their roles associated to the process.
- briefs - the associated process briefs; a separate element for each.
- linkedRecords - the associated linked records; a separate element for each.
- udfData - the contents of any user-defined custom fields associated to the Process record.
- attachments - a child element containing the ids of any file attachments the process has.

Processes and their activities can be nested within a hierarchy. In the absence of a function to extract the full hierarchy with a single GET call, to retrieve the hierarchy of a process a number of successive calls is required. The Parent Project field is a Boolean value which will be blank for processes that have no hierarchy, or are at the top level of the hierarchy. Use the id or code of the Parent Project to traverse the levels. Use the Activities endpoints to retrieve the process's activity details. Master processes will not have a parent project id or a process name.

If a process is linked to a record such as an Audit, the linkedRecords element contains a list of the linked records, containing the id and business key of the record (audit code, supplier code, or site code), its type, name and status, and the date it was created.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 3 - Process* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
code	HTTP 404 Not found	Invalid {code} - blank or not found

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a Process record. Use this function to determine when the process was last updated.

Endpoint address: /services/rest/project/{id} **HTTP method:** HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/project/20

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header showing the date and time of the last update of the requested Processord. For example:

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 417 Expectation Failed	Invalid {id} - not found
id	HTTP 404 Not found	Invalid {id} - not numeric

Update Record

Description

Updates an existing Process record. Use this function to update a Process in Supplier Evaluation based on data sourced from the external system.

Process records can be updated, such as to set start/end dates, process brief details, or other attributes, from an external system. Updates do not trigger any automatic re-scheduling of a Process. Dates and other values are simply updated.

The PATCH method allows for partial updates to a record, to update values without the need to submit the full record. This endpoint supports either an XML or JSON payload. The creation of new records is not supported.

Endpoint address: /services/rest/project/{id}

HTTP method: PATCH

Request Details

The body of the request contains a projectFullDTO to specify the updates to the Process record, which is based on the projectFullDTO type returned when retrieving a Process.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supply Chain Collaboration Cloud Service Data Dictionary, Volume 3 - Process* for details of their mapping to the fields within the UI. The data dictionary indicates which fields are capable of being updated.

The Process unique id is included in the request URI to identify the Process to be updated. Each transaction will represent a single Process. The payload need only include the mandatory values and the values of elements to be updated. A Process at any status can be updated.

Updating the Process record does not trigger the Process to be re-scheduled, even if the start or end dates are updated. Updates to the Supplier, Site and Process Manager are not supported.

Example JSON Requests

JSON requests consist of the following elements:

`op` specifies the operation to be performed on a field:

- `replace` a value
- `add` a value to a list
- `remove` a value from a list
- `move` a value within a list

`path` specifies the location of the field to be updated.

`value` specifies the value to be applied by the update.

Note that JSON indexes are zero based.

1. Replace an existing value:

```
{
  "op" : "replace",
  "path" : "/udfData/rtfField001",
  "value" : "new value"
},
```

2. Remove an item, in this case at the first index:

```
{
  "op" : "remove",
  "path" : "/sites/0"
},
```

3. Add a new item, in this case at the first index:

```
{
  "op" : "add",
  "path" : "/sites/0",
  "value": {
    "code": "A0006-00001",
    "supplierCode": "A0006"
  }
},
```

4. Replacing a value in multiple nested collections, referencing the index of each localeDataMap is keyed by locale, in this case referencing the default locale, but 'en_GB' or 'fr' for example would also be valid:

```
{
  "op" : "replace",
  "path" : "/briefs/0/sections/0/briefQuestions/1/availableAnswer/1/
localeDataMap/null",
  "value" : {
    "description": "new value"
  }
},
```

Example XML Requests

XML requests consist of the following elements:

The XML element specifies the operation to be performed on a field:

- replace a value
- add a value to a list
- remove a value from a list
- There is no move operation for XML, remove and add are used instead.

sel specifies the location of the field to be updated.

Note that XML indexes are 1 based.

1. Replace an existing value:

```
<replace sel="ns4:projectFullDTO/ns4:udfData/ns1:rtfField001/text()
[2]">new
value</replace>
```

2. Remove an item from a collection, in this case the first item:

```
<remove sel="ns4:projectFullDTO/ns4:sites[1]"/>
```

3. Add two items, in this case two sites:

```
<add sel="ns4:projectFullDTO">
  <ns4:sites>
    <ns2:code>A0006-00001</ns2:code>
```

```

        <ns2:supplierCode>A0006</ns2:supplierCode>
    </ns4:sites>
    <ns4:sites>
        <ns2:code>A0006-00002</ns2:code>
        <ns2:supplierCode>A0006</ns2:supplierCode>
    </ns4:sites>
</add>

```

4. Replacing an existing value with a code to a glossary entry:

```

<replace sel="/ns4:activityFullDTO/ns4:activitySubStatus">
    <ns4:activitySubStatus>
        <ns6:code>ARTWORK</ns6:code>
    </ns4:activitySubStatus>
</replace>

```

This example is updating a code value which is a key to glossary record, in this case the Activity Sub Status. For this aggregation type relationship, the code must be specified as shown above. Specifying the code in the following way will result in an incorrect update:

```

<replace sel="/ns4:activityFullDTO/ns4:activitySubStatus/ns6:code">
    <ns6:code>ARTWORK</ns6:code>
</replace>

```

Response Details

The Process record is updated. All updatable values will be replaced with the values in the request.

If successful, an HTTP 200 response is sent with a body containing a ProjectRecordLink element. The ProjectRecordLink element consists of the returned elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Process record's internal ID
recordLink	string	URI to the Process record
projectId	string	Code ID business key to the Process record
projectName	string	Process record's description

Error Responses

If the supplied data does not result in a valid Process, an HTTP 417 response is sent with an ErrorMessage/Message message stating the validation errors. The request should not be reattempted with the same content.

If the validation fails, no updates are made for the request.

Error Messages

Element	Message	Meaning
id	<i>Process Record not found in the portal, based on Process ID</i>	Cannot locate the Process record

Element	Message	Meaning
projectType	Process Type with code {0} is not found in the portal	Invalid process type

ActivityRestService

This section describes the API for managing process activities. The following functions are available:

- [List of Values](#): retrieves a list of activities
- [Retrieve Record by ID](#): retrieves an Activity record using its unique identifier
- [Retrieve Record by Business Key](#): retrieves an Activity record using its business key
- [Check Record Modification Timestamp](#): retrieves the timestamp when an Activity record was last updated
- [Update Record](#): updates an existing Activity record
- [Update Status](#): updates the status of an existing Activity record

List of Values

Description

Retrieves a list of activities in a paged list. Use this function to locate Activity record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/activity **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
offset	Optional	int	No	No	60
pageSize	Optional	int	No	No	30
recordId	Optional	string	Yes	Yes	1
projectId	Optional	string	Yes	Yes	PR0000007
projectName	Optional	string	Yes	Yes	New Potato Product Launch
activityName	Optional	string	Yes	Yes	Artwork
activityCode	Optional	string	Yes	Yes	762
activityStatus	Optional	string	Yes	Yes	STARTED~COMPL ETED
activitySubStatus	Optional	string	Yes	Yes	MESSAGE SENT
activityType	Optional	string	Yes	Yes	PRODUCT_ARTW ORK

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Wildcard Search Supported?	Example
templateCode	Optional	string	Yes	Yes	ARTWORK
linkedRecord	Optional	string	Yes	No	Note: Options are: ASSESSMENT SUPPLIER SITE AUDIT PRODUCT_RECORD PRODUCT_SPECIFICATION
personResponsible Name	Optional	string	Yes	Yes	John Smith
proposedStartDate	Optional	string	No	No	2015-05-19
proposedEndDate	Optional	string	No	No	2015-05-19
actualStartDate	Optional	string	No	No	2015-05-19
actualEndDate	Optional	string	No	No	2015-05-19
sequenceNumber	Optional	string	Yes	Yes	1.0
modifiedSince	Optional	string	No	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	No	2015-05-19 13:30:39

Example URLs

.../services/rest/user/?offset=2&pageSize=20 .../services/rest/user/?
activityStatus=STARTED~COMPLETED

Response Details

For a successful response, XML is returned with a ProjectActivityLinkList root element containing an entries element for each matched process. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Process Activity record's internal ID
recordLink	string	URI to the ActivityRestService Retrieve record service for this activity
activityCode	string	Process Activity's business key
projectRecordId	string	The parent Process's internal ID
projectId	string	The parent Process's business key

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

In the event that an error occurs, an HTTP 500 response is sent.

Retrieve Record by ID

Description

Retrieves a single Activity record's details using the record's internal unique ID (which is not visible in the UI). Use this function to retrieve the full details of an individual process activity.

Endpoint address: /services/rest/activity/{id} **HTTP method:** GET

Request Details

The URL contains the {id} parameter that determines the record to retrieve.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
locale	Optional	string	Yes	en_GB~fr Locale country code. If used, fields that have language translations have a localeData element returned for the specified locales if data is present.

Example URL

.../services/rest/activity/5

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of a projectActivityFullDTO root element containing the individual attributes of the requested Activity record.

File Structure

The main elements of the projectActivityFullDTO root element:

- activityTemplateSnapshot and Template - the settings from the associated template.
- briefs - the associated process briefs; a separate element for each.
- linkedRecords - the associated linked records; a separate element for each.
- udfData - the contents of any user-defined custom fields associated to the Activity record.
- attachments - a child element containing the ids of any file attachments the activity has.

If an activity is linked to a record such as an Audit, the linkedRecords element contains a list of the linked records, containing the id and business key of the record (audit code, supplier code, or site code), its type, name and status, and the date it was created.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 3 - Process* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not found	Invalid {id} - not numeric
id	HTTP 417 IllegalStateException: Cannot find the Activity to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key

Description

Retrieves a single Activity record's details using its business key (code) and its parent process's code. Use this function to retrieve the full details of an individual activity using its key code.

Endpoint address: /services/rest/activity/byKey/{activityCode} **HTTP method:** GET

Request Details

There are no request parameters, but the URL contains the {activityCode} parameter that determines the record to retrieve.

Example URL

.../services/rest/activity/byKey/762

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of a projectActivityFullDTO root element containing the individual attributes of the requested Activity record.

File Structure

The main elements of the projectActivityFullDTO root element:

- activityTemplateSnapshot and Template - the settings from the associated template.
- briefs - the associated process briefs; a separate element for each.
- linkedRecords - the associated linked records; a separate element for each.
- udfData - the contents of any user-defined custom fields associated to the Activity record.
- attachments - a child element containing the ids of any file attachments the activity has.

If an activity is linked to a record such as an Audit, the linkedRecords element contains a list of the linked records, containing the id and business key of the record (audit code, supplier code, or site code), its type, name and status, and the date it was created.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 3 - Process* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
activityCode	HTTP 404 Not found	Invalid {activityCode} - blank or not found

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of an Activity record. Use this function to determine when the process activity was last updated.

Endpoint address: /services/rest/activity/{id} **HTTP method:** HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/activity/5

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header showing the date and time of the last update of the requested Activity record. For example:

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 417 Expectation Failed	Invalid {id} - not found
id	HTTP 404 Not found	Invalid {id} - not numeric

Update Record

Description

Updates an existing Process Activity record. Use this function to update an Activity in Supplier Evaluation based on data sourced from the external system.

Activity records can be updated, such as to set start/end dates, activity brief details, or other attributes, from an external system. Updates do not trigger any automatic re-scheduling of a Process. Dates and other values are simply updated.

The PATCH method allows for partial updates to a record, to update values without the need to submit the full record. This endpoint supports either an XML or JSON payload. The creation of new records is not supported.

Endpoint address: /services/rest/activity/{id}

HTTP method: PATCH

Request Details

The body of the request contains an activityFullDTO to specify the updates to the Activity record, which is based on the activityFullDTO type returned when retrieving an Activity.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supply Chain Collaboration Cloud Service Data Dictionary, Volume 3 - Process* for details of their mapping to the fields within the UI. The data dictionary indicates which fields are capable of being updated.

The Activity unique id is included in the request URI to identify the Activity to be updated. Each transaction will represent a single Activity. The payload need only include the mandatory values and the values of elements to be updated. Can update any updatable value in the Activity record, with the exception of the Activity Status field which can be updated using the Update Status endpoint.

Activities at Not Started, Started or Approval Required status can be updated; Completed Activities cannot be updated.

Duration (Days) is an integer. Completed By can be set to a valid user or to *Auto Complete*. Brief question answers can be populated by identifying the corresponding question.

Updating the Activity record does not trigger the Process to be re-scheduled, even if the activity durations are updated. Updates to the Supplier, Site and Process Manager are not supported. This means that the overall pattern of activity dates within a Process could appear differently to those generated by the system scheduling tool.



Note:

See the [Update Record](#) function of the ProjectRestService for examples of XML and JSON requests.

Response Details

The Activity record is updated. All updatable values will be replaced with the values in the request (excluding Activity Status).

If successful, an HTTP 200 response is sent with a body containing an ActivityRecordLink element. The ActivityRecordLink element consists of the returned elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Activity record's internal ID
recordLink	string	URI to the Activity record
activityCode	string	Code ID business key to the Activity record
projectId	string	Activity record's description

Error Response

If the supplied data does not result in a valid Activity, an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

If the validation fails, no updates are made for the request.

Error Messages

Element	Message	Meaning
code	<i>Activity not found in the portal, based on Activity Code</i>	Cannot locate the Activity record
status	<i>Activity is at Completed status</i>	Cannot update if already completed
subStatus	<i>Sub Status with code {0} is not found in the portal</i>	Invalid sub status
completedBy	<i>Completed By User {0} is not found in the portal</i>	Invalid user

Update Status

Description

Updates the status of an existing Process Activity record. Use this function to update an Activity's status from the external system, triggering any further updates to the associated Process or to other Activities within the Process.

Two alternative endpoints are available, to locate the Activity either by its record id or by its Code business key. They both return a record link to the Activity. These endpoints support a JSON payload.

Endpoint address: /services/rest/activity/{id}/status and /services/rest/activity/byKey{activityCode}/status

HTTP method: PUT

Request Details

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supply Chain Collaboration Cloud Service Data Dictionary, Volume 3 - Process* for details of their mapping to the fields within the UI. The data dictionary indicates which fields are capable of being updated.

The Activity unique id is included in the request URI to identify the Activity to be updated. Each transaction will represent a single Activity. The payload will only include the Activity Status to be updated.

If completing an Activity the Completed Date and Completed By fields can be updated using the Update Record endpoint.

Activities at Not Started, Started or Approval Required status can be updated; Completed Activities cannot be updated.

Update of the status will trigger the processing to apply updates to the associated Process or to other Activities within the associate Process, including:

- Update of keys and gates when the Activity is Completed.
- Auto-completion of Gates.

- If the last Activity in the Process is set to Completed, the Process Status is set to Completed.

Response Details

The Activity Status is updated. All updatable values will be replaced with the values in the request (excluding Activity Status).

If successful, an HTTP 200 response is sent with a body containing an ActivityRecordLink element. The ActivityRecordLink element consists of the returned elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Activity record's internal ID
recordLink	string	URI to the Activity record
activityCode	string	Code ID business key to the Activity record
projectId	string	Activity record's description

Error Responses

If the supplied data does not result in a valid Activity, an HTTP 417 response is sent with an ErrorMessage/Message message stating the validation errors. The request should not be reattempted with the same content.

Error Messages

Element	Message	Meaning
code	<i>Activity not found in the portal, based on Activity Code</i>	Cannot locate the Activity record
status	<i>Activity is at Completed status</i>	Cannot update if already completed
status	<i>Status with code {0} is not found in the portal</i>	Invalid status

ProductRecordRestService

This section describes the API for managing Product records. The following functions are available:

- [List of Values](#): retrieves a list of Product records
- [Retrieve Record by ID](#): retrieves a Product record using its unique identifier
- [Retrieve Record by Business Key](#): retrieves a Product record using its business key
- [Check Record Modification Timestamp](#): retrieves the timestamp when a Product record was last updated
- [Create Record](#): creates a new Product record
- [Update Record](#): updates an existing Product record

**Note:**

When integrated with Merchandising, using the ProductRecordRest API to make changes that affect the Product Codes table can have an impact on the integration.

List of Values

Description

Retrieves a list of Product records in a paged list. Use this function to retrieve a simple list of product titles and IDs, or to locate Product record IDs prior to a retrieve or update operation. Parameters are available to apply specific selection criteria for filtering the returned records.

Endpoint address: /services/rest/productRecord **HTTP method:** GET

Request Details

Parameters are passed as URI parameters.

URI Parameters

Parameter Name	Mandatory/ Optional	Value Type	Multiple Value Separator (~) Supported?	Example
offset	Optional	int	No	60
pageSize	Optional	int	No	30
modifiedSince	Optional	string	No	2015-05-19 13:30:39
modifiedUntil	Optional	string	No	2015-05-19 13:30:39
productRecordCode	Optional	integer	No	123456
altProductNumber	Optional	string	No	ABC001
division	Optional	string	Yes	DIV_A~DIV_B Must be the code of an entry in the Divisions glossary

Example URLs

.../services/rest/productRecord/?offset=2&pageSize=20.../services/rest/productRecord/?
productRecordCode=123456

Response Details

For a successful response, XML is returned with a ProductRecordLinkList root element containing an entries element for each matched Product Record. The entries element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Product record's internal ID

Element	Type	Description
recordLink	string	URI to the ProductRecordRestService Retrieve record service for this Product record
code	string	Code business key to the Product record
title	string	Product's title

The returned XML also contains a totalRecords element, which states the total number of retrievable records that match the filter parameters.

Error Messages

Element	Message	Meaning
modifiedSince	Invalid dates cause stack trace to come back to "response"	
modifiedUntil	Invalid dates cause stack trace to come back to "response"	
pageSize	Error Code: INVALIDPAGESIZE	Page Size must be between 1 and 100
offset	Error Code: INVALIDOFFSET	Offset must be a positive integer

Retrieve Record by ID

Description

Retrieves a single Product record's details using the record's internal unique ID (which is not visible in the UI). Use this function to retrieve the full details of an individual Product Record.

Endpoint address: /services/rest/productRecord/{id} **HTTP method:** GET

Request Details

The URL contains the {id} parameter that determines the record to retrieve.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 5 - Product* for details of their mapping to the fields within the Supplier Evaluation UI.

URI Parameters

Parameter Name	Mandatory/Optional	Value Type	Multiple Value Separator (~) Supported?	Example
locale	Optional	string	Yes	en_GB~fr Locale country code. If used, fields that have language translations have a localeData element returned for the specified locales if data is present.

Example URL

.../services/rest/productRecord/99

Response Details

For a successful response, an HTTP 301 message is returned, which is automatically acted upon to retrieve XML consisting of a productRecordFullDTO root element containing the individual attributes of the requested Product record. If an ID is not specified, a list of all Product records is returned (per the List of Values function).

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 5 - Product* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 IllegalStateException: Cannot find the Product Record to return with id:<###>	Invalid {id} - not found

Retrieve Record by Business Key

Description

Retrieves a single Product record's details using its business key (code). Use this function to retrieve the full details of an individual Product record using its Supplier Evaluation product code.

The Product record's business key is the product code that is visible in the UI. This an individual number assigned by Supplier Evaluation. It is not the product's traded unit identifier such as a SKU, that is, the product number, which is also visible in the UI.

Endpoint address: /services/rest/productRecord/byKey/{code} **HTTP method:** GET

Request Details

There are no request parameters, but the URL contains the {code} parameter that determines the record to retrieve.

Example URL

.../services/rest/productRecord/byKey/31

Response Details

If successful, an HTTP 301 redirect response is sent containing a header Location with the URI of the document requested.

The returned URI represents the call to be made to the Retrieve Record by ID function (with the id substituted with the relevant value), which will return a productRecordFullDTO root element containing the individual attributes of the requested Product record.

**Note:**

If your method of calling the API is configured to automatically follow HTTP 301 redirects, the call may automatically return a productRecordFullDTO root element containing the individual attributes of the requested Product record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 5 - Product* for details of their mapping to the fields within the Supplier Evaluation UI.

Error Messages

Element	Message	Meaning
code	HTTP 404 Not Found	Invalid {code} - blank
code	HTTP 404 Not Found	Invalid {code} - not found
code	NumberFormatException: For input string: <###>	Invalid {code} - not numeric

Check Record Modification Timestamp

Description

Retrieves the last modification date and time of a Product record. Use this function to determine when a Product record's details were last updated.

Endpoint address: /services/rest/productRecord/{id} **HTTP method:** HEAD

Request Details

There are no request parameters, but the URL contains the {id} parameter that determines the record to retrieve.

Example URL

.../services/rest/productRecord/99

Response Details

If successful, an HTTP 200 response is sent containing the Last-Modified header to showing the date and time of the last update of the requested Product record.

```
HTTP/1.1 200 OK
Date: Wed, 13 Jul 2016 07:52:14 GMT
Last-Modified: Fri, 08 Jul 2016 06:44:46 GMT
Content-Type: application/xml
Content-Length: 0
```

Error Messages

Element	Message	Meaning
id	HTTP 404 Not Found	Invalid {id} - not numeric
id	HTTP 417 Expectation Failed	Invalid {id} - not found

Create Record

Description

Creates a new Product record. Use this function to create new Product records in Supplier Evaluation based on data sourced from the external system.

Dependencies: If linked to a supplier or site, the Supplier/Site must be present in the application and its record ID obtained. If assigning a Business Category, Technologist, or Other Contact, the record must be present in the application and its record ID obtained.

Technologist and Other Contacts may be omitted to default to TBC if the TBC user is present in the application. For the status to be Active, the Product record must be linked to a supplier. For more information, see [Dependencies](#).



Note:

The Use Alternative Code for Product API system parameter enables a facility for holding alternative supplier codes against Supplier records. This feature is of use if the external system does not hold the Supplier Evaluation supplier codes. If enabled, and the supplier code submitted by the API is not found in Supplier Evaluation, a secondary lookup will be performed to attempt to locate the Supplier using the alternative codes facility.

Endpoint address: /services/rest/productRecord **HTTP method:** POST

Request Details

The body of the request contains a ProductRecordFullIDTO to specify the Product record to create. Compared to retrieving a Product record (which uses the same ProductRecordFullIDTO type), this request is much shorter. Only the attributes that are to be populated on the created Product record need to be included. As a minimum, this must include the fields shown in the following table.

Product Record Mandatory Fields

Field Name	Element Name
Product Title	title
Status	productRecordStatus
Variant Name	variantName
Quantity	quantity
Technologist	technologist
Other Contacts	productRecordOtherUser
Supplier Details	supplier
Site Details	site
Deleted Record	deleted (Set to false)
Language	label18NKey

**Note:**

To ensure a valid unique business key code is automatically assigned to the record being created, a value of zero must be passed in the <code> element of the XML when submitting the POST request.

Example Request XML

Where the record is linked to another record, such as the Technologist in this case, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

**Note:**

The API does not apply status-based validation as is applied using the Supplier Evaluation UI when manually progressing a Product record.

Single Variant Mode

```
<productCovered>
  <productTitle/>
  <variantName/>
  <quantity/>
  <retailerProductNumber/>
</productCovered>
```

Multi Variant Mode

```
<productCovered>
  <productTitle/>
  <variantName/>
  <quantity/>
  <retailerProductNumber/>
</productCovered>
<productCovered>
  <productTitle/>
  <variantName/>
  <quantity/>
  <retailerProductNumber/>
</productCovered>
```

Where the record is linked to another record, such as the Technologist in this case, the business key must be provided (not the description) in order to form the link between the records. In general, the business key will be the code attribute of the linked record.

**Note:**

The API does not apply status-based validation as is applied using the Supplier Evaluation UI when manually progressing a Product record.

It is therefore possible, using the API, to create a Product record at Active status without the supplier or site being populated.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 5 - Product* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a ProductRecordLink root element. The root element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Newly created Product record's internal ID
recordLink	string	URI to the newly created Product record, for use in a GET request
code	string	Code business key to the newly created Product record
title	string	Product's title

Error Messages

If the supplied data does not result in a valid Product record (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

Element	Message	Meaning
productCoverage	ERROR: class <<hostname>>.ProductRecord.productCovered - The condition is invalid	No tag provided
code	ERROR: class <<hostname>>.ProductRecord.code - A Product Record's code is mandatory	No tag provided
code	ERROR: class <<hostname>>.ProductRecord.code - A Product Record's code is mandatory	No value provided
code	ERROR: class <<hostname>>.ProductRecord.code - A Product Record's code must be unique:<###>	Code already exists
code	HTTP Status 400 - Bad request	Code is not numeric
title	DtoValidationException: title is mandatory	No tag provided No value provided
productRecordStatus	HTTP 417 IllegalStateException: Cannot locate keyword for class <<hostname>>.PoductRecordStstus using status:<###>	No tag provided No value provided Status not found
variantName	ERROR: class <<hostname>>.ProductRecord.productCovered[###].variantName - Variant name is mandatory	No tag provided No value provided
quantity	ERROR: class <<hostname>>.ProductRecord.productCovered[###].quantity - Quantity name is mandatory	No tag provided No value provided

Element	Message	Meaning
technologist	MySQLIntegrityConstraintViolationException: Cannot add or update a child row: a foreign key constraint fails ('<###>'.t_prdrctd,CONSTRAINT '<###>' FOREIGN KEY ('fk_technologist') REFERENCES 't_user' ('c_id'))	No tag provided No value provided
technologist	ObjectNotFoundException: No row with the given identifier exists: [com.micros.creations.core.domain.User#<###>]	Technologist not found
productRecordOtherUser/ user	DtoValidationException: ProductRecordOtherUser has no valid user for role:<###> ProductRecordOtherUser Roles are mandatory for the following missing Roles: [<###>]	No tag provided User not found
productRecordOtherUser/ user	ObjectNotFoundException: No row with the given identifier exists: [<<hostname>>.User#<###>]	No value provided
productRecordOtherUser/ role	DtoValidationException: ProductRecordOtherUser has no valid user for role:<###> ProductRecordOtherUser Roles are mandatory for the following missing Roles: [<###>]	No tag provided User not found
productRecordOtherUser/ role	ObjectNotFoundException: No row with the given identifier exists: [<<hostname>>.Role#<###>]	No value provided
supplier	HTTP 417 IllegalStateException: Cannot locate keyword for class <<hostname>>.Company using code:<###>	No tag provided
supplier	MySQLIntegrityConstraintViolationException: Cannot add or update a child row: a foreign key constraint fails ('<###>'.t_prdrctd,CONSTRAINT '<###>' FOREIGN KEY ('fk_supplier') REFERENCES 't_company' ('c_id'))	No value provided Supplier not found
site	HTTP 417 IllegalStateException: Cannot locate keyword for class <<hostname>>.Site using id:<###>, code:<###>, supplierCode:<###> and supplierId:<###>	No tag provided No value provided Supplier not found
countryWhereSold	HTTP 417 IllegalStateException: Cannot locate keyword for class <<hostname>>.CountryWhereSold using code:<###> and spec type:<###>	No value provided
countryWhereSold	ObjectNotFoundException: No row with the given identifier exists: [<<hostname>>.CountryWhereSold#<###>]	Country not found

Element	Message	Meaning
	ERROR: class <<hostname>>.ProductRecord.barcode[<###> - Bar Code must be either 8 or 13 characters long	Invalid code length
	ERROR: class<<hostname>>.ProductRecord.shippingCaseCode[<###> - Shipping Case Code must be either 14 digits long	Invalid code length

Update Record

Description

Updates an existing Product record. Use this function to update a Product record's details in Supplier Evaluation based on data sourced from the external system.

Endpoint address: /services/rest/productRecord/{id} **HTTP method:** PUT

Request Details

The body of the request contains a ProductRecordFullDTO to specify the updates to the Product record. Compared to retrieving a Product record (which uses the same ProductRecordFullDTO type), this request is much shorter. As a minimum, the values specified as mandatory for the Create Record function (see above) must be included.

The request content is the same as that for creating a Product Record. After the call, the Product Record is updated to match the request.



Note:

When updating records, all values must be included. If a value or element is omitted from the request, the field contents will be cleared on the Product record.

See the associated WADL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 5 - Product* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing a ProductRecordLink element. The ProductRecordLink element consists of the elements shown in the following table.

Returned Elements

Element	Type	Description
recordId	long	Product record's internal ID
recordLink	string	URI to the Product record, for use in a GET request
code	string	Code business key to the Product record
title	string	Product's title

Error Responses

If the supplied data does not result in a valid Product record (such as a missing mandatory field), an HTTP 417 response is sent with an ErrorMessage/Message XML body message stating the validation errors. The request should not be reattempted with the same content.

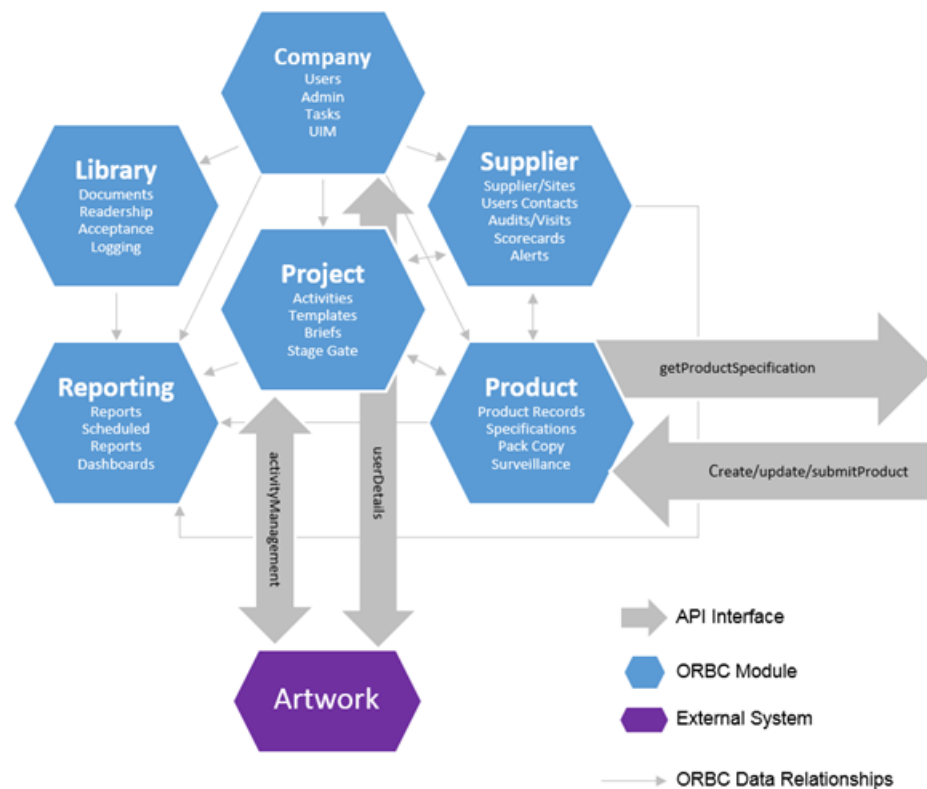
5

SOAP APIs

The following SOAP APIs are available:

- [Project Activities](#): Project Activities

Figure 5-1 Overview of SOAP APIs



Parameters and Filtering

Various parameters can be included in calls to the APIs, generally to define what data is to be returned, but can also control how records are returned. The following table lists some commonly used parameters.

Common Parameters

Parameter	Type	Description
offset	int	Used with pageSize to control the paging of a returned list of records. Specifies the starting point for the retrieval of records. If not specified, zero is assumed. For example, to retrieve 150 records: 1. A call with offset = 0 and pageSize = 50 returns the first 50 records. 2. Then, a call with offset = 50 and pageSize = 50 returns the next 50. 3. Then, a call with offset = 100 and pageSize = 50 returns the final 50.
batchSize	int	Used with offset to control the paging of a returned list of records. Specifies the number of entries in each page of returned list of records. If not specified, 30 is assumed. The maximum is 100.



Note:

Date/Time parameters must be provided in the YYYY-MM-DD hh:mm:ss format.

When retrieving records, the returned XML only includes elements that actually contain data; empty elements are omitted.

The following example provides the first 50 Product Specifications that changed status to either Active or Off Range in May 2016. Searching with either fromDate or toDate specified without specStatus, will filter based on the last amended date of the Product Specification.

```
<soapenv:Envelope xmlns:soapenv="<<hostname>>/soap/envelope/" xmlns:v1="<<servicename>>/">
  <soapenv:Header/>
  <soapenv:Body>
    <v1:getProductSpecification>
      <request>
        <batchsize>50</batchsize>
        <offset>0</offset>
        <specStatus>ACTIVE</specStatus>
        <specStatus>OFF_RANGE</specStatus>
        <fromDate>2016-05-01T00:00:00</fromDate>
        <toDate>2016-05-31T24:00:00</toDate>
      </request>
    </v1:getProductSpecification>
  </soapenv:Body>
```

Project Activities

This section describes the API for Artwork integration. The following function is available:

- **Activity Management:** updates existing Artwork activities

Activity Management

Description

Updates the sub status of existing Artwork Project Activities. This function is used for the integration of Supplier Evaluation with an external Artwork Management system, to update the status of an artwork activity from the external system.



Note:

This service has been superseded by the Artwork Restful Service.

Endpoint address: /services/activityManagement **HTTP method:** updateActivityStatus

Request Details

The body of the request contains an updateActivityStatus/activityUpdateRequest element consisting of the elements shown in the following table.

Request Elements

Element	Type	Description
activityName	string	Project activity's name
activityRecordId	string	Project activity record's internal ID
projectId	string	ID business key to the Project record
projectRecordId	string	Project record's internal ID
subStatusCode	string	Code business key to the activity's sub status

After the call, the Project Activity record's sub status is updated to match the request.

See the associated WSDL for a full list of the attributes, and the *Oracle Retail Supplier Evaluation Cloud Service Data Dictionary, Volume 3 - Process* for details of their mapping to the fields within the Supplier Evaluation UI.

Response Details

If successful, an HTTP 200 response is sent with a body containing an updateActivityStatusResponse root element, consisting of a return element populated with SUCCESS.

Error Messages

If an error occurs, the updateActivityStatusResponse root element's return element will contain FAILURE.

Appendix: Secure Development Guide

The web service APIs provided by Supplier Evaluation are designed to be secure by default. When consuming these services, there are recommended best practices for the clients which call them. These development and usage guidelines are provided to maintain secure use, and preserve integrity throughout the system domain.

The Supplier Evaluation web services are secured using OAuth 2.0. In order to use them, calling clients require an External System within Supplier Evaluation and a corresponding Application within IDCS with which to authenticate. Transport encryption is enabled using TLS v1.2 as a minimum standard.

Best Practices for Web Service Use

This section describes some best practices.

Store External System Credentials Securely

In order to grant access to web services, credentials must be set up within Supplier Evaluation in an External System record and a corresponding Application record in IDCS. These credentials, used by the client to call web services, should be managed securely in the client's domain. It is advised not to store passwords in plaintext, either in a file or a database table.

Configure External Systems Permissions Appropriately

The permissions of External Systems are configurable, and determine which services may be accessed. The permissions model governs not only the service which is accessible (for example, The Product Service), but also the endpoints (or actions) permissible on that service (for example, Read a Product or Add a Product).

It is recommended that External Systems are set up with the minimum authorities required to perform their required function. External System records should have granular permissions wherever feasible, rather than be granted access to many disparate functions. This minimizes the level of disclosure possible if credentials are compromised.

Maintain Strong Passwords for External System Records

Configurable parameters within the Supplier Evaluation Oracle Identity Cloud Service (IDCS) or Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) tenant govern the strength of passwords used for External Systems and the rules surrounding their expiry. These parameters are set to values which strike a suitable balance between usability and security by default.

It is advised that should a client wish to change any of these parameter values, it should only be to increase security. Changes which slacken security are *not* recommended.

Those responsible for maintaining clients which consume services should note that passwords for External Systems expire in the same manner as those for regular application users.



Note:

A process for updating the credentials used by the client should be put in place, as IDCS or OCI IAM does not send automated notifications of pending password expiry.

For full details of this functionality, and of External Systems in general, see the *Oracle Retail Supplier Evaluation Cloud Service Administration Guide*.

Make Use of the Web Service Log

Intrusion detection mechanisms are employed within the hosted environment of Supplier Evaluation to capture to prevent malicious behavior at the network level. In addition to this, protections against XML injection attacks exist at the logical tier.

The application also provides a Web Service Log for users with sufficient permissions to see what calls have been made to its services. This provides a user-centric audit of the web service activity within the application, and can help clients monitor usage, or gather metrics to understand usage behavior further.

Ensure Your Client Supports TLS v1.2 or Higher

All Supplier Evaluation deployments require communication over HTTPS, and use TLS v1.2 as a minimum encryption protocol. Clients of the web service must be developed within a technology which supports TLS v 1.2 or higher. For example, Java 7 does not support TLS v1.2 out of the box.

Limit the Size of Web Service Payloads

The web services are designed for continued, real-time use. It is recommended that calls to the services are made in a manner to retrieve or send only what is required.

Where larger result sets are necessary however, provision is made within the web service APIs to allow paging of data. By providing an offset and a page size, it is possible to retrieve a page of results at a time.

Appendix: Using the Data Privacy API

The Supplier Evaluation Data Privacy web service API provides a means for the retailer/portal owner to execute data privacy requests. The requests are classified as Right to Access, where an individual requests details of all their personal data that is held in the system; and Right to be Forgotten, where the individual requests that their personal data be erased from the system.

**Note:**

The provision of the Data Privacy API is a security enhancement which aids the retailer/portal owner to carry out data privacy requests.

It is the responsibility of the retailer/portal owner to manage the fulfilment of data privacy requests. In order to do so, they will need to build an application to call the Data Privacy API, and handle the returned data. The API will be permanently available for access on demand.

The retailer/portal owner's responsibilities include verifying the identity of the individual making the request, presenting the information back to them in a secure manner, and deleting any local copies of the data; also the configuration of the portal's Terms and Conditions to include statements about the consent to store personal data, and any opt-out procedure.

This appendix provides additional detail to support the [DataPrivacyService](#) section of this document. It covers the following areas:

- [Definition of Personal Data](#): an explanation of some of the terminology used, and the type of data within Supplier Evaluation that is considered to be personal data.
- [Personal Data Flow](#): an overview of the flow of data within Supplier Evaluation which is considered to be personal data.
- [Right to Access Requests](#): further detail on this type of request, with an example of its execution.
- [Right to be Forgotten Requests](#): further detail on this type of request, with an example of its execution.
- [Other Aspects of Data Privacy](#): other aspects of Data Privacy and how they relate to Supplier Evaluation.

**Note:**

The Supplier Evaluation and Supply Chain Hub modules have separate APIs for managing data privacy requests, such as GDPR right to access and right to be forgotten requests. For more details, see the *Oracle Retail Supply Chain Collaboration Cloud Service Integration User Guide*.

Definition of Personal Data

The following terms are used to describe the actors involved in the handling and processing of personal data:

- **Data Subject** - The person whom the personal data describes. In Supplier Evaluation, this refers to an employee of the retailer/portal owner, an employee of a supplier, or an employee of a third party such as a design agency or certification body.
- **Data Controller** - The retailer/portal owner. In a cloud environment, Oracle becomes the data custodian, but the retailer/portal owner retains ownership and control of the data.
- **Data Processor** - Applications that work with personal data, such as Supplier Evaluation, and any external applications that share its data.

The definition of personal data is any information that can be mapped to a unique user, whether the identifiable information is stored with the key or not.

The personal data held in Supplier Evaluation is business contact information, such as:

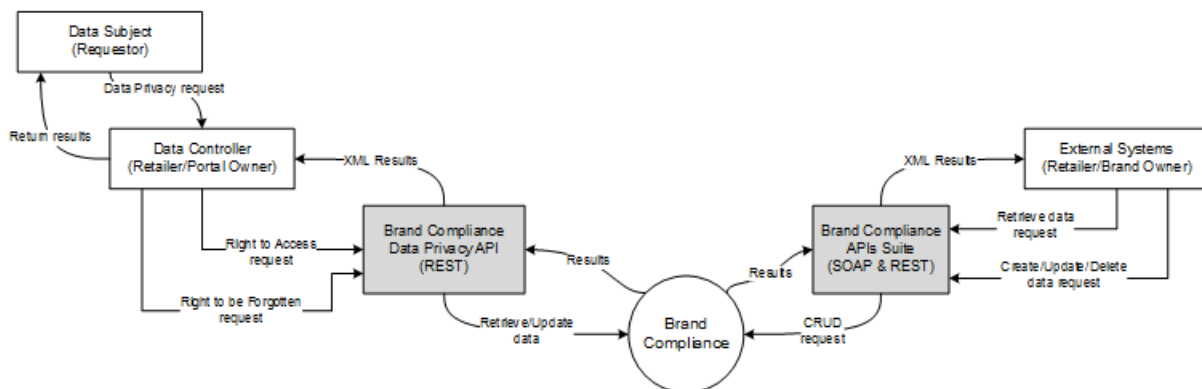
- Name of the data subject
- Contact addresses
- Email addresses
- Telephone and Fax numbers

A general assumption is that user-defined/custom fields and file attachments are not used to store personal data. This is applied as a policy constraint rather than specific controls within the system.

Personal Data Flow

The following diagram outlines the flow of personal data within Supplier Evaluation:

Figure 7-1 Personal Data Flow



The Data Subject (requestor) makes the request to the Data Controller (retailer/portal owner) who submits the request to the Supplier Evaluation Data Privacy API. The API performs the search or erase Supplier Evaluation processing accordingly and returns the results for the retailer/portal owner to relay back to the requestor.

The full suite of Supplier Evaluation APIs allows the retailer/portal owner to make requests from external systems to retrieve, create, update, (and in some cases delete) Supplier Evaluation data. The data handled by these APIs may include personal data.

Access to Supplier Evaluation data through the APIs is secured by an extension of the system's configurable Permissions security model. Based on the role-based access control model which restricts users' access to data and functionality based on their role within the system, external systems are granted access to the API services. If necessary, the access can be restricted to individual endpoints of the services. Authentication is achieved using individual ids and passwords.

**Note:**

Supplier Evaluation does not *push* data to external systems; the retailer/portal owner *pulls* the data. It is therefore the responsibility of the retailer/portal owner to trace and flag any personal data that has been taken from Supplier Evaluation, as appropriate.

Right to Access Requests

The Right to Access service facilitates the request from the data subject (individual requestor) to the data controller (retailer/portal owner) to provide details of what personal data concerning them is held, and to provide a copy of the data in electronic format.

The majority of personal data is held within User/Person records (an account for each user of the system) and Contact records (supplier users may be designated specific contact roles). There are also instances of personal data that are not linked to a user account, such as the names and email addresses of general contacts, who are not users of the system. The retailer/portal owner must decide which of that data returned by a search relates to the requestor.

When a retailer/portal owner receives a request, they use the *Right to Access* application they have developed to call the Data Privacy API. The API searches Supplier Evaluation and returns the results for the retailer/brand owner to analyze, and edit or reformat if necessary, before relaying back to the requestor.

Personal Data Searched and Returned

The following table describes the areas of the system that hold personal data, and which fields are searched and returned.

Personal Data

Record/ Element Name	Details
User <user>	Each user of the system has a User record and a Person record which contain their Name, Email, Phone, Mobile, Fax, and Address details. Each account has a Login id, which is unique to the user. Names and email addresses are not necessarily unique. Search Field: Name Returned Fields: Record Type; Name; Email; Phone; Mobile; Fax; Login; Address; User Type (RETAILER, SUPPLIER or SITE); User record id; Person record id

Record/ Element Name	Details
Contact <contact>	A supplier user may be designated a contact for the supplier or its sites. Contact records share data with the User and Person records; they contain the Name, Email, Alternative Email, Phone, Mobile, Fax, and Address fields. A contact may be an individual who is not a user of the system. Search Field: Name (where contact is flagged as not a user of the system) Returned Fields: Record Type; Name; Email; Alternative Email; Phone; Mobile; Fax; Address; Contact record id; Supplier Name; Supplier Code
Company <retailer>	The single Company record holds the main retailer/portal owner contact details of a Contact Name and associated Email, Phone, Fax, and Address fields. The contact may or may not be a user of the system. Search Field: Contact Name Returned Fields: Record Type; Contact Name; Email; Phone; Fax; Address; Company record id
Certification Body <certificationBody>	A glossary of Certification Bodies is used to assign to Third Party Audits. Each record contains a Contact Name and supporting Email, Phone, Fax, and Address, which refers to an individual at a third-party organization. Search Field: Contact Name Returned Fields: Record Type; Contact Name; Email; Phone; Fax; Address; Certification Body record id
Supplier <supplier>	The Supplier record contains a Contact Name and associated Email, Phone, Fax, and Address fields, which refer to an individual who may or may not be a user of the system. Search Field: Contact Name Returned Fields: Record Type; Contact Name; Email; Phone; Fax; Address; Supplier record id; Supplier Name; Supplier Code The Supplier record also contains billing details entered using the registration wizard, including the name of an Individual or Department, a Billing Contact Name, and associated Email, Phone, Fax, and Address. They refer to an individual who may or may not be a user of the system. Search Field: Individual or Department; Billing Contact Name Returned Fields: Record Type; Individual or Department; Billing Contact Name; Email; Phone; Fax; Address; Supplier record id; Supplier Name; Supplier Code
Audit/Visit <audit>	The Audit/Visit record may have free text Name entries in the People Present table for attendees who are not users of the system. Entries for users also store the name in a <i>snapshot</i> text field. The Audit/Visit Issue record may have non-conformance actions with a free text Assigned To Name or a free text Completed By Name. Audit Checklists also share the Assigned To Name when a non-conformance is being created for a Checklist. Search Field: Name in People Present table; Issue Assigned To Name; Issue Completed By Name Returned Fields: Record Type; Name, Assigned To Name; Completed By Name; Audit/Visit record id; Audit/Visit Template Name; Audit/Visit Code; Supplier Name; Supplier Code; Site Name; Site Code; Audit/Visit Issue record id; Audit/Visit Issue Code

Security and Logging

The following Service and Endpoint configurations must be configured within the Supplier Evaluation Admin area, and assigned to the External Systems that are to have access to the Right to Access service:

Service:	DATAPRIV	Data Privacy Service
Endpoint:	DATAPRIV_ACCESS_POST	Creates a new DataPriv request and returns a link to the record's details
Endpoint:	DATAPRIV_ACCESS_GET	Retrieves DataPriv details for a given id

All calls to the service will be logged in the Supplier Evaluation Web Service Log.

The Data Privacy records created by the service are not viewable in Supplier Evaluation, and are not purged.

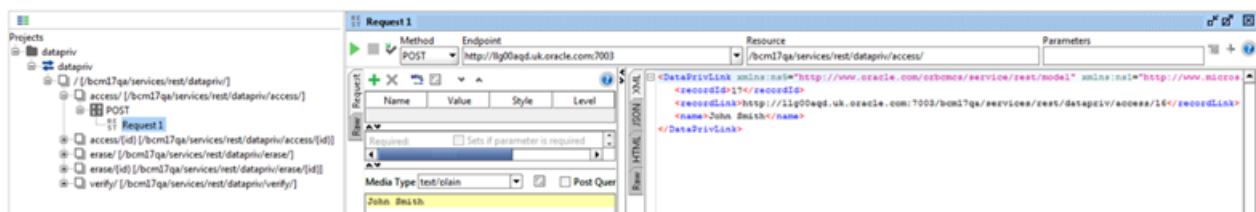
Example of Using the Right to Access Service

The request returns the actual data held, not the context in which it is used within the system. For example, to return the name and contact details of a Technologist, but not the fact that within the system what they are responsible for approving.

The stages involved in executing a Right to Access request are as follows. For this worked example, the call is made using a web service testing application.

1. Download the datapriv WADL definition from the /services area of the Supplier Evaluation portal.
2. Submit a POST call to the services/rest/datapriv/access service, passing the name of the individual in the payload.

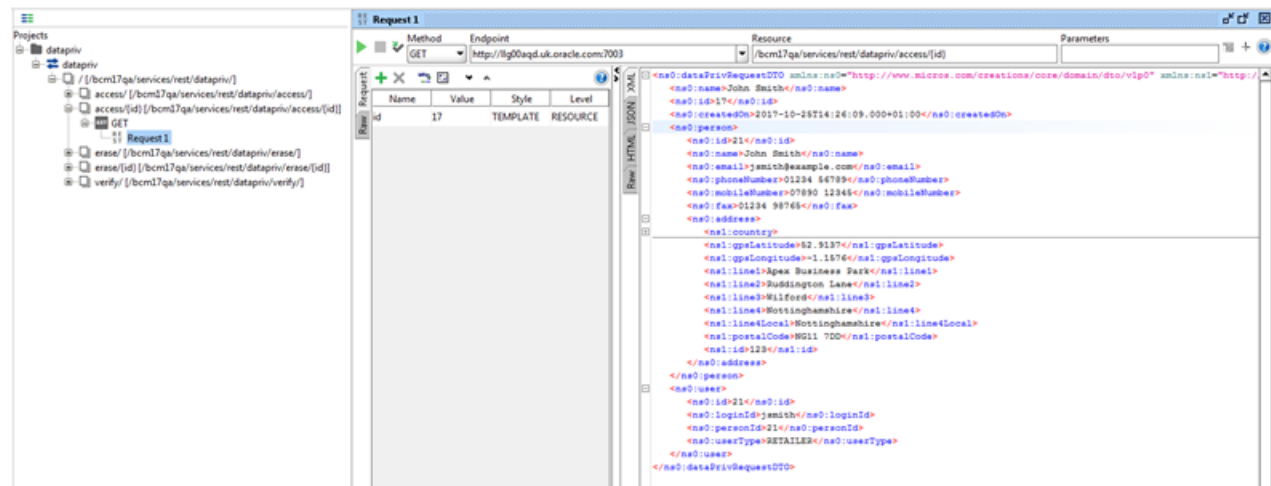
Figure 7-2 Access Post Request



A recordLink element is returned with a link to a Data Privacy record.

3. Submit a GET call to the services/rest/datapriv/access service, passing the returned recordId value as the id parameter.

Figure 7-3 Access Get Request



A message is returned containing the personal data that matches the name, within a `dataprivRequestDTO` root element. The structure of the XML message is as follows:

Figure 7-4 Access Returned Message



Returned Elements

Element Name	Details
name	The name of the individual as passed on the POST request.
id	The record id of the Data Privacy record created for this request.
createdOn	Timestamp of when the request was submitted.

Element Name	Details
person	Personal data fields from the Person records that match the name parameter.
user	Personal data fields from the User records that match the name parameter.
certificationBody	Personal data fields from the Certification Body records that match the name parameter.
retailer	Personal data fields from the Company/Retailer record that matches the name parameter.
supplier	Personal data fields from the Supplier record that match the name parameter.
site	Personal data fields from the Site record that match the name parameter.
contact	Personal data fields from the Contact records that match the name parameter.
audit	Personal data fields from the Audit/Visit and Issue records that match the name parameter.

The personal data fields are contained within tags that are similar to their (English) field labels in the UI:

```
<ns0:user>
  <ns0:id>7</ns0:id>
  <ns0:loginId>john smith</ns0:loginId>
  <ns0:name>John Smith</ns0:name>
  <ns0:email>jsmith-supplier@example.com</ns0:email>
  <ns0:userType>SUPPLIER</ns0:userType>
</ns0:user>
```

Additional metadata fields are included to identify the record by its code/id, name/description, and type.

Address elements may contain multiple localeData elements for country language translations:

```
<ns0:address>
  <ns1:country>
    <ns2:code>GB</ns2:code>
    <ns2:countryType>COUNTRY</ns2:countryType>
    <ns2:internetDomainSuffix>co.uk</ns2:internetDomainSuffix>
  <ns2:localeData>
    <ns1:description>Reino Unido</ns1:description>
    <ns1:locale>es</ns1:locale>
    <ns1:id>1904</ns1:id>
  </ns2:localeData>
```

There may be duplicates of the elements, where the system has more than one user with the same name, or where there are multiple records associated to an individual.

4. Edit the message using a text editor to remove duplicates, or entries that do not relate to the individual who made the request, and reformat if necessary, before delivering back to the requestor.

Right to be Forgotten Requests

The Right to be Forgotten service facilitates the request from the data subject (individual requestor) to the data controller (retailer/portal owner) to erase the personal data concerning

them. The erasure of data is subject to certain predefined conditions within the system, and to the discretion of the data controller as to if/what data may be erased. The erasure is not reversible.

Data is erased by obfuscation, that is, replacing the value with a string of random text.

When a retailer/portal owner receives a request, they use the *Right to be Forgotten* application they have developed to call the Data Privacy API. The API validates the request and erases the requested data from Supplier Evaluation. The retailer/portal owner then relays the result back to the requestor.

Personal Data Updated

Generally, personal data is erased from User and Contact records, with the obfuscation of names and email addresses being reflected elsewhere in the system wherever they are referenced. Personal data that is not linked to a User or Contact record may also be erased; the retailer/portal owner must make a judgement on whether the data relates to the individual making the request.

The following table describes which areas of the system may have data erased.

Erasable Personal Data

Record	Fields
Users	Name; Email; Phone; Mobile; Fax; Login; Address
Contacts	Name; Email; Alternative Email; Phone; Mobile; Fax; Address
Company	Contact Name; Email; Phone; Fax
Design Agency	Contact Name; Email; Phone; Fax; Address
Counter Ticket Producer	Contact Name; Email; Phone; Fax; Address
Certification Body	Contact Name; Email; Phone; Fax; Address
Supplier	Contact Name; Email; Phone; Fax Individual or Department; Billing Contact Name; Email; Phone; Fax in the Billing Details
Audit/Visit	Name in the People Present table Assigned To Name; Completed By Name in the Issue record

Text fields are replaced with randomly-generated text; the numeric GPS address fields are blanked; email addresses are set to x@example.com where x is 15 characters of randomly-generated text; countries and region selections are set to blank. If a field is blank, it remains unchanged.

If a User record is updated, the account is also deactivated and the password is expired.



Note:

The obfuscation of data in Supplier Evaluation as a result of a Right to be Forgotten erasure request is not reversible.

Personal data may also be updated using the individual APIs, or manually edited within the system. Physical deletion is not possible, due to referential integrity.

Exceptions

The key purpose of Supplier Evaluation is to assure due diligence in the management of the retailer/portal owner's products, and the suppliers thereof. There are legal or regulatory obligations to retain the approved technical specifications for a set number of years, and during that time they may be required as legal evidence as part of a product safety issue. Where the approval or authorization by an individual forms part of the due diligence evidence, it is not feasible for the individual to be *forgotten*.

The following table indicates which fields are considered to be authorizations, and are preserved for due diligence purposes. These exceptions are therefore not erasable.

Preserved Personal Data

Record	Field
Assessment	Completed By

Purging Inactive User Accounts

The Right to be Forgotten service may also be used for purging user accounts that are no longer active, that is, those of users who have left the organization or are no longer involved in using the system. This is achieved by calling the API in the same way as if processing a Right to be Forgotten request.

Automatic date-based purging, such as a period of time since the user's account was disabled, or since their last login, is not provided, however the retailer/portal owner could feasibly develop a means of recursively calling the API based on a set of logic rules.

Security and Logging

The following Service and Endpoint configurations must be configured within the Supplier Evaluation Admin area, and assigned to the External Systems that are to have access to the Right to be Forgotten service:

Service:	DATAPRIV	Data Privacy Service
Endpoint:	DATAPRIV_VERIFY_POST	Creates a new verification request for a login id
Endpoint:	DATAPRIV_ERASE_POST	Creates a new Erasure request and returns a link to the record's details
Endpoint:	DATAPRIV_ERASE_GET	Retrieves Erase details for a given id

All calls to the service will be logged in the Supplier Evaluation Web Service Log.

The Data Privacy records created by the service are not viewable in Supplier Evaluation, and are not purged.

Entries within the Supplier Evaluation Change History logs, Attachments logs, and Comments logs use a foreign key to reference the name of the user who updated the record, attached the file, or added the comment. This means that if the name of the user is obfuscated in the User record, the name will automatically become overwritten with randomly-generated text in the logs.

Updates to records to erase personal data through the Data Privacy API are not recorded in the record's Change History log. This avoids showing what the value was before and after obfuscation.

Example of Using the Right to be Forgotten Service

The stages involved in executing a Right to be Forgotten request are as follows. For this worked example, the call is made using a web service testing application.

1. Download the datapriv WADL definition from the /services area of the Supplier Evaluation portal.
2. Use the Right to Access service to retrieve the individual's set of personal data (see [Right to Access Requests](#)).
3. Edit the returned XML message using a suitable text editor, removing duplicate user and person elements.

Figure 7-5 XML for Forgotten Returned

```

1 <ns0:dataPrivRequestDTO xmlns:ns0="http://www.micros.com/creation
2 http://www.micros.com/creations/core/domain/dto/v1p0/full" xmlns:
3 <ns0:name>John Smith</ns0:name>
4 <ns0:id>17</ns0:id>
5 <ns0:createdOn>2017-10-25T14:26:09.000+01:00</ns0:createdOn>
6 <ns0:person>
7 <ns0:id>21</ns0:id>
8 <ns0:name>John Smith</ns0:name>
9 <ns0:email>jsmith@example.com</ns0:email>
10 <ns0:phoneNumber>01234 56789</ns0:phoneNumber>
11 <ns0:mobileNumber>07890 12345</ns0:mobileNumber>
12 <ns0:fax>01234 98765</ns0:fax>
13 <ns0:address>
14 </ns0:address>
15 </ns0:person>
16 <ns0:person>
17 <ns0:id>7</ns0:id>
18 <ns0:name>John Smith</ns0:name>
19 <ns0:email>jsmith-supplier@example.com</ns0:email>
20 <ns0:phoneNumber>09876 54321</ns0:phoneNumber>
21 <ns0:mobileNumber>07890 12345</ns0:mobileNumber>
22 <ns0:fax>09876 54320</ns0:fax>
23 <ns0:address>
24 </ns0:address>
25 </ns0:person>
26 <ns0:user>
27 <ns0:id>21</ns0:id>
28 <ns0:loginId>jsmith</ns0:loginId>
29 <ns0:personId>21</ns0:personId>
30 <ns0:userId>RETAILER</ns0:userId>
31 </ns0:user>
32 <ns0:user>
33 <ns0:id>7</ns0:id>
34 <ns0:loginId>john smith</ns0:loginId>
35 <ns0:name>John Smith</ns0:name>
36 <ns0:email>jsmith-supplier@example.com</ns0:email>
37 <ns0:userId>SUPPLIER</ns0:userId>
38 <ns0:address>
39 <ns1:id>36</ns1:id>
40 </ns0:address>
41 </ns0:user>
42 <ns0:designAgency>

```

In this example, there are two pairs of user and person elements because there are two user accounts with the name John Smith - one for a retailer user and one for a supplier user (user name and email address are not unique, it is the login id that uniquely identifies a user).

In order to pass validation, there must only be one user and person element in the submitted XML message. Remove any user and person elements that are not the individual being erased. Delete all the data between the opening and closing tags, just leaving a single person and user:

Figure 7-6 XML for Forgotten One Person

```

1 <ns0:dataPrivRequestDTO xmlns:ns0="http://www.micros.com/creation:
  http://www.micros.com/creations/core/domain/dto/vlp0/full" xmlns:
2   <ns0:name>John Smith</ns0:name>
3   <ns0:id>17</ns0:id>
4   <ns0:createdOn>2017-10-25T14:26:09.000+01:00</ns0:createdOn>
5   <ns0:person>
6     <ns0:id>21</ns0:id>
7     <ns0:name>John Smith</ns0:name>
8     <ns0:email>jsmith@example.com</ns0:email>
9     <ns0:phoneNumber>01234 56789</ns0:phoneNumber>
10    <ns0:mobileNumber>07890 12345</ns0:mobileNumber>
11    <ns0:fax>01234 98765</ns0:fax>
12    <ns0:address>
13  </ns0:person>
14  <ns0:user>
15    <ns0:id>21</ns0:id>
16    <ns0:loginId>jsmith</ns0:loginId>
17    <ns0:personId>21</ns0:personId>
18    <ns0:userType>RETAILER</ns0:userType>
19  </ns0:user>
20  <ns0:designAgency>

```

 Note:

The message is XML and can be edited using any UTF-8 compliant text editor.

It is vital that the structure and syntax of the message is maintained (with matching pairs of tags), so editing with a tool that recognizes the XML file format is recommended. For details of the message structure, see [Right to Access Requests](#).

Metadata fields that are included in the message to identify the records by their code/id, name/description or type are not erased. For details of which fields may be erased, see the [Erased Personal Data](#) table.

Person/User/Contact Relationships

There may be a combination of user, person, and contact elements for an individual, depending on their role within the system:

- Each user of the system has a single User record and a single Person record.
- If a supplier user is a contact, in addition to their User and Person records, they have a Contact record for each of their designated contact roles.
- If an individual is a contact but they are not actually a user of the system, they have a Person record, plus a Contact record for each of their designated contact roles.

When submitting a Right to be Forgotten request, the following validation checks are applied:

- Only one user element must be present.
- Only one person must be present.
- The associated person element must be present for a user or a contact element.
- The personId in the user element must match the id in the person element.
- The personId in the contact element must match the id in the person element.
- The user id in the contact element must match that in the user element.
- All contacts must be linked to the same person.

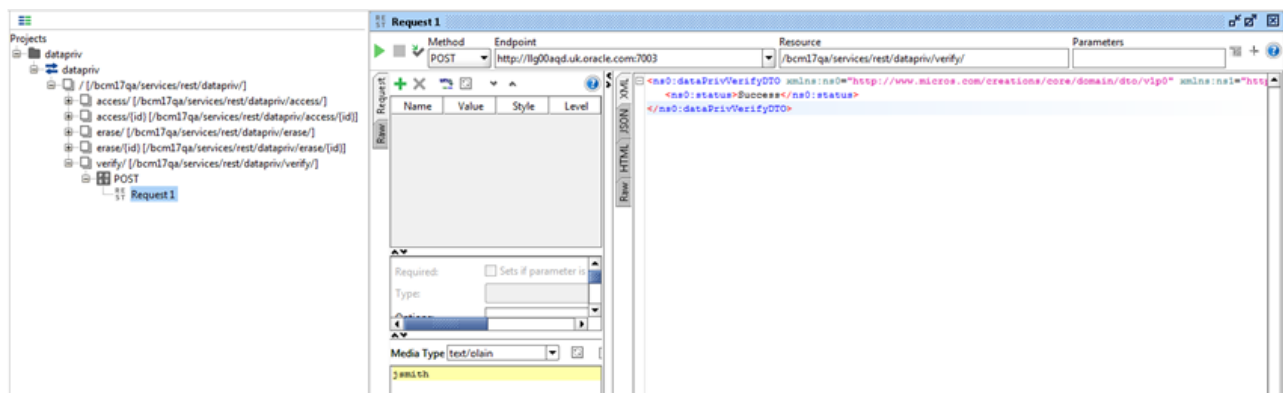
- A person element must have an associated user or contact element.
4. Make further edits to the XML message to check that the data in the elements for designAgency, supplier, and so on, all relate to the individual making the request. If not, delete the elements.

When submitted, the personal data elements that remain in the message will result in the corresponding fields being erased in Supplier Evaluation (the field contents are replaced with randomly-generated text).

If an element is removed, the field contents remain unchanged.

5. Optionally submit a POST call to the services/rest/datapriv/verify service, passing the login id of the individual (obtained from the User element of the XML message returned in Step 2) in the payload:

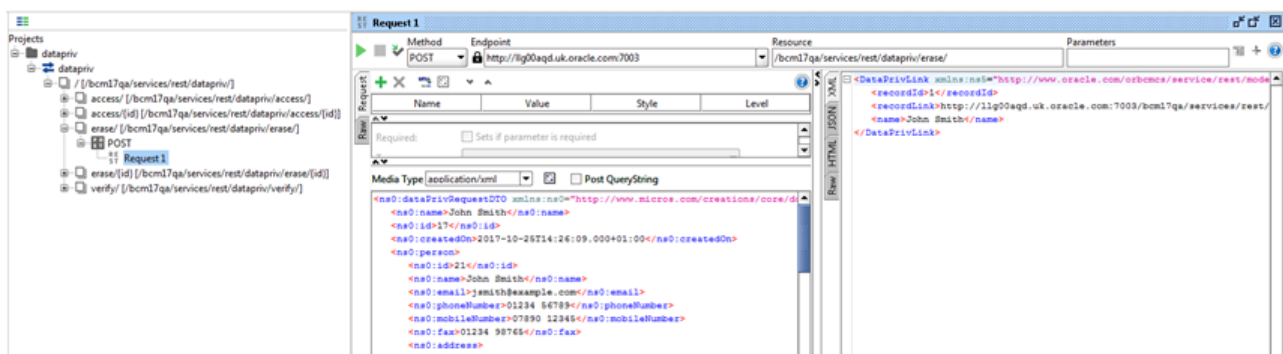
Figure 7-7 Forgotten Post Call



This step checks whether the individual is responsible for the completion of an Assessment, in which case they will not be permitted to be erased.

6. Submit a POST call to the services/rest/datapriv/erase service, passing the XML message in the payload:

Figure 7-8 Forgotten Post Second Call

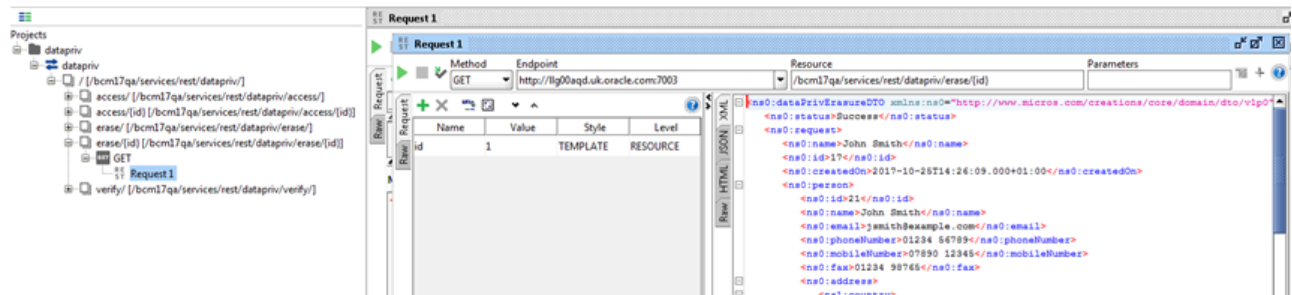


A recordLink element is returned with a link to a Data Privacy record.

For possible failure conditions, see the [Error Messages](#) table.

- Submit a GET call to the services/rest/datapriv/erase service, passing the returned recordId value as the id parameter:

Figure 7-9 Forgotten Get Call



A confirmation message is returned, with the original data in a dataprivErasureDTO root element.

- Optionally perform another a Right to Access request to check that the personal data has been erased as expected.

Otherwise, viewing the erased records within Supplier Evaluation will show the obfuscated data:

Figure 7-10 Forgotten Removed Data

User Details Custom Fields Change History Attachments			
Details			
Name:	auW4VXQvXJcxlPHOYVoTDCVNB0YwY66088K0mLEdpmYCrmmM	Email:	2QF8CYa8aCayfu@example.com
Login Id:	NWPubeoXHLPyommZ2fJl	Phone:	n40PPAmyJKue5Wakae5stR0z9hm1l
Login id Disabled?:	Yes	Mobile Phone:	bttsvokGXDAscBkmswXAMhEdU0gCiu
Password Expired?:	Yes	Fax:	0VCYxKw6zAx1WAlhRf0kRdt17POC
Job Title:	-		
Department:	-		
Team Manager:	-		
Delegates:	-		
Language:	English (British)		
Time Zone:	Greenwich Mean Time (Europe/London)		
Comments:	-		
Testing Support Enabled?:	-		
External Authentication:	No		
Address			
Local Address			
Country:	-		
Address 1:	nKmk3UoztExFo3Muqpg		
Address 2:	PuhoUWuHRS3Zbrymg		
City:	BvPhQlURToFYOnF23PLX		
County:	5CJ36dAU4CwRxb4AQY0m		
Post Code:	jS1uG		
GPS latitude:	-		
GPS longitude:	-		

- Advise the requestor of the outcome of their request.

Other Aspects of Data Privacy

This section describes some other aspects of Data Privacy, and how they relate to Supplier Evaluation.

**Note:**

The data passed to IDCS or OCI IAM includes personal information (user name and email address). The user may also manually enter personal contact information in IDCS or OCI IAM.

The Supplier Evaluation *Right to Access* and *Right to Erase/Right to be Forgotten* data privacy services are not extended to access IDCS or OCI IAM, so the retailer must also check for a corresponding user profile in IDCS or OCI IAM, and where necessary, remove the personal information from IDCS or OCI IAM.

When a new user is created in Supplier Evaluation, the User record has a flag set to indicate that its data has been passed to IDCS or OCI IAM.

Data Portability

Data Portability is the right for the data subject to receive the data concerning them, in a common, structured machine-readable format. The Right to Access service and other APIs provide access to the majority of Supplier Evaluation data. The Reports module also allows the export of data in various formats.

Privacy by Design and Data Minimization

Privacy by Design and Data Minimization relates to making data privacy and protection integral to the design of the system, such as minimizing the data set to that which is necessary to perform the function of the system, and restricting the access to personal data to just the users that need it. The personal data held in Supplier Evaluation is already minimal; it relates to business contact data.

Access to the Supplier Evaluation data is restricted by the Permissions security model, based on the user's roles and what organization they are associated to; the configurable access controls are granular, down the level of individual fields.

There is no need to specifically mark fields as *sensitive* in Supplier Evaluation, however, the retailer/portal owner could chose to do so by configuring the field label system text themselves.

Data Deletion

The deletion of data is limited due to the due diligence nature of Supplier Evaluation. Certain data must be retained indefinitely, so options to delete or purge data are limited to a select few areas. Where data is deleted, it is not possible for it to be re-accessed.

The Right to be Forgotten solution includes the retailer/portal owner's ability to purge personal data from inactive user accounts. As certain data cannot be physically deleted, the erasure is by obfuscating the appropriate fields.

Notice and Consent

Notice and Consent is necessary where personal data is being held or processed. In Supplier Evaluation, the Terms and Conditions pages can be configured by the retailer/portal owner to include the necessary notices for the user to accept or otherwise.

Availability

Backup and disaster recovery facilities for Supplier Evaluation are provided as standard as part of the Cloud Service solution.

The majority of Supplier Evaluation data is available for export using the suite of secure APIs.

Tracking Technologies

Supplier Evaluation does not include any Tracking Technologies such as IP addresses or device fingerprints. A single cookie is used, for session connectivity.

Separation of Duties

Some tasks, such as the approval or completion of an Audit, require actions by separate users; the completion of the tasks is controlled in Supplier Evaluation by workflow rules.

Logging

Maintenance of Supplier Evaluation records is recorded in the Change History log. Read access is not logged.

The Web Service log records details of all API calls; these are accessible only to administrators.

The Supplier Evaluation data is secured by HTTPS/SSL in transit; data at rest in the MySQL application database is not encrypted.

Appendix: Using the Report Attachment API

The Supplier Evaluation Reports module provides a suite of *standard* reports with the facility for building new reports using the Oracle BI Publisher Template Builder plug-in for Word. Reports are scheduled in the Report Scheduler area where the recipients are assigned, along with whether the report is a single or recurring run, and the type of output (PDF, Excel, RTF, or HTML). When the report has been generated, it is stored as a file attachment in the Report Outputs area, within a hierarchical folder structure.

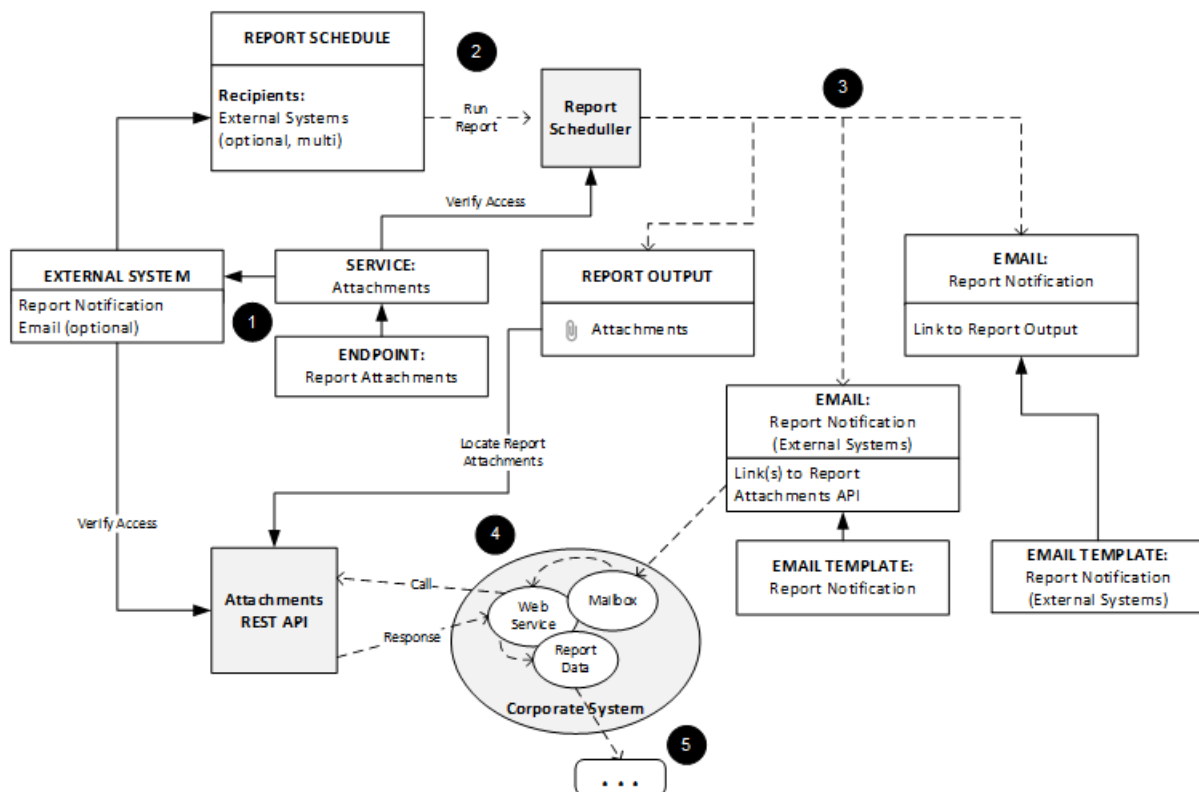
The Report Attachment API provides a means of automatically retrieving the report output attachments from the Report Outputs area, for taking the data into an external system for further analysis or processing.

When a report is generated, the recipients are notified by an email which contains a link to the report. An external system can be defined as a recipient. The External System record in the Supplier Evaluation Admin area can have an email address assigned, making it available for selection as the recipient of a report. The email sent to the external system contains a link to the report along with a link to the actual attachment file; the Report Attachment API can then be used to retrieve the attachment using this link.

The data returned by the API is a Base64 encoded text string which must then be processed by a Base64 decoder to convert it to the actual report file.

The following diagram describes the stages of the process for using the Report Attachment API to retrieve a report output.

Figure 8-1 Process Flow for Using the Report Attachment API



1. To access the Report Attachment API, the external system must be registered in Supplier Evaluation as an External System with the ATTACHMENT service, the REPORT_ATTACHMENT_GET endpoint, and an email address assigned.
2. When a report is scheduled, the external system is selected as a recipient.
3. When the report runs, if the recipient is an external system, an email is sent containing a link to the report output attachment (may be more than one if multiple formats have been selected).

In order to prevent the URI to the Attachment record being predictable, a 32-character non-sequential globalId key is used.

The template of the REPORTNOTIFICATION email may be configured in the Supplier Evaluation Admin area; the default content is:

Subject: Report [Report Name] is ready to view

Body: The report [Report Name] has been created.

Use the following link(s) to retrieve the attachment data from the attachment API.

File name: [File Name].[File Type] [Link to Report Attachments API] File name: [File Name] File Type: [File Type] [Link to Report Attachments API]

4. The external system initiates the call to the Report Attachment API. This may be triggered automatically on receipt of the email, or could be triggered by another event, by a scheduled job, or by a manual user action.

5. The external system processes the returned messages and report attachment data according to its intended purpose. The data must first be decoded from Base64 encoding.

Response Example:

```
<ns1:attachmentFullDTO
  xmlns:ns1="http://www.micros.com/creations/core/domain/dto/v1p0/full"
  xmlns:ns2="http://www.micros.com/creations/core/domain/dto/v1p0/simple">
  <ns1:attachedBy>
    <ns2:code>oau</ns2:code>
    <ns2:disabled>>false</ns2:disabled>
    <ns2:externalAuthenticationUser>>false</ns2:externalAuthenticationUser>
    <ns2:timeZone>Europe/London</ns2:timeZone>
    <ns2:userType>RETAILER</ns2:userType>
    <ns2:id>22</ns2:id>
  </ns1:attachedBy>
  <ns1:attachedOn>2018-06-28T10:20:27.000+01:00</ns1:attachedOn>
  <ns1:dataLength>3786</ns1:dataLength>
  <ns1:fileName>RT1.pdf</ns1:fileName>
  <ns1:localeData>
    <ns1:description><![CDATA[Test]]></ns1:description>
    <ns1:id>1190</ns1:id>
  </ns1:localeData>
  <ns1:id>1190</ns1:id>
  <ns1:ownerClass>com.micros.creations.core.domain.ReportResult</ns1:ownerClass>
  <ns1:ownerId>156</ns1:ownerId>
  <ns1:data>JVBERi0xLjYNCjUgMCBvYmoNCjw8DQovVHlwZSAvUGFnZQ0KL1Bhc...</ns1:data>
</ns1:attachmentFullDTO>
```



Note:

It is the responsibility of the retailer/portal owner to build an application to call the Report Attachment API and handle the returned messages and data, including any further processing of the data or interfaces with external systems.

For portals that use the Enhanced Access Control (EAC) level of permissions, the retrieval of reports as attachments through the API does not apply EAC area permissions filtering (as is the case for all other Supplier Evaluation APIs).

Appendix: Accessibility

Accessibility involves making your application usable for persons with disabilities such as low vision or blindness, deafness, or other physical limitations. This means, for example, creating applications that can be:

- Used without a mouse (keyboard only).
- Used with assistive technologies such as screen readers and screen magnifiers.
- Used without reliance on sound, color, animation, or timing.

User Interface

This user interface is designed using Oracle JavaScript Extension Toolkit (JET). Oracle JET components have built-in accessibility support that conforms to the Web Content Accessibility Guidelines version 2.0 at the AA level (WCAG 2.0 AA), developed by the World Wide Web Consortium (W3C). Please note that since different browsers themselves support accessibility somewhat differently, the user experience tends to differ on different web browsers.

Oracle JET components provide support for:

- Keyboard and touch navigation
Oracle JET components follow the Web Accessibility Initiative - Accessible Rich Internet Application (WAI-ARIA) guidelines.
- Zoom
Oracle JET supports browser zooming up to 200%.
- Screen reader
Oracle JET supports screen readers such as JAWS, Apple VoiceOver, and Google Talkbalk by generating content that complies with WAI-ARIA standards, and no special mode is needed.
- Oracle JET component roles and names
Each Oracle JET component has an appropriate role, such as button, link, and so on, and each component supports an associated name (label), if applicable.
- Sufficient color contrast
Oracle JET provides the Alta theme which is designed to provide a luminosity contrast ratio of at least 4.5:1.

Besides the use of JET components, the following updates were made to the application:

- Added alternative texts to images/logos/icons as needed.
- Added labels as needed to GUI widgets.
- Color Contrast was adjusted.
- Improved error identification and display, so screens readers can read meaningful information.

Appendix: LCA Third-Party Integration

Supplier Evaluation can be configured to allow for the integration of Assessments with third parties. This is of use where the retailer or supplier is using a third party such as a Life Cycle Assessment (LCA) partner, to source specialist data, for example sustainability, environmental and ethical analysis information about a supplier or its products.

For a third party to integrate with the Supplier Evaluation system, the retailer portal owner must register them as an Accredited Body and create an External System account, granting them access to the Scorecard API (they may also be granted access to other APIs for the retrieval of supporting Supplier, Site or Product data).

Assessments can be configured to be the type which will be used to capture data from a third party, in which case one or more Accredited Bodies are assigned. When an Assessment of that type is created, one of the permitted Accredited Bodies is selected. That third party is then able to retrieve the Assessment using the Scorecard REST API to supply their data by populating the answers to its questions. They then submit the Assessment's response answers using the same API.

Once an Assessment is returned it is reviewed, approved and completed. The data provided by the third party then being available for viewing within the Assessment and for use in reports. Summary Attributes may be configured to label responses in a way that they can be shown in the information panels of the associated, Assessment, Site or Product.

The integration may be *supplier owned*, where the supplier liaises with the third party and has responsibility for the final completion of the Assessment, or *retailer owned*, where the retailer typically works with the third party without involving the supplier.

When registering a third party as an Accredited Body it is possible to configure a link to the third party's proprietary system, in which case an option will appear in the top level Accredited Bodies navigation Tasks menu.

Accredited Bodies

The retailer controls which third parties may collaborate by creating an Accredited Body record to register them as a partner, and an External System record to grant them access to the Scorecard API (endpoints for read and update access). The combination of Accredited Body and External System forms the security controls for which third parties may access the API, and to restrict access to specific Assessments.

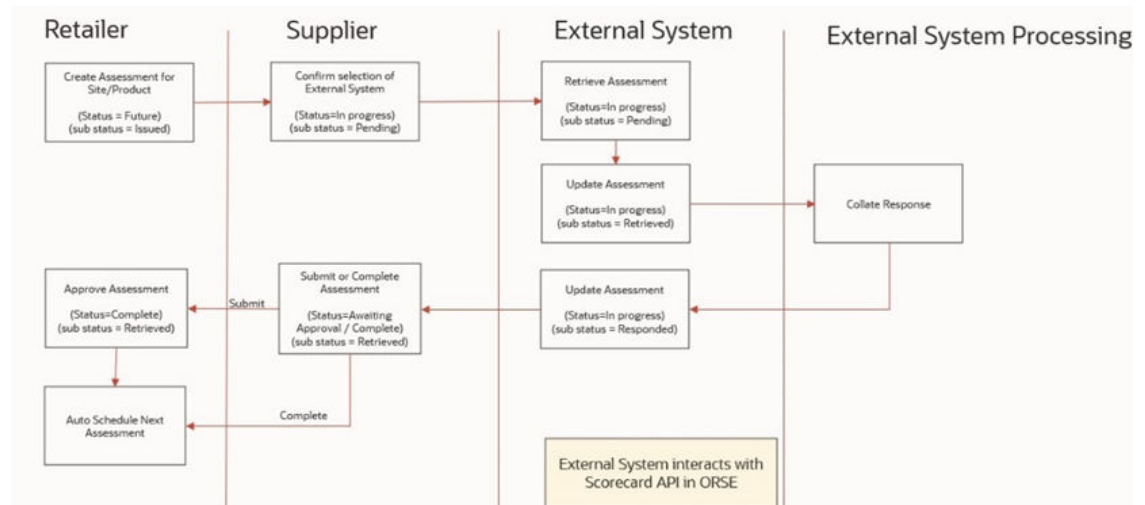
An Integration Type field on the Assessment Template identifies those where data is to be sourced from third parties. The options of Retailer Owned and Supplier Owned define whether the Assessments are being managed by the retailer or the supplier. Once an Accredited Body is registered, it can be selected within the template as being one of the third parties who may be assigned to Assessments of that type (multiple Accredited Bodies may be assigned to a template).

When an Assessment is created from a template with an Accredited Body assigned, the user selects which single third party will be used to source the data for that Assessment. When the Assign Accredited Body action is used, the status of the Assessment becomes In Progress, and its sub status is set to Pending - meaning it is available for retrieval by the designated third party.

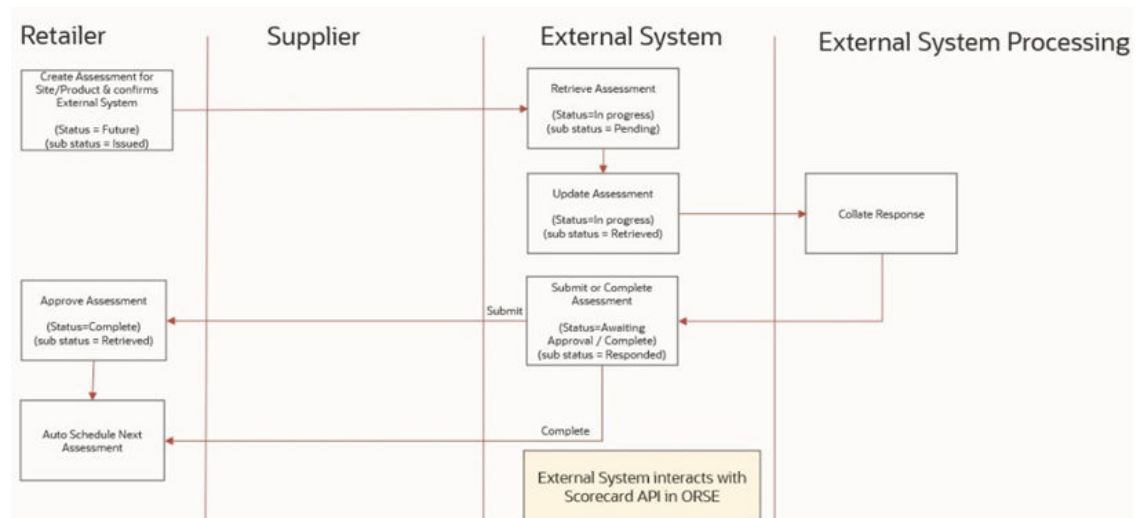
Assessments Integration

The process for third parties to retrieve, respond to, and return Assessments they have been assigned is shown in the following diagrams. The key difference between the two models is: for Supplier Owned, the supplier is notified of the new Assessment and returns the populated Assessment to the retailer, whereas for Retailer Owned, the Assessment is created by the retailer and is assigned directly to the third party, who returns the populated Assessment to the retailer.

Supplier Owned Integration Type

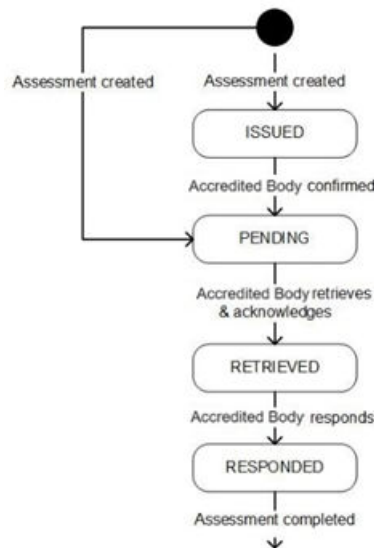


Retailer Owned Integration Type



Sub Status

A Sub Status field on the Assessment is used to control the workflow of those being progressed with a third party:



Third parties retrieve Assessments that have been assigned to them and submit an update to set the sub status to indicate they have been retrieved. Once the questions have been populated with responses that represent the data being provided, a further update is submitted, setting the sub status to indicate they have responded.

Assessment templates have options to set whether the third party and/or the supplier can set an Assessment as being complete, or whether they are submitted for the retailer's approval.

Using the Scorecard API

The Scorecard API is a RESTful API with XML based interaction and payload, supporting GET and PUT methods for the retrieval and update of Assessments.

Access to the API and its endpoints is controlled by the retailer portal owner, as is the access to the individual Assessments. The retailer portal owner may also grant access to other APIs, such as the Supplier, Site or Product APIs for retrieval of additional supplier or item information. Authentication can be configured utilizing both OAuth 2.0 and Basic Authentication, this must also be configured by the retailer portal owner. The recommendation is to utilize OAuth wherever possible.

The third party's access to Assessments is restricted to those to which they are the designated Accredited Body, and which are at the appropriate status for retrieval.

Assessments are retrieved with a GET method, and then the responses to the Assessment's questions can then be returned with a PUT method to update the Assessment. The third party should create a process to periodically call the API to determine if new assessments are available for retrieval. The system does not send notifications when Assessments are assigned to a third party.

When calling the Assessment API, the `modifiedSince` and `modifiedUntil` parameters can be used to focus on any assessments that have been created or amended since the previous call. Filtering on the status of the Assessment can be used to exclude any that have previously been completed since the previous call. To facilitate this, the external system will need to record the retailer specific ORSE portal's URL, authentication method, last called time (for the polling service) and integration frequency/time.

The response to the Get List of Values call will provide a list of links to the Assessments available for retrieval. The individual Assessments are then retrieved by a Retrieve Record by ID call. Once retrieved a call should be made to update the Sub Status of the Assessment,

setting it to RETRIEVED to indicate it has been retrieved by the third party – either with the Update Sub Status endpoint or with a full Update message.

The third party can then analyze the questions in the Assessment in the external system and collate the answers before returning their responses with a full Update response. This is achieved by returning the retrieved payload with the responses populated and the Sub Status set to RESPONDED to indicate it is ready for the retailer or supplier to review.

Assessments may be configured to allow the third party to set the Status to COMPLETE, or to set to AWAITING APPROVAL, in which case the supplier or retailer will review for approval before completing it.

The remainder of this section provides more detail on the integration process and the two use cases.

Assessments Integration Process

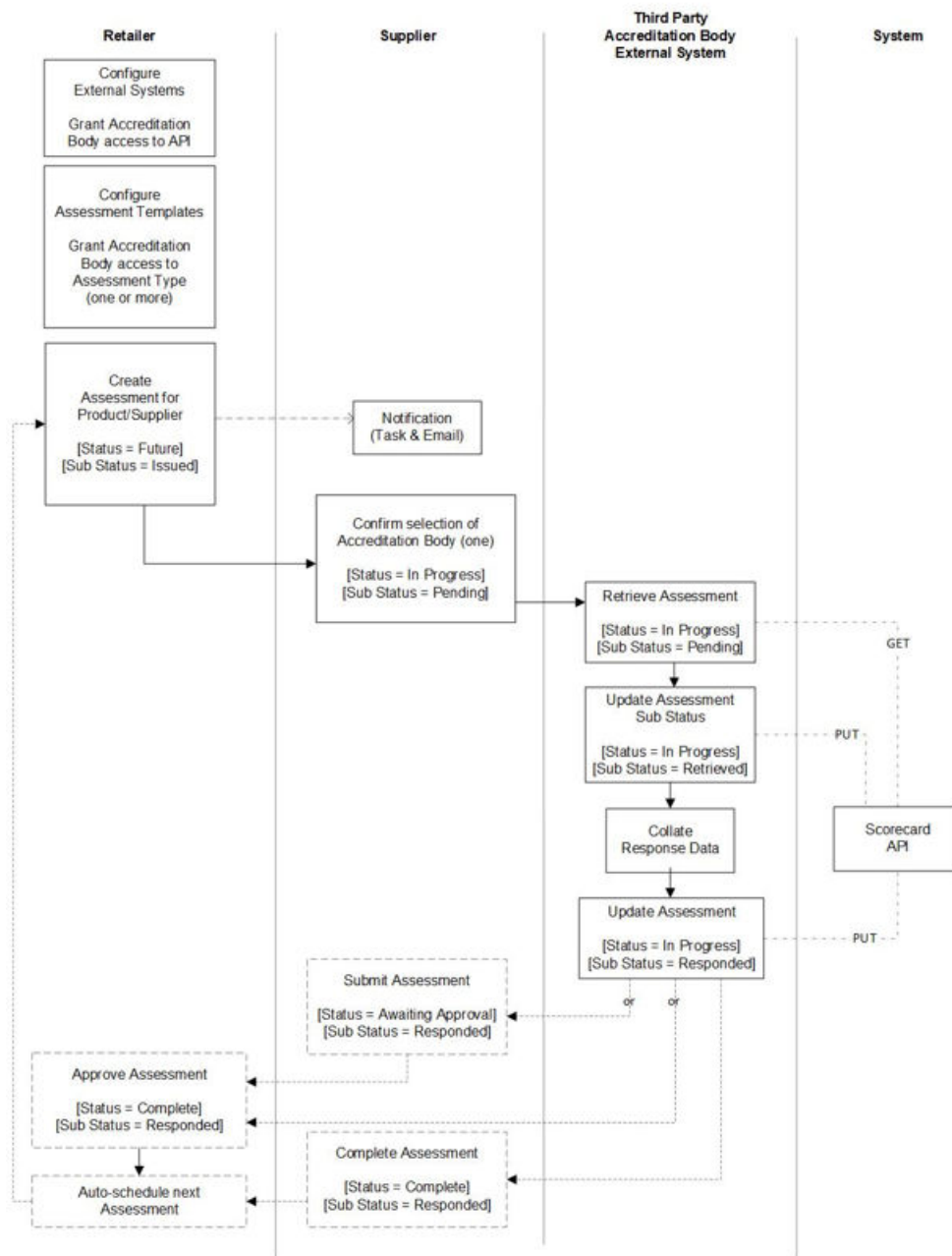
Consider the following when using the Scorecard API to provide third party data as part of the Assessments integration process.

- Assessments may be Item Assessments for an item/product or Site Assessments for a supplier's site location. The type is indicated by the **Assessment Level** field. The same process and endpoints are used for both types.
- As the same Product Number (SKU) may be used by different suppliers (and may change), it is recommended that the **Product Code** be used to identify a product's Assessments, as it is the unique key of the Product Record.
- Assessment questions have a **Question Code**, which can be used to map the responses to the questions. Where questions have multiple answer options (checkboxes, radio buttons and picklists), the individual permissible answers each have an **Answer Code** to be used when populating the answer.
- The **Sub Status** is initially PENDING, with Status being IN PROGRESS. This is the state when it should be retrieved. Once retrieved the Sub Status should be updated to RETRIEVED. When responses are submitted the Sub Status should be set to RESPONDED, and the Status set to COMPLETE or AWAITING APPROVAL.
 - The Assessment's template will control whether the third party may progress the Assessment to the Completed status - populating the Completed On (date), Completed By (name of the External System), or whether it must be set to Awaiting Approval for the retailer or supplier to finalize the workflow.
- The third party will liaise offline with the retailer and/or supplier if further action is required.
- The API applies minimal **validation**; full validation and business logic (calculations, workflow triggers, status changes) when an Assessment record is subsequently edited and progressed within the system. Validation is included in the API to:
 - Prevent the Accredited Body and Integration Type, *Accredited Body Can Complete Assessment* and *Retailer Approval Required* values being changed.
 - Control whether the Status can be set to Complete or Awaiting Approval based on the rules for the assessment type:
 - * If Accredited Body Can Complete Assessment is Yes, Status can only be set to Complete if Retailer Approval Required is No.
 - * If Accredited Body Can Complete Assessment is Yes, Status can only be set to Awaiting Approval if Retailer Approval Required is Yes.

- It is important that when resubmitting an Assessment for update that the **full payload remains populated** as it was when retrieved - with just the response, status, and any comments data being updated - otherwise the data will be cleared.
- For the scenario where a third party is to **periodically update or refresh** data as an on-going task, the assessments can be configured by the retailer to **automatically schedule the next Assessment** on completion.

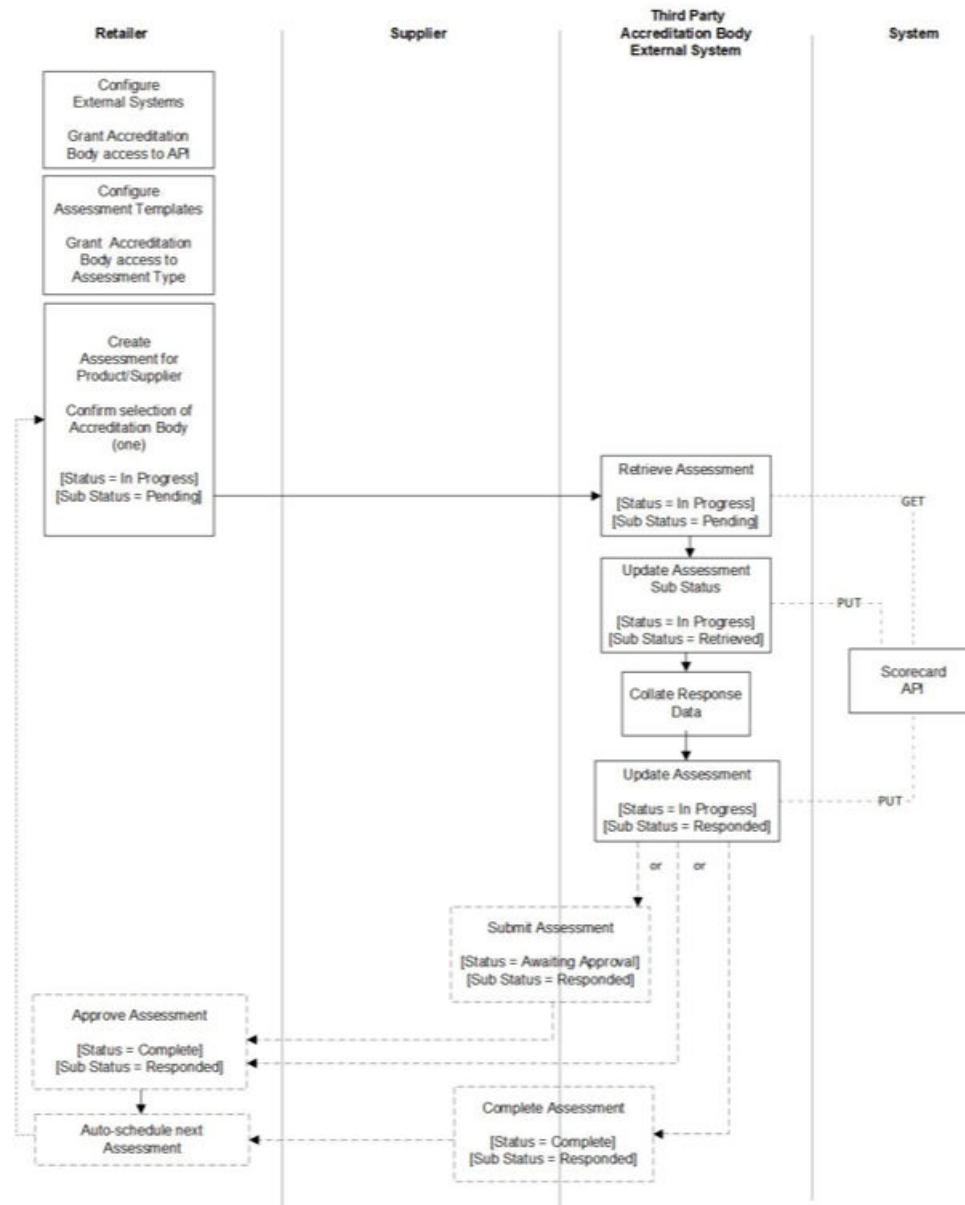
Supplier Owned Integration Type

For this use case, the retailer requests individual suppliers to progress an Assessment which contains questions for which responses requires third party data. The retailer typically creates the Assessment for the supplier to take responsibility for its completion; alternatively, the supplier may create the Assessment themselves. An Assessment may contain a combination of questions requiring responses to be sourced from the third party and also responses from the supplier. The supplier then submits the Assessment to the retailer if approval is required, otherwise completes it themselves.



Retailer Owned Integration Type

For this use case, the retailer requests a third party to provide data for a product or range of products. The retailer creates the Assessment, which is associated to a supplier (so the supplier will also have access to the Assessment). The third party submits the Assessment to the retailer if approval is required, else completes it themselves.



Enabling Sub Status Endpoints

To enable the endpoints for updating an Assessment's sub status, add the following Endpoint Access records to the Scorecard Service Access record in the Roles & Permissions section of the Admin area:

Code: SCORECARD

Description: Scorecard RESTful Service

Endpoint Access: SCORECARD_SUB_STATUS and
SCORECARD_SUB_STATUS_BY_CODE

Select Endpoint Access

Search scorecard_sub

Available

☐ Description
☒ SCORECARD_SUB_STATUS - Updates an In Progress Scorecards sub status and returns a link to the record's details
☒ SCORECARD_SUB_STATUS_BY_CODE - Locates a scorecard by its code to update it's sub status, returns a link to the record's details

Selected

☒ Description
☒ SCORECARD_CHECK - Retrieves the header of information of a single assessment
☒ SCORECARD_EXISTS - Retrieves the detail of a single assessment by its business key
☒ SCORECARD_GET - Retrieves the details of a single assessments
☒ SCORECARD_GETVALUES - Retrieves a single page list of assessments
☒ SCORECARD_SUB_STATUS - Updates an In Progress Scorecards sub status and returns a link to the record's details
☒ SCORECARD_SUB_STATUS_BY_CODE - Locates a scorecard by its code to update it's sub status, returns a link to the record's details

Page 1 of 1

Ok Cancel Clear

Part II

Supply Chain Hub

This part of the *Oracle® Retail Supply Chain Collaboration Cloud Service Implementation Guide* contains information about Supply Chain Hub.

- [Overview](#)
- [Supply Chain Hub](#)
- [Procure to Pay](#)
- [Return to Vendor](#)
- [Inventory and Sales](#)
- [Items Management](#)
- [Cost Changes Management](#)
- [Supplier Performance](#)
- [Quality Control](#)
- [Deals Management](#)
- [Supplier Penalties](#)
- [Retail Price](#)
- [Financial Summary](#)
- [Appointments Management](#)
- [Customer Order Fulfillment](#)
- [Reports](#)
- [Supply Chain Hub Frameworks](#)

Overview

Oracle Retail Supply Chain Hub works as the entry point to a group of applications that combines retailers and suppliers in the same portal. Depending on which applications are available in the Hub the users will be presented with a summary in the homepage and according to the user type (supplier or retailer) will have visibility on Operational News and Queries.

Operational News is a functionality that allows the retailers to share news with targeted suppliers in a central place, the retailers have permissions to create and edit them while the suppliers can only view them.

Another functionality is Queries, this functionality aims to allow the dialog between retailers and suppliers regarding a specific entity, like a Purchase Order or Shipments. This way the dialog is focused on that specific entity and makes easier the context.

Working as Hub, Oracle Retail Supply Chain Hub adapts to the applications that are installed in the system not only with different charts being presented in the homepage but also with a menu with access to all those applications so switching between applications keeps the context of the user in a seamless experience:

Procure to Pay

- Purchase Orders:
 - Purchase Order Download/Upload
 - Purchase Order Acknowledgment
 - Purchase Order Changes/Approval
- Shipments:
 - ASN Download/Upload
 - ASN Creation/Update
 - ASN Cancellation
- Finance Documents:
 - Merchandise Invoice Creation
 - ASN Creation/Update
 - Finance Document Status View

Items Management

- Item Creation
- Item Maintenance
- Item Download/Upload via Spreadsheet

Cost Changes Management

- Cost Changes Creation
- Cost Changes Amendment/Cancellation

- Cost Changes Download/Upload via Spreadsheet

Returns to Vendor

- RTV Creation Workflow
- RTV Download/Upload via Spreadsheet
- RTV Cancellation

Inventory and Sales

- View Inventory
- View Sales

Supplier Performance

- Monitor Supplier Performance major KPIs
- Maintain and Record Supplier Questionnaires

Operational News

- Create News
- View News
- Maintain News

Quality Control

- Create jobs manually
- View and Execute Jobs
- Download existing Jobs
- Upload new Jobs or execute existing ones
- Define, Update or View rules to automate Job creation
- Define, Update or View templates to be used in the Job creation process

Deals Management

- Deals Creation
- Deals Maintenance
- Deals Download/Upload via Spreadsheet

Supplier Penalties

- Penalties Creation
- Penalties Maintenance

Retail Price

- View and Download Price Events
- Create new Price Event via UI
- Approve, Reject and Cancel Price Events
- Propose Changes to Price Events
- Upload new Price Events or update existing non extracted Price Events

Financial Summary

- View Financial Transactions and Status

Appointments Management

- Book and Maintain Appointments via UI
- Book and Maintain Appointments via Spreadsheet
- Appointments Slots Management via Download/Upload Spreadsheet
- Appointments Slot Exceptions Maintenance

Customer Order Fulfillment

- View Customer Orders
- Commit/Dispatch Customer Orders
- Customer Orders Download/Upload via Spreadsheet
- Customer Order Subscription
- Cancel Customer Orders
- Customer Orders Invoicing

Reports

- Share External Reports with Suppliers

Getting Started

Learn how to get started with Oracle Retail Supply Chain Collaboration Cloud Service.

Oracle Retail Data Store (RDS)

The Oracle Retail Data Store (RDS) is a set of infrastructure and tools that allows you to build extensions on top of the latest SaaS version for Retail application data without affecting the original Retail applications. These extensions may consist of database objects, web services, and user interfaces. The core of RDS is a data replication implementation that uses Oracle GoldenGate to replicate application data from Retail applications to a centralized database. The data is kept in sync with the source application database in near-real-time. This data is made available through Oracle REST Data Services (ORDS) and Application Express (APEX) workspaces.

The Oracle Retail Supply Chain Hub replicates its Data Model to RDS enabling the possibility of extending it.

Modules

The Oracle Retail Supply Chain Hub follows a modular approach where an application is composed of different modules that can evolve independently and can be reused in the future by other applications.

The following describes the modules and corresponding hierarchies between them that compose the Oracle Retail Supply Chain Hub. The [Supply Chain Hub Frameworks](#) section describes all modules and submodules.

Table 11-1 Modules

MODULE_ID	MODULE_DESC	MODULE_ALIAS
APPOINTMENTS_MANAGEMENT	Appointments Management Module	SPAP
COST_CHANGES_MGMT	Cost Changes Management	SPCC
CUSTOMER_ORDER_FULFILLMENT	Customer Order Fulfillment	SPCO
DEALS_MANAGEMENT	Deals Management Module	SPDM
FINANCIAL_SUMMARY	Financial Summary Module	SPFS
INVENTORY_AND_SALES	Inventory and Sales	SPIS
ITEMS_MANAGEMENT	Items Management	SPIM
PRICE_MANAGEMENT	Retail Price Module	SPPM
PROCURE_TO_PAY	Procure to Pay Module	SPPP
QUALITY_CONTROL	Quality Control Module	QLCT
REPORTS	Reports Module	RPTS
RETURN_TO_VENDOR	Return To Vendor Module	SPRV
SUPPLIER_PENALTIES	Supplier Penalties Module	SPPY
SUPPLIER_PERFORMANCE	Supplier Performance	SPPF
SUPPLY_CHAIN_HUB	Supply Chain Hub Module	SPCH

Table 11-2 Submodules

MODULEID	SUBMODULEID1	SUBMODULE ID2	SUBMODULEID3	SUBMODULEID4
APPOINTMENTS_MANAGEMENT	NA	NA	NA	NA
APPOINTMENTS_MANAGEMENT	CUSTOM_VALIDATION	NA	NA	NA
APPOINTMENTS_MANAGEMENT	DASHBOARD	NA	NA	NA
APPOINTMENTS_MANAGEMENT	EVENTS	NA	NA	NA
APPOINTMENTS_MANAGEMENT	RCVDOC	NA	NA	NA
APPOINTMENTS_MANAGEMENT	UPLDNLD	NA	NA	NA
APPOINTMENTS_MANAGEMENT	UPLDNLD	APPOINTMENTS_DATA	NA	NA
APPOINTMENTS_MANAGEMENT	UPLDNLD	SLOTS_DATA	NA	NA
COST_CHANGES_MGMT	NA	NA	NA	NA
COST_CHANGES_MGMT	DASHBOARD	NA	NA	NA
COST_CHANGES_MGMT	PUBLISH	NA	NA	NA
COST_CHANGES_MGMT	PUBLISH	COST_CHANGE	NA	NA

Table 11-2 (Cont.) Submodules

MODULEID	SUBMODULEID1	SUBMODULE ID2	SUBMODULEID3	SUBMODULEID4
COST_CHANGES_MGMT	RCVDOC	NA	NA	NA
COST_CHANGES_MGMT	SERVICES	ORDS	PUBLISH	NA
COST_CHANGES_MGMT	SERVICES	ORDS	PUBLISH	COST_CHANGE
COST_CHANGES_MGMT	UPLDNLD	NA	NA	NA
COST_CHANGES_MGMT	UPLDNLD	COST_CHANGE_DATA	NA	NA
CUSTOMER_ORDER_FULFILLMENT	NA	NA	NA	NA
CUSTOMER_ORDER_FULFILLMENT	CUSTOM_VALIDATION	NA	NA	NA
CUSTOMER_ORDER_FULFILLMENT	DASHBOARD	NA	NA	NA
CUSTOMER_ORDER_FULFILLMENT	EVENTS	NA	NA	NA
CUSTOMER_ORDER_FULFILLMENT	INTEGRATION	NA	NA	NA
CUSTOMER_ORDER_FULFILLMENT	PUBLISH	NA	NA	NA
CUSTOMER_ORDER_FULFILLMENT	PUBLISH	ASNIN	NA	NA
CUSTOMER_ORDER_FULFILLMENT	PUBLISH	FINANCE_DOCUMENTS	NA	NA
CUSTOMER_ORDER_FULFILLMENT	PUBLISH	FULFILLMENT_ORDER	NA	NA
CUSTOMER_ORDER_FULFILLMENT	RCVDOC	NA	NA	NA
CUSTOMER_ORDER_FULFILLMENT	SUBSCRIPTION	NA	NA	NA
CUSTOMER_ORDER_FULFILLMENT	UPLDNLD	NA	NA	NA
CUSTOMER_ORDER_FULFILLMENT	UPLDNLD	COF	NA	NA
DEALS_MANAGEMENT	NA	NA	NA	NA
DEALS_MANAGEMENT	CUSTOM_VALIDATION	NA	NA	NA
DEALS_MANAGEMENT	DASHBOARD	NA	NA	NA
DEALS_MANAGEMENT	PUBLISH	NA	NA	NA
DEALS_MANAGEMENT	PUBLISH	DEAL	NA	NA
DEALS_MANAGEMENT	RCVDOC	NA	NA	NA

Table 11-2 (Cont.) Submodules

MODULEID	SUBMODULEID1	SUBMODULE ID2	SUBMODULEID3	SUBMODULEID4
DEALS_MANAGEMENT	SERVICES	ORDS	NA	NA
DEALS_MANAGEMENT	UPLDNLD	NA	NA	NA
DEALS_MANAGEMENT	UPLDNLD	DEALS	NA	NA
FINANCIAL_SUMMARY	NA	NA	NA	NA
FINANCIAL_SUMMARY	DASHBOARD	NA	NA	NA
FINANCIAL_SUMMARY	SERVICES	ORDS	FINANCIAL_SUMMARY	NA
FINANCIAL_SUMMARY	SUBSCRIPTION	NA	NA	NA
FINANCIAL_SUMMARY	SUBSCRIPTION	PAYMENT	NA	NA
INVENTORY_AND_SALES	NA	NA	NA	NA
ITEMS_MANAGEMENT	NA	NA	NA	NA
ITEMS_MANAGEMENT	DASHBOARD	NA	NA	NA
ITEMS_MANAGEMENT	EVENTS	NA	NA	NA
ITEMS_MANAGEMENT	RCVDOC	NA	NA	NA
ITEMS_MANAGEMENT	SERVICES	ORDS	NA	NA
ITEMS_MANAGEMENT	UPLDNLD	NA	NA	NA
ITEMS_MANAGEMENT	UPLDNLD	ITEM_DATA	NA	NA
ITEMS_MANAGEMENT	UPLDNLD	SUPP_DFLT	NA	NA
PRICE_MANAGEMENT	NA	NA	NA	NA
PRICE_MANAGEMENT	CUSTOM_VALIDATION	NA	NA	NA
PRICE_MANAGEMENT	EVENTS	NA	NA	NA
PRICE_MANAGEMENT	PUBLISH	NA	NA	NA
PRICE_MANAGEMENT	PUBLISH	PRICE_EVENTS	NA	NA
PRICE_MANAGEMENT	RCVDOC	NA	NA	NA
PRICE_MANAGEMENT	UPLDNLD	NA	NA	NA

Table 11-2 (Cont.) Submodules

MODULEID	SUBMODULEID1	SUBMODULE ID2	SUBMODULEID3	SUBMODULEID4
PRICE_MANAGEMENT	UPLDNLD	PCEVT_DATA	NA	NA
PRICE_MANAGEMENT	UPLDNLD	PCEVT_SIMP_DATA	NA	NA
PROCURE_TO_PAY	NA	NA	NA	NA
PROCURE_TO_PAY	DASHBOARD	NA	NA	NA
PROCURE_TO_PAY	RCVDOC	NA	NA	NA
PROCURE_TO_PAY	UPLDNLD	NA	NA	NA
PROCURE_TO_PAY	UPLDNLD	DOC_DATA	NA	NA
PROCURE_TO_PAY	UPLDNLD	PURCHASE_ORDER_DATA	NA	NA
PROCURE_TO_PAY	UPLDNLD	SHIPMENT_ASNIN_DATA	NA	NA
QUALITY_CONTROL	NA	NA	NA	NA
QUALITY_CONTROL	CUSTOM_VALIDATION	NA	NA	NA
QUALITY_CONTROL	DASHBOARD	NA	NA	NA
QUALITY_CONTROL	EVENTS	NA	NA	NA
QUALITY_CONTROL	PUBLISH	NA	NA	NA
QUALITY_CONTROL	PUBLISH	RTV	NA	NA
QUALITY_CONTROL	RCVDOC	NA	NA	NA
QUALITY_CONTROL	RULE_ENGINE	NA	NA	NA
QUALITY_CONTROL	UPLDNLD	NA	NA	NA
QUALITY_CONTROL	UPLDNLD	JOB	NA	NA
REPORTS	NA	NA	NA	NA
RETURN_TO_VENDOR	NA	NA	NA	NA
RETURN_TO_VENDOR	CUSTOM_VALIDATION	NA	NA	NA
RETURN_TO_VENDOR	DASHBOARD	NA	NA	NA
RETURN_TO_VENDOR	PUBLISH	NA	NA	NA
RETURN_TO_VENDOR	PUBLISH	RTV	NA	NA
RETURN_TO_VENDOR	RCVDOC	NA	NA	NA
RETURN_TO_VENDOR	UPLDNLD	NA	NA	NA
RETURN_TO_VENDOR	UPLDNLD	RTV	NA	NA
SUPPLIER_PENALTIES	NA	NA	NA	NA
SUPPLIER_PENALTIES	CUSTOM_VALIDATION	NA	NA	NA

Table 11-2 (Cont.) Submodules

MODULEID	SUBMODULEID1	SUBMODULE ID2	SUBMODULEID3	SUBMODULEID4
SUPPLIER_PENALTIES	DASHBOARD	NA	NA	NA
SUPPLIER_PENALTIES	EVENTS	NA	NA	NA
SUPPLIER_PENALTIES	PUBLISH	NA	NA	NA
SUPPLIER_PENALTIES	PUBLISH	FINANCE_DOCUMENTS	NA	NA
SUPPLIER_PENALTIES	RCVDOC	NA	NA	NA
SUPPLIER_PENALTIES	RULE_ENGINE	NA	NA	NA
SUPPLIER_PERFORMANCE	NA	NA	NA	NA
SUPPLIER_PERFORMANCE	SERVICES	ORDS	NA	NA
SUPPLIER_PERFORMANCE	UPLDNLD	NA	NA	NA
SUPPLIER_PERFORMANCE	UPLDNLD	QUESTN	NA	NA
SUPPLIER_PERFORMANCE	UPLDNLD	QUESTN_REPLY	NA	NA
SUPPLY_CHAIN_HUB	NA	NA	NA	NA
SUPPLY_CHAIN_HUB	OPERATIONAL_NETWORKS	NA	NA	NA

12

Supply Chain Hub

This section describes the common functionality and configurations available for Supply Chain Hub.

Functionalities

Supply Chain Hub works as the entry point to a group of applications that combines retailers and suppliers in the same portal. Depending on which applications are available in the Hub the users will be presented with a summary in the homepage and according to the user type (supplier or retailer) will have visibility on Operational News and Queries.

Operational News is a functionality that allows the retailers to share news with targeted suppliers in a central place, the retailers have permissions to create and edit them while the suppliers can only view them.

Another functionality is Queries, this functionality aims to allow the dialog between retailers and suppliers regarding a specific entity, like a Purchase Order or Shipment. This way the dialog is focused on that specific entity and makes easier the context.

Working as true Hub, Supply Chain Hub adapts to the applications that are installed in the system not only with different charts being presented in the homepage but also with a menu with access to all those applications so switching between applications keeps the context of the user in a seamless experience.

13

Procure to Pay

This section describes the common functionality and configurations available for Procure to Pay.

Procure to Pay features include the possibility of creating and maintaining Purchase Orders, Shipments and create Finance Documents. It also includes the functionality of downloading and uploading majority of those entries being a more effective way to maintain in bulk. Each of the entities maintained in Procure to Pay have specific sections to details of the implementation details.

Purchase Orders

This section describes the Purchase Orders features.

Implementation Overview

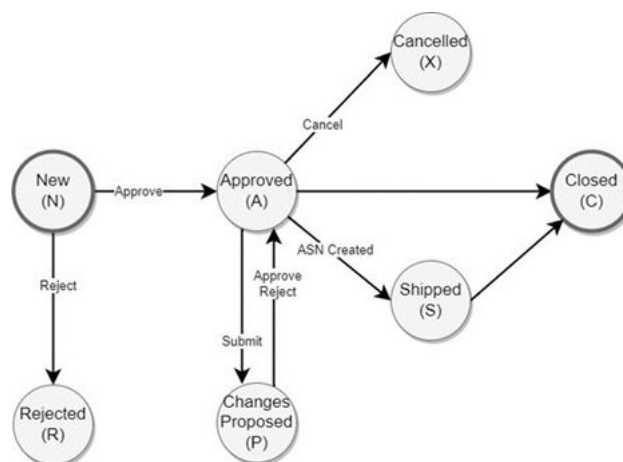
This section provides an implementation overview of the Purchase Orders functionality.

Functionality

Procure to Pay, Purchase Orders, will allow retailers and suppliers to create and maintain purchase orders.

Status Workflow

Figure 13-1 Procure to Pay Purchase Orders Status Workflow



Purchase Order follows a workflow path, and the status can change depending on:

- The user permissions

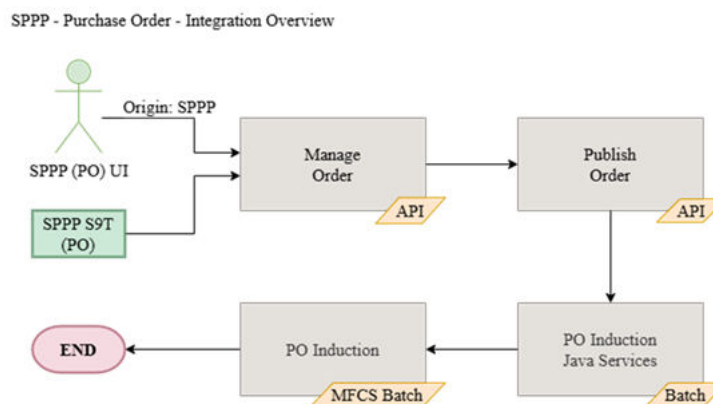
- The current status of the purchase order

Purchase Order Status:

- New – New is the initial status of a Purchase Order. In this status a Purchase Order can be approved to send the Purchase Order to MFCS or Rejected to change the status of the Purchase Order to rejected.
- Rejected – Reject status is for Purchase Orders that were not previously approved.
- Approved – This status is used after the Purchase Order is Approved, in this state the Order has been sent to MFCS.
- Changes Proposed – Changes proposed is the status used when a user without approval permissions updates an approved Purchase Order. During this status the changes made on the purchase order can be approved and then sent to MFCS or rejected and the changes are discarded.
- Cancelled – Cancelled status is used when the Purchase Order is cancelled in Procure to Pay.
- Shipped – Shipped status is used whenever a shipment has been created for the Purchase Order.
- Closed – Closed is the final status of the order once MFCS closes the order. This status can also be used for MFCS side cancelled orders, i.e., not done via SCH.

Integration High Level

Figure 13-2 SPPP Purchase Order Integration Overview



The integration of Purchase Orders from Procure to Pay to MFCS described above is done in the following order:

- Purchase Orders are created and managed in Procure to Pay, either through the UI or via upload functionality.
- After being approved or cancelled, Purchase Orders are published by a batch process.
- Purchase Orders Induction Java Services adds Purchase Orders to an MFCS induction file, which is transferred to MFCS SFTP/FTS.
- Finally, MFCS induction batch for Purchase Orders is executed and integrates Purchase Orders to MFCS.

Configuration

This section describes the Procure to Pay configuration.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Schedule Overview

Figure 13-3 Procure to Pay Purchase Orders Batch Schedule Overview



Other Cloud Services Configuration

This section describes implementation details that must be reviewed from an Oracle Retail Merchandising System perspective.

Batches

The MFCS integration order induction job, POINDUCT_JOB, should be configured to execute with high frequency to assure that purchase orders changes made in Procure to Pay are integrated into MFCS as soon as possible with final target the merchandising tables via parameter "RMS".

Other Implementation Considerations

This section describes implementation considerations for Purchase Order features.

Custom Validations - Purchase Orders

The module performs validations by implementing Custom Validation Groups of the Custom Validations Framework described in the [Supply Chain Hub Frameworks](#) section. The groups listed below can be used to add Custom Validations for Purchase Orders.

Group Id	Executed When
SPPP_ORDER_APPROVE	Purchase Order is Approved
SPPP_ORDER_CANCEL	Purchase Order is Cancelled
SPPP_ORDER_REJECT	Purchase Order is Rejected
SPPP_ORDER_SUBMIT	Purchase Order is Submitted for Approval

Materialized Views

This application requires some results to be materialized in tables using the Materialized View Framework described in the [Supply Chain Hub Frameworks](#) section, below is the list for this application. Materialized views used in Procure to Pay, Purchase Orders, are listed in this section.

ORCV_MV_SPPP_DASH_ALL_PO

Materialize View to support Supply Chain Hub dashboard “All POs Available” showing all Purchase Orders available.

- **Pre-Dependencies:** None
- **Post-Dependencies:** None
- **Frequency:** Ad-hoc.

ORCV_MV_SPPP_DASH_PO_NEW

Materialize View to support Supply Chain Hub dashboard “New POs to be Approved” showing Purchase Orders with Status “New”.

- **Pre-Dependencies:** None
- **Post-Dependencies:** None
- **Frequency:** Ad-hoc

ORCV_MV_SPPP_DASH_PO_PENDING

Description: Materialize View to support Supply Chain Hub dashboard “POs Pending Action” showing Purchase Orders with pending actions by Status.

Pre-Dependencies: None

Post-Dependencies: None

Frequency: Ad-hoc.

ORCV_MV_SPPP_DASH_PO_QTY_L5W

Materialize View to support Supply Chain Hub dashboard “PO Quantity Expected” showing Purchase Orders expected Quantities.

- **Pre-Dependencies:** None
- **Post-Dependencies:** None
- **Frequency:** Ad-hoc.

ORCV_MV_SPPP_DASH_PO_WAIT_ASN

Materialize View to support Supply Chain Hub dashboard “POs Waiting Shipment” showing Purchase Orders with Status "Approved" without shipment.

- **Pre-Dependencies:** None
- **Post-Dependencies:** None
- **Frequency:** Ad-hoc.

ORCV_MV_SPPP_DASH_PO_WAIT_DOC

Materialize View to support Supply Chain Hub dashboard “POs Waiting Invoices” showing Purchase Orders with Status "Approved" or "Closed" without invoice.

- **Pre-Dependencies:** None
- **Post-Dependencies:** None
- **Frequency:** Ad-hoc.

ORCV_MV_SPPP_DASH_PO_WAIT_REV

Materialize View to support Supply Chain Hub dashboard “POs Waiting Review” showing Purchase Orders with Status "Changes Proposed" without revision.

- **Pre-Dependencies:** None
- **Post-Dependencies:** None
- **Frequency:** Ad-hoc.

ORCV_MV_SPPP_PO_LAST_REV

Revision number for Purchase Orders.

Pre-Dependencies: None

Post-Dependencies: None

Frequency: Ad-hoc, Nightly

ORCV_MV_SPPP_PO_REJ_CANCEL

Returns the number of cancelled and rejected POs, by supplier and department displayed in Procure to Pay Home page.

Pre-Dependencies: None

Post-Dependencies: None

Frequency: Ad-hoc.

ORCV_MV_SPPP_PO_TOTALS

Returns the total quantity and total cost of the POs, displayed in Procure to Pay Home page grid.

- **Pre-Dependencies:** None
- **Post-Dependencies:** None

- **Frequency:** Ad-hoc.

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

Table 13-1 Codes

CODE TYPE	CODE TYPE DESCRIPTION
ORRC	Purchase Order Reason Codes
ORST	Purchase Order Status
ORTP	Purchase Order Type
P RTP	Purchase Order Type
PYMT	Payment Method

Templates

Procure to Pay, Purchase Orders, uses a spreadsheet download and upload process, developed as part of the [Supply Chain Hub Frameworks](#) framework, for creation, maintenance, and extraction of data. To support these processes there are some templates which must be configured, and several configurations should be reviewed as part of the implementation.

Templates SPPP Purchase Orders Data

Table 13-2 Template Key: SPPP_PURCHASE_ORDER_DATA

ACTION TYPE	ROLE NAME
Download	ORCV_SPPP_ORDER_APPR_JOB
Download	ORCV_SPPP_ORDER_EDIT_JOB
Download	ORCV_SPPP_ORDER_VIEW_JOB
Download Blank	ORCV_SPPP_APP_JOB
Upload	ORCV_SPPP_ORDER_APPR_JOB
Upload	ORCV_SPPP_ORDER_EDIT_JOB

Table 13-3 Orders Worksheet

ACTION	STATUS	MANDATORY CELLS / FIELDS	NOTES
Create	New	Action, Master order number, order number, status, location type, location, supplier, currency code, exchange rate terms	When user has edit permissions Create an order status New.

Table 13-3 (Cont.) Orders Worksheet

ACTION	STATUS	MANDATORY CELLS / FIELDS	NOTES
Create	Approve	Action, Master order number, order number, status, location type, location, supplier, currency code, exchange rate terms	When user has approval permissions Create an approved order
Update	New	Action, Master order number, order number, status, location type, location, supplier, currency code, exchange rate terms	If the upload is done by a user with edit permissions, the order keeps the status New, if it is done by a user with approval permissions, the order status will change to Approved. Update order/items data (Applicable to scenarios where the order only exists in the SPPP application)
Update	Approve	Action, Master order number, order number, status, location type, location, supplier, currency code, exchange rate terms	If the upload is done by a user with approval permissions, the order keeps the approved status, if it is done by a user with edit permissions, the order status will change to Change Proposed. If the order does not yet exist in SPPP, the order will be created based on information that exists in MFCS
Update	Change Proposed	Action, Master order number, order number, status, location type, location, supplier, currency code, exchange rate terms	If the upload is done by a user with edit permissions, the order keeps the change proposed status, if it is done by a user with approval permissions, the order status will change to Approved. If the order does not yet exist in SPPP, the order will be created based on information that exists in MFCS.
Approve	Not Applicable	Action, Master order number, order number, status, location type, location, supplier, currency code, exchange rate terms	When the user has approval permissions. Only for order status evolution. The order will be approved (applicable to existing SPPP orders with status New or Change Proposed) Order items worksheet data will not be used; worksheet can be left blank, or if there is data, leave the action field empty.

Table 13-3 (Cont.) Orders Worksheet

ACTION	STATUS	MANDATORY CELLS / FIELDS	NOTES
Acknowledge	Not Applicable	Action, Master order number, order number, status, location type, location, supplier, currency code, exchange rate terms	When user has edit permissions. This action will only be executed successfully if there is a higher approved order revision in the MFCS than the one in the SPPP. If the order does not yet exist in SPPP, the order will be created based on information that exists in MFCS. Order items worksheet data will not be used; worksheet can be left blank, or if there is data, leave the action field empty
Cancel	Not Applicable	Action, Master order number, order number, status, location type, location, supplier, currency code, exchange rate terms	When the user has approval permissions. Only for order status evolution. The order will be cancelled (applicable to existing orders with status Approve or Change Proposed). If the order does not yet exist in SPPP, the order will be created based on information that exists in MFCS. Order items worksheet data will not be used; worksheet can be left blank, or if there is data, leave the action field empty.
Reject	Not Applicable	Action, Master order number, order number, status, location type, location, supplier, currency code, exchange rate terms	When the user has edit permissions, can only reject orders with status New. When user has approval permissions, can reject order with status New and Change Proposed. If the order has change proposed status, status orders will stay approve and order's data will be reverted to the last approved values. Order items worksheet data will not be used; worksheet can be left blank, or if there is data, leave the action field empty.

Table 13-4 Order Items Worksheet

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Action, Quantity Item or Reference Item or VPN Supplier pack size Country of sourcing	Add items to delivery. Applicable when the action on the orders worksheet is Create or Update

Table 13-4 (Cont.) Order Items Worksheet

ACTION	MANDATORY CELLS / FIELDS	NOTES
Update	Action, Quantity Item or Reference Item or VPN Supplier pack size Country of sourcing	Update delivery items data Applicable when the action on the orders worksheet is Update
Cancel	Action, Item or Reference Item or VPN	Cancel delivery items data (zero out quantity value) Applicable when the action on the orders worksheet is Update

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
PPAC	Purchase Order Actions
PPST	Purchase Orders Status
PPIA	Purchase Order Items Actions

Interactive Grid Additional Columns

This application supports adding additional fields to the Interactive Grids by using the Interactive Grid Additional Columns functionality described in the [Supply Chain Hub Frameworks](#) section.

The Interactive Grids that support this functionality are listed below:

Interactive Grid	Grid ID	User Types	Page	Non Based View	Entity Key
Purchase Orders Dashboard Grid	P1_IG_ORDERS	Retail Administrator, Retailer User, Supplier Administrator, Supplier User	Home	ORCV_V_SPPP_ORD_NB	ORDER_NO

Interactive Grid	Grid ID	User Types	Page	Non Based View	Entity Key
Deliveries Grid	P2_IG_DELIVERY_GR ID	Retail Adminis trator, Retailer User, Supplier Adminis trator, Supplier User	Create or Edit PO	ORCV_V_SPPP_ORD_ NB	ORDER_N O

References

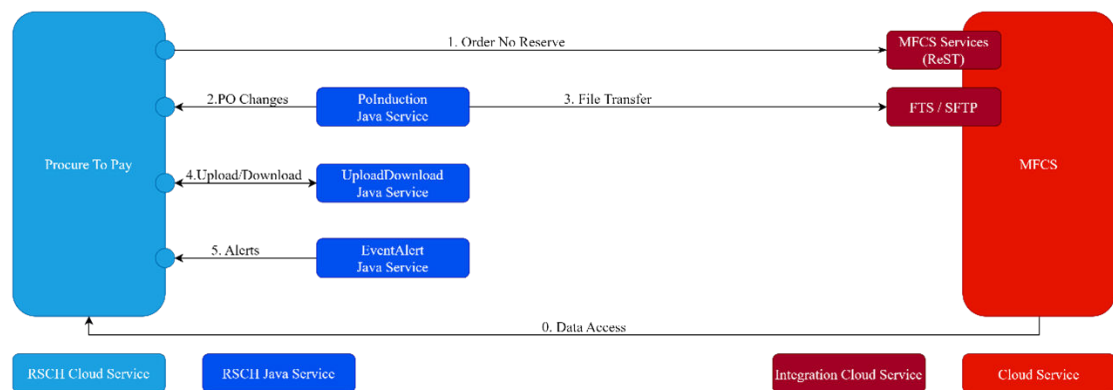
Procure to Pay, Purchase Orders, also uses the Queries module to enable communications between retailer and supplier users, the details about Queries are available in the [Supply Chain Hub Frameworks](#) section. It also has dependencies from other modules batch jobs listed in POM Configuration that are documented in the [Supply Chain Hub Frameworks](#) section.

Operations

This section describes Purchase Order operations.

Integration by Modules

Figure 13-4 Procure to Pay Purchase Orders Integration by Module



STEP	DESCRIPTION
0	RSCH Data Access of MFCS data
1	Order No Reserve calls MFCS Services with a request to reserve Order No.
2	PO Induction Java Services checks if Procure to Pay has purchase orders to be published and produces ODS files to be consumed by MFCS PO Induction Job.
3	PO Induction Java Services sends produced files to Integration File Transfer.
4	Integration File Transfer publishes files to MFCS File Transfer Services (FTS) or MFCS shared folder (depending on configuration).

STEP	DESCRIPTION
5	Procure to Pay calls Upload Download Java Service to transform files, depending on the request details, Upload Download Java Service might need to callback to Procure to Pay to get more details.
6	Event Alert Java Service checks if there are alerts to be triggered.

Reference

For operational details on integration and order reserve services, refer to the [Supply Chain Hub Frameworks](#) section.

Materialized Views Refresh Detail

This section describes Materialized Views Refresh Details.

ORCV_MV_SPPP_DASH_ALL_PO

Parameter	Description	Value
Result Synonym	Synonym name pointing to the latest result table.	ORCV_MV_SPPP_DASH_ALL_PO
Result Table A	Name of _A result table.	ORCV_SPPP_DASH_ALL_PO_A_R
Result Table B	Name of _B result table.	ORCV_SPPP_DASH_ALL_PO_B_R
Source View	View used to refresh the result tables.	ORCV_V_SPPP_DASH_ALL_PO_R
Driving View	View used to distribute the refresh of result tables.	ORCV_V_SPPP_DASH_ALL_PO_RC
Pre-Processor	Pre-process procedure to execute, if applicable.	ORCV_COMMON_MV_REFRESH COMMON_PRE_PROCESSOR
Check Function	Check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if restart is allowed or not.	N
Max Concurrent Threads	The maximum concurrent threads to refresh result tables.	4
Filter Fields	Column(s) to distribute the refresh of result tables.	DEPT, SUPPLIER

ORCV_MV_SPPP_DASH_PO_NEW

Parameter	Description	Value
Result Synonym	Synonym name pointing to the latest result table.	ORCV_MV_SPPP_DASH_PO_NEW

Parameter	Description	Value
Result Table A	Name of _A result table.	ORCV_SPPP_DASH_PO_NEW_A_R
Result Table B	Name of _B result table.	ORCV_SPPP_DASH_PO_NEW_B_R
Source View	View used to refresh the result tables.	ORCV_V_SPPP_DASH_PO_NEW_R
Driving View	View used to distribute the refresh of result tables.	ORCV_V_SPPP_DASH_PO_NEW_RC
Pre-Processor	Pre-process procedure to execute, if applicable.	ORCV_COMMON_MV_REFRESH .COMMON_PRE_PROCESSOR
Check Function	Check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if restart is allowed or not.	N
Max Concurrent Threads	The maximum concurrent threads to refresh result tables.	4
Filter Fields	Column(s) to distribute the refresh of result tables.	DEPT, SUPPLIER

ORCV_MV_SPPP_DASH_PO_PENDING

Parameter	Description	Value
Result Synonym	Synonym name pointing to the latest result table.	ORCV_MV_SPPP_DASH_PO_PENDING
Result Table A	Name of _A result table.	ORCV_SPPP_DASH_PO_PENDING_A_R
Result Table B	Name of _B result table.	ORCV_SPPP_DASH_PO_PENDING_B_R
Source View	View used to refresh the result tables.	ORCV_V_SPPP_DASH_PO_PENDING_R
Driving View	View used to distribute the refresh of result tables.	ORCV_V_SPPP_DASH_PO_PENDING_RC
Pre-Processor	Pre-process procedure to execute, if applicable.	ORCV_COMMON_MV_REFRESH .COMMON_PRE_PROCESSOR
Check Function	Check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if restart is allowed or not.	N

Parameter	Description	Value
Max Concurrent Threads	The maximum concurrent threads to refresh result tables.	4
Filter Fields	Column(s) to distribute the refresh of result tables.	DEPT, SUPPLIER

ORCV_MV_SPPP_DASH_PO_QTY_L5W

Parameter	Description	Value
Result Synonym	Synonym name pointing to the latest result table.	ORCV_MV_SPPP_DASH_PO_QTY_L5W
Result Table A	Name of _A result table.	ORCV_SPPP_DASH_PO_QTY_L5W_A_R
Result Table B	Name of _B result table.	ORCV_SPPP_DASH_PO_QTY_L5W_B_R
Source View	View used to refresh the result tables.	ORCV_V_SPPP_DASH_PO_QTY_L5W_R
Driving View	View used to distribute the refresh of result tables.	ORCV_V_SPPP_DASH_PO_QTY_L5W_RC
Pre-Processor	Pre-process procedure to execute, if applicable.	ORCV_COMMON_MV_REFRESH COMMON_PRE_PROCESSOR
Check Function	Check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if restart is allowed or not.	N
Max Concurrent Threads	The maximum concurrent threads to refresh result tables.	4
Filter Fields	Column(s) to distribute the refresh of result tables.	DEPT, SUPPLIER

ORCV_MV_SPPP_DASH_PO_WAIT_ASN

Parameter	Description	Value
Result Synonym	Synonym name pointing to the latest result table.	ORCV_MV_SPPP_DASH_PO_WAIT_ASN
Result Table A	Name of _A result table.	ORCV_SPPP_DASH_PO_WAIT_ASN_A_R
Result Table B	Name of _B result table.	ORCV_SPPP_DASH_PO_WAIT_ASN_B_R
Source View	View used to refresh the result tables.	ORCV_V_SPPP_DASH_PO_WT_ASN_R
Driving View	View used to distribute the refresh of result tables.	ORCV_V_SPPP_DASH_PO_WT_ASN_RC

Parameter	Description	Value
Pre-Processor	Pre-process procedure to execute, if applicable.	ORCV_COMMON_MV_REFRESH . COMMON_PRE_PROCESSOR
Check Function	Check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if restart is allowed or not.	N
Max Concurrent Threads	The maximum concurrent threads to refresh result tables.	4
Filter Fields	Column(s) to distribute the refresh of result tables.	DEPT, SUPPLIER

ORCV_MV_SPPP_DASH_PO_WAIT_DOC

Parameter	Description	Value
Result Synonym	Synonym name pointing to the latest result table.	ORCV_MV_SPPP_DASH_PO_WAIT_DOC
Result Table A	Name of _A result table.	ORCV_SPPP_DASH_PO_WAIT_DOC_A_R
Result Table B	Name of _B result table.	ORCV_SPPP_DASH_PO_WAIT_DOC_B_R
Source View	View used to refresh the result tables.	ORCV_V_SPPP_DASH_PO_WAIT_DOC_R
Driving View	View used to distribute the refresh of result tables.	ORCV_V_SPPP_DASH_PO_WAIT_DOC_RC
Pre-Processor	Pre-process procedure to execute, if applicable.	ORCV_COMMON_MV_REFRESH . COMMON_PRE_PROCESSOR
Check Function	Check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if restart is allowed or not.	N
Max Concurrent Threads	The maximum concurrent threads to refresh result tables.	4
Filter Fields	Column(s) to distribute the refresh of result tables.	DEPT, SUPPLIER

ORCV_MV_SPPP_DASH_PO_WAIT_REV

Parameter	Description	Value
Result Synonym	Synonym name pointing to the latest result table.	ORCV_MV_SPPP_DASH_PO_WAIT_REV
Result Table A	Name of _A result table.	ORCV_SPPP_DASH_PO_WAIT_REV_A_R
Result Table B	Name of _B result table.	ORCV_SPPP_DASH_PO_WAIT_REV_B_R
Source View	View used to refresh the result tables.	ORCV_V_SPPP_DASH_PO_WT_REV_R
Driving View	View used to distribute the refresh of result tables.	ORCV_V_SPPP_DASH_PO_WT_REV_RC
Pre-Processor	Pre-process procedure to execute, if applicable.	ORCV_COMMON_MV_REFRESH . COMMON_PRE_PROCESSOR
Check Function	Check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if restart is allowed or not.	N
Max Concurrent Threads	The maximum concurrent threads to refresh result tables.	4
Filter Fields	Column(s) to distribute the refresh of result tables.	DEPT, SUPPLIER

ORCV_MV_SPPP_PO_LAST_REV

Parameter	Description	Value
Result Synonym	Synonym name pointing to the latest result table.	ORCV_MV_SPPP_PO_LAST_REV
Result Table A	Name of _A result table.	ORCV_SPPP_PO_LAST_REV_A_R
Result Table B	Name of _B result table.	ORCV_SPPP_PO_LAST_REV_B_R
Source View	View used to refresh the result tables.	ORCV_V_SPPP_PO_LAST_REV_R
Driving View	View used to distribute the refresh of result tables.	ORCV_V_SPPP_PO_LAST_REV_RC
Pre-Processor	Pre-process procedure to execute, if applicable.	ORCV_COMMON_MV_REFRESH . COMMON_PRE_PROCESSOR
Check Function	Check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y

Parameter	Description	Value
Restart Indicator	Flag to indicate if restart is allowed or not.	N
Max Concurrent Threads	The maximum concurrent threads to refresh result tables.	4
Filter Fields	Column(s) to distribute the refresh of result tables.	SUPPLIER

ORCV_MV_SPPP_PO_REJ_CANCEL

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPPP_PO_REJ_CANCEL
Result Table A	Name of _A result table.	ORCV_SPPP_PO_REJ_CANCEL_A_R
Result Table B	Name of _B result table.	ORCV_SPPP_PO_REJ_CANCEL_B_R
Source View	The View contains business logic to be used to populate the result table.	ORCV_V_SPPP_PO_REJ_CANCEL_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_SPPP_PO_REJ_CANCEL_RC
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH.COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the maximum concurrent threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	SUPPLIER

ORCV_MV_SPPP_PO_TOTALS

Parameter	Description	Value
Result Synonym	Synonym name pointing to the latest result table.	ORCV_MV_SPPP_PO_TOTALS
Result Table A	Name of _A result table.	ORCV_SPPP_PO_TOTALS_A_R
Result Table B	Name of _B result table.	ORCV_SPPP_PO_TOTALS_B_R
Source View	View used to refresh the result tables.	ORCV_V_SPPP_PO_TOTALS_R
Driving View	View used to distribute the refresh of result tables.	ORCV_V_SPPP_PO_TOTALS_RC

Parameter	Description	Value
Pre-Processor	Pre-process procedure to execute, if applicable.	ORCV_COMMON_MV_REFRESH .COMMON_PRE_PROCESSOR
Check Function	Check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if restart is allowed or not.	N
Max Concurrent Threads	The maximum concurrent threads to refresh result tables.	4
Filter Fields	Column(s) to distribute the refresh of result tables.	ORDER_NO

Shipments

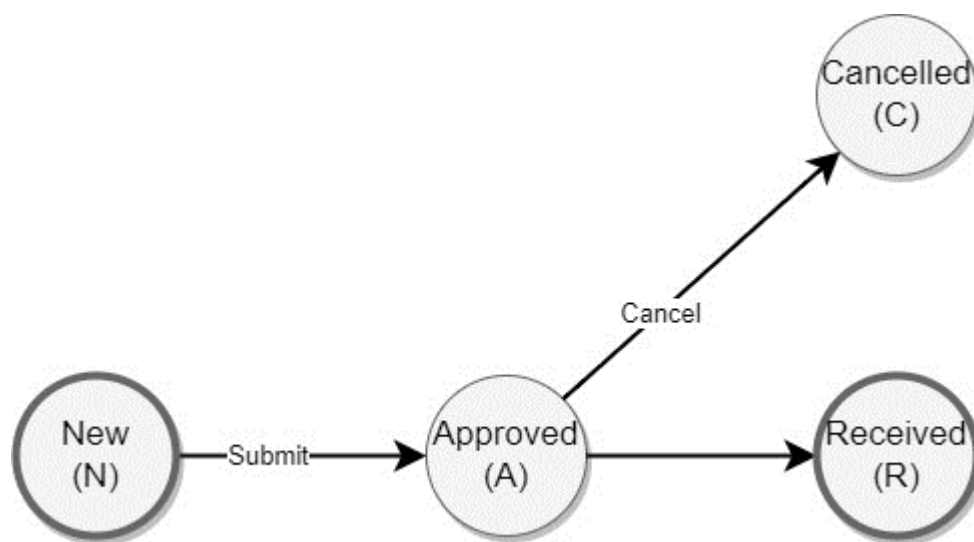
This section describes the Shipments features.

Implementation Overview

Procure to Pay, Shipments, will allow retailers and suppliers to create and maintain purchase ASNs.

Status Workflow

Figure 13-5 Shipments Status Workflow



Shipments follow a workflow path, and the status can change depending on:

- The user permissions
- The status of the Shipment

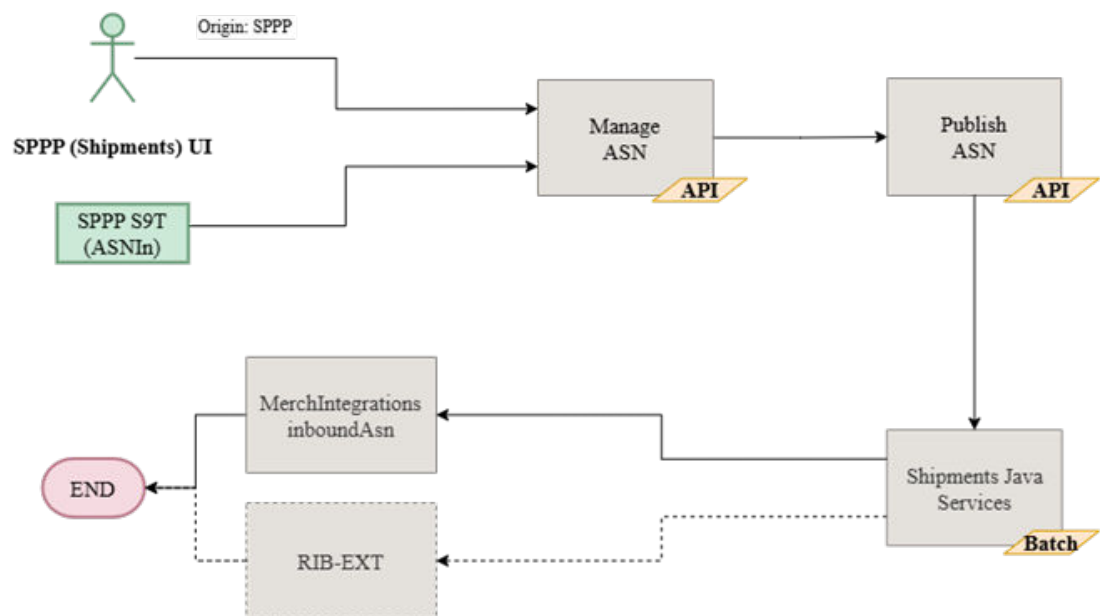
Shipment Status:

- New – New is the initial status while creating the Shipment in SPCH, this status is only an interim status and it is not possible to save a shipment in this status.
- Approved – This status is used once a Shipment is submitted and integrated into MFCS. Whenever updates are made to the Shipment it remains in Approved Status.
- Cancelled – Cancelled status is used when the shipment is cancelled in Procure to Pay.
- Received – This status is shown once the Shipment has been received in MFCS.

Integration High Level

Figure 13-6 Procure to Pay Shipments Integration Overview

SPPP - Shipments - Integration Overview



Note:

Shipments Java Service integration via RIB-EXT is an optional flow to integrate with SIOCS or WMS that requires enabling from the configuration. For more details on this process, see the [Supply Chain Hub Frameworks](#) section.

Configuration

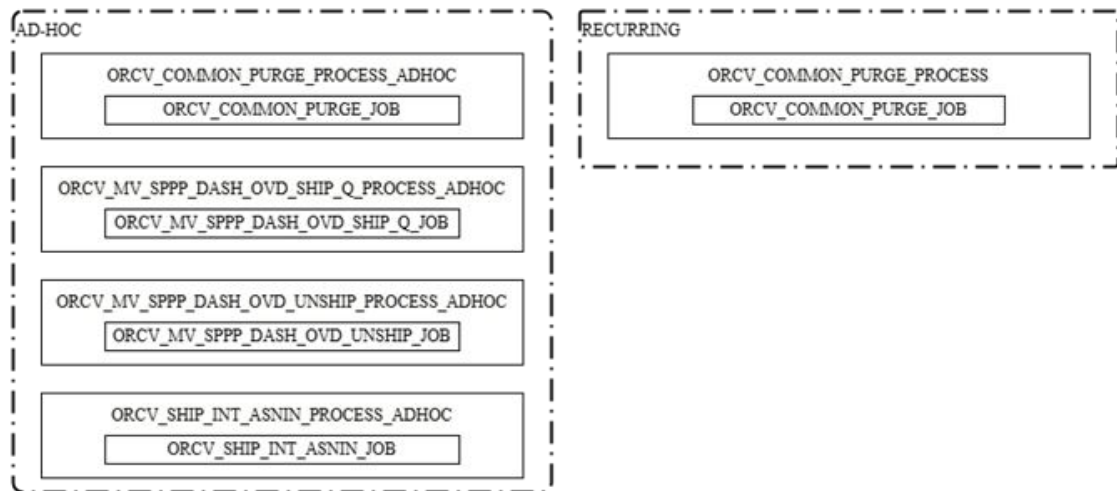
This section describes configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Schedule Overview

Figure 13-7 Procure to Pay Shipments Batch Schedule Overview



Other implementation Considerations

This section describes implementation considerations.

Custom Validations

The module performs validations by implementing Custom Validation Groups of the Custom Validations Framework described in the [Supply Chain Hub Frameworks](#) section. The groups listed below can be used to add Custom Validations for Purchase Orders.

Group ID	Executed When
SPPP_ASN_CANCEL	Shipment is cancelled
SPPP_ASN_SUBMIT	Shipment is created or updated

Materialize Views

This application requires some results to be materialized in tables using the Materialized View Framework described in the [Supply Chain Hub Frameworks](#) section, below is the list for this application. Materialized views used in Procure to Pay, Purchase Orders, are listed in this section.

ORCV_MV_SPPP_DASH_OVD_UNSHIP

Materialize View to support Supply Chain Hub dashboard “Overdue Commitment (Units) - Top 5” showing Purchase Orders without shipments by department that are expected to be already shipped.

Pre-Dependencies: None**Post-Dependencies:** None**Frequency:** Ad-hoc.

ORCV_MV_SPPP_DASH_OVD_SHIP_Q

Materialize View to support Supply Chain Hub dashboard showing Purchase Orders with shipments with quantities remaining.

Pre-Dependencies: None**Post-Dependencies:** None**Frequency:** Ad-hoc.

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
ASST	Shipment status.
ASRC	Shipment reason codes

Templates

This section describes templates.

Template SPPP ASNIN Data

Table 13-5 Template Key: SPPP_ASNIN_DATA

ACTION TYPE	ROLE NAME
Download	ORCV_SPPP_ASN_APPR_JOB
Download	ORCV_SPPP_ASN_VIEW_JOB
Download Blank	ORCV_SPPP_APP_JOB
Upload	ORCV_SPPP_ASN_APPR_JOB

Table 13-6 ASN Worksheet

ACTI ON	STATUS	MANDATORY CELLS / FIELDS	NOTES
Creat e	New	Action, ASN, Supplier, To Location Type, To location, Shipment Date, Estimated Arrival Date	Creates an ASN with approved status.

Table 13-6 (Cont.) ASN Worksheet

ACTI ON	STATUS	MANDATORY CELLS / FIELDS	NOTES
Upda te	Approve	Action, ASN, Supplier, To Location Type, To location, Shipment Date, Estimated Arrival Date	The action needs to be “Update” at the ASN Container worksheet even if the update is only at the container or item level. Updates the ASN.
Canc el	Approve	Action, ASN, Supplier, To Location Type, To location, Shipment Date, Estimated Arrival Date	Cancel an ASN.

Table 13-7 ASN Container Worksheet

ACTI ON	STATUS	MANDATORY CELLS / FIELDS	NOTES
Creat e	New	Action, ASN, Container	Adds container to the ASN.
Upda te	Approve	Action, ASN, Container	The action needs to be “Update” in the ASN worksheet even if the update is only at the container level. Updates the container.
Canc el	Approve	Action, ASN, Container	Removes all quantities for the items associated with the container.

Table 13-8 ASN Items Worksheet

ACTI ON	STATUS	MANDATORY CELLS / FIELDS	NOTES
Creat e	New	Action, ASN, Order No, Item, Description, Shipped Quantity, Received Quantity	Adds items to the ASN.

Table 13-8 (Cont.) ASN Items Worksheet

ACTI ON	STATUS	MANDATORY CELLS / FIELDS	NOTES
Upda te	Approve	Action, ASN, Order No, Item, Description, Shipped Quantity, Received Quantity	The action needs to be "Update" in the ASN worksheet and ASN Container worksheet if the item is associated with a container even if the update is only at the item level.
Canc el	Approve	Action, ASN, Order No, Item, Description, Shipped Quantity, Received Quantity	The item quantity is adjusted to 0. If the remaining quantity for the ASN is null the cancel action should be done at ASN worksheet level.

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
ASAC	Shipments Actions
ASST	Shipment Status

Interactive Grid Additional Columns

This application supports adding additional fields to the Interactive Grids by using the Interactive Grid Additional Columns functionality described in the [Supply Chain Hub Frameworks](#) section.

The Interactive Grids that support this functionality are listed below:

Interactive Grid	Grid ID	User Types	Page	Non Based View	Entity Primary Key
Shipments Dashboard Grid	P1_IG_SHIPMENTS	Retail Administrator, Retailer User, Supplier Administrator, Supplier User	Home	ORCV_V_SPPP_ASN_NB	ASN_NO

Interactive Grid	Grid ID	User Types	Page	Non Based View	Entity Primary Key
Purchase Orders Grid	P3_IG_PO	Retail Administrator, Retailer User, Supplier Administrator, Supplier User	Create or Edit ASN	ORCV_V_SPPP_ASN_NB	ASN_NO

References

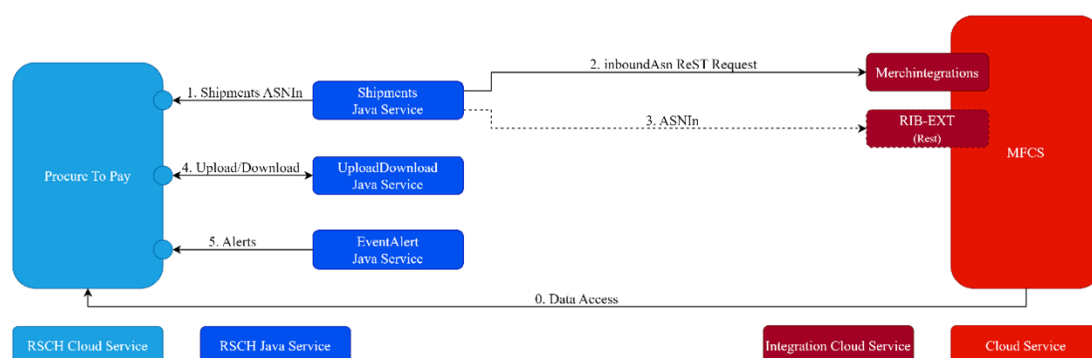
Procure to Pay, Shipments, also uses the Queries module to enable communication between retailer and supplier users, details about Queries are available in the [Supply Chain Hub Frameworks](#) section. It also has dependencies from other modules batch jobs listed in POM Configuration that are documented in the [Supply Chain Hub Frameworks](#) section.

Operations

This section describes operations.

Integration by Module

Figure 13-8 Procure to Pay Shipments Integration by Module



STEP	DESCRIPTION
0	RSCH Data Access of MFCS data
1	Shipments Java Services checks if Procure to Pay has shipments to be published
2	If there are ASNIn to be published, Shipments Java Services sends to MFCS via RIB-EXT REST
3	Procure to Pay calls Upload Download Java Service to transform files, depending on the request details, Upload Download Java Service might need to callback to Procure to Pay to get more details.
4	Event Alert Java Service checks if there are alerts to be triggered.

Reference

The Shipments ASNIN integration service will support the integration of ASNINs managed in the Procure to Pay module into MFCS via RIB-EXT REST. For more details on this process, consult the [Supply Chain Hub Frameworks](#) section.

Materialized Views Refresh Detail

This section describes the Materialized Views Refresh Details for Shipments.

ORCV_MV_SPPP_DASH_OVD_SHIP_Q

Parameter	Description	Value
Result Synonym	Synonym name pointing to the latest result table.	ORCV_MV_SPPP_DASH_OVD_SHIP_Q
Result Table A	Name of _A result table.	ORCV_SPPP_DASH_OVD_SHIP_Q_A_R
Result Table B	Name of _B result table.	ORCV_SPPP_DASH_OVD_SHIP_Q_B_R
Source View	View used to refresh the result tables.	ORCV_V_SPPP_DASH_OVD_SHIP_Q_R
Driving View	View used to distribute the refresh of result tables.	ORCV_V_SPPP_DASH_OVD_SHIP_Q_RC
Pre-Processor	Pre-process procedure to execute, if applicable.	ORCV_COMMON_MV_REFRESH. COMMON_PRE_PROCESSOR
Check Function	Check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if restart is allowed or not.	N
Max Concurrent Threads	The maximum concurrent threads to refresh result tables.	4
Filter Fields	Column(s) to distribute the refresh of result tables.	DEPT, SUPPLIER

ORCV_MV_SPPP_DASH_OVD_UNSHIP

Parameter	Description	Value
Result Synonym	Synonym name pointing to the latest result table.	ORCV_MV_SPPP_DASH_OVD_UNSHIP
Result Table A	Name of _A result table.	ORCV_SPPP_DASH_OVD_UNSHIP_A_R
Result Table B	Name of _B result table.	ORCV_SPPP_DASH_OVD_UNSHIP_B_R
Source View	View used to refresh the result tables.	ORCV_V_SPPP_DASH_OVD_UNSHIP_R

Parameter	Description	Value
Driving View	View used to distribute the refresh of result tables.	ORCV_V_SPPP_DASH_OVD_UNSHIP_RC
Pre-Processor	Pre-process procedure to execute, if applicable.	ORCV_COMMON_MV_REFRESH . COMMON_PRE_PROCESSOR
Check Function	Check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if restart is allowed or not.	N
Max Concurrent Threads	The maximum concurrent threads to refresh result tables.	4
Filter Fields	Column(s) to distribute the refresh of result tables.	DEPT, SUPPLIER

Finance Documents

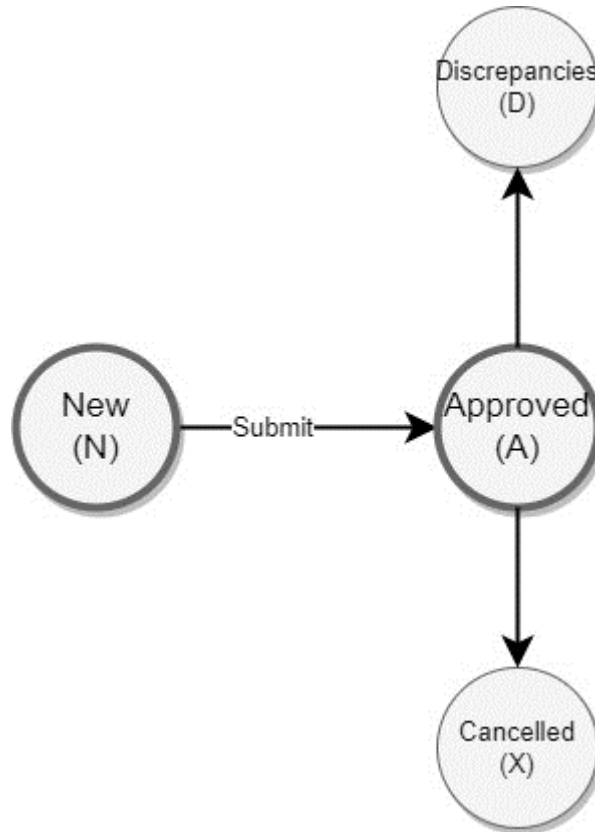
This section describes the Finance Documents features.

Implementation Overview

Procure to Pay, Purchase Orders, will allow retailers and suppliers to create and maintain financial documents.

Status Workflow

Figure 13-9 Procure to Pay Finance Documents Status Workflow



Finance Documents follow a workflow path, and the status can change depending on:

- The user permissions

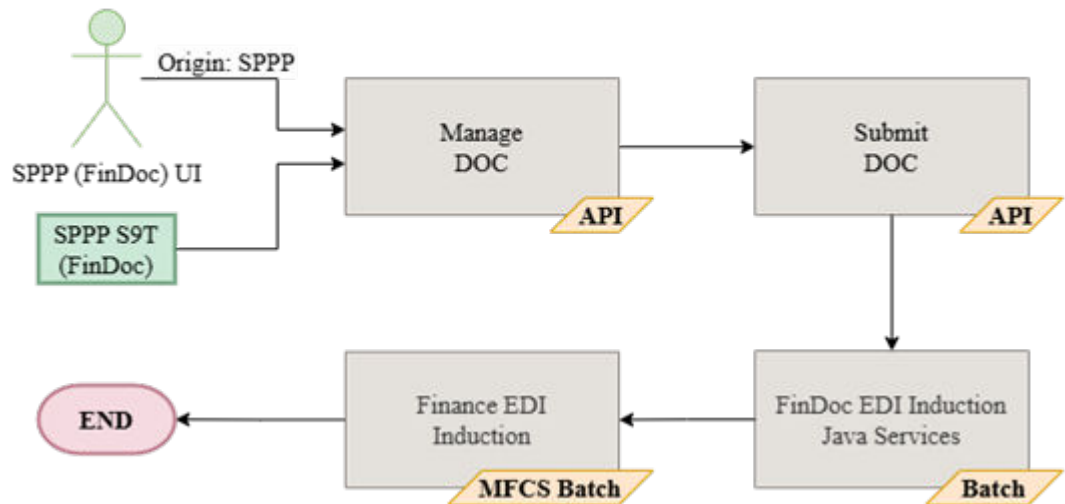
Finance Document Status:

- New – New is the initial status while creating the Document in SPCH, this status is only an interim status and is not possible to save a Finance Document in this status.
- Approved – This status is used once a Finance Document is submitted and integrated into IMCS.
- Discrepancies – This status is used if any discrepancies are found during the matching process in IMCS.
- Cancelled – Cancelled status is used if the Finance Document is cancelled in IMCS

Integration High Level

Figure 13-10 Procure to Pay Finance Documents Integration Overview

SPPP - Finance Documents - Integration Overview



Configuration

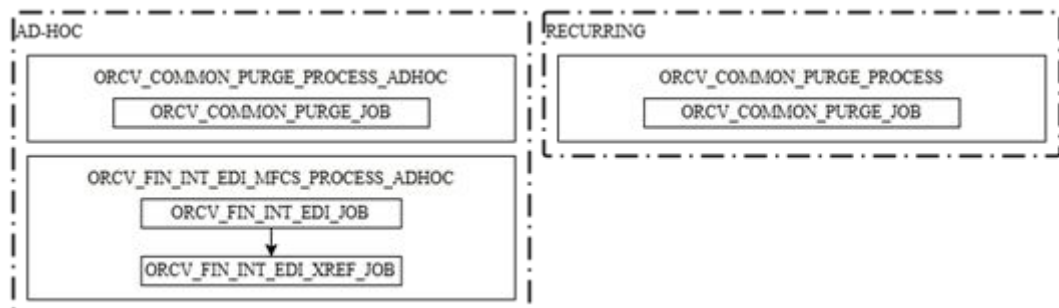
This section describes configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Schedule Overview

Figure 13-11 Procure to Pay Finance Documents Batch Schedule Overview



Other Cloud Services Configuration

This section describes implementation details that must be reviewed from an Oracle Retail Merchandising System perspective.

System Options

MFCS System Option LOCALIZATION_CONFIG_OPTIONS.DEFAULT_TAX_TYPE defines the default tax type used for Procure to Pay, Finance Documents.

Other implementation Considerations

This section describes implementation considerations.

Custom Validations - Finance Documents

The module performs validations by implementing Custom Validation Groups of the Custom Validations Framework described in the [Supply Chain Hub Frameworks](#) section. The groups listed below can be used to add Custom Validations for Finance Documents.

Group Id	Executed When
SPPP_FINDOC_SUBMIT	Financial Document is created

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
DOCC	Finance Document – Doc types, create mode
DOCV	Finance Document – Doc types, view mode
FDST	Finance Documents status.

Templates

This section describes templates.

Template SPPP Finance Document Data

One of these templates will be active, according to the configuration that exists in the MFCS for the default tax type.

Table 13-9 Templates

TEMPLATE KEY	STATUS
SPPP_DOC_SVAT_DATA	Active (if MFCS system option for default tax type is SVAT) Inactive (if MFCS system option for default tax type is different than SVAT)

Table 13-9 (Cont.) Templates

TEMPLATE KEY	STATUS
SPPP_DOC_SALES_DATA	Active (if MFCS system option for default tax type is SALES) Inactive (if MFCS system option for default tax type is different than SALES)
SPPP_DOC_GTS_DATA	Active (if MFCS system option for default tax type is GTS) Inactive (if MFCS system option for default tax type is different than GTS)

Table 13-10 Template Key: SPPP_DOC_SVAT_DATA or SPPP_DOC_SALES_DATA or SPPP_DOC_GTS_DATA

ACTION TYPE	ROLE NAME
Upload	ORCV_SPPP_FINDOC_APPR_JOB
Download Blank	ORCV_SPPP_APP_JOB

Table 13-11 Template Worksheets

WORKSHEET KEY	WORKSHEET NAME	SEQUENCE
SPPP_DOCS	Documents	1
SPPP_DOC_REFS	Document References	2
SPPP_DOC_ITEMS	Document Items	3
SPPP_DOC_VAT	Document VAT	4
SPPP_DOC_ITEM_TAX	Document Item Tax	4

**Note:**

In case MFCS default tax type equals 'SVAT' the SPPP_DOC_VAT worksheet is used, in case of 'GTS' SPPP_DOC_ITEM_TAX is used, otherwise none of the previous worksheets are used.

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
FDAC	Finance Document Actions
FDOR	Finance Document reference type

Interactive Grid Additional Columns

This application supports adding additional fields to the Interactive Grids by using the Interactive Grid Additional Columns functionality described in the [Supply Chain Hub Frameworks](#) section.

The Interactive Grids that support this functionality are listed below:

Interactive Grid	Grid ID	User Types	Page	Non Based View	Entity Primary Key
Invoices Dashboard Grid	P1_IG_INVOICES	Retail Administrator, Retailer User, Supplier Administrator, Supplier User	Home	ORCV_V_SPPP_DOC_NB	DOC_ID, SUPPLIER
Purchase Orders Grid	P4_IG_PO	Retail Administrator, Retailer User, Supplier Administrator, Supplier User	Create or Edit Document	ORCV_V_SPPP_ORD_NB	ORDER_NO
Purchase Orders/ Return to Vendor Grid	P4_IG_ORDRTV	Retail Administrator, Retailer User, Supplier Administrator, Supplier User	Create or Edit Document	ORCV_V_SPPP_ORD_NB	ORDER_NO

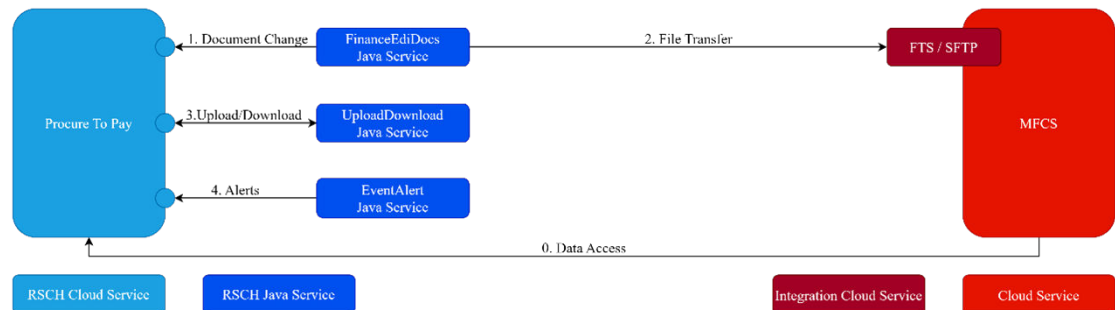
References

Procure to Pay, Finance Documents, also use the Queries module to enable communication between retailer and supplier users, details about Queries are available in the [Supply Chain Hub Frameworks](#) section. It also has dependencies from other modules batch jobs listed in POM Configuration that are documented in the [Supply Chain Hub Frameworks](#) section.

Operations

This section describes operations.

Integration by Module

Figure 13-12 Procure to Pay Finance Documents Integration by Module

STEP	DESCRIPTION
0	RSCH Data Access of MFCS data
1	Finance Document Induction Java Services checks if Procure to Pay has documents to be published
2	Finance Document Induction Java Services sends produced files to Integration File Transfer.
3	Integration File Transfer publishes files to MFCS File Transfer Services (FTS) or MFCS shared folder (depending on configuration).
4	Procure to Pay calls Upload Download Java Service to transform files, depending on the request details, Upload Download Java Service might need to callback to Procure to Pay to get more details.
5	Event Alert Java Service checks if there are alerts to be triggered.

Reference

The EDI finance document injector service will support the integration of Invoices and Credit Notes managed in the Procure to Pay module into IMCS via induction batch. For more details on this process, please consult the [Supply Chain Hub Frameworks](#) section.

Return to Vendor

The Return to Vendor module is an application available in Supply Chain Hub. This chapter describes the implementation considerations and the operation details for this application.

Implementation Overview

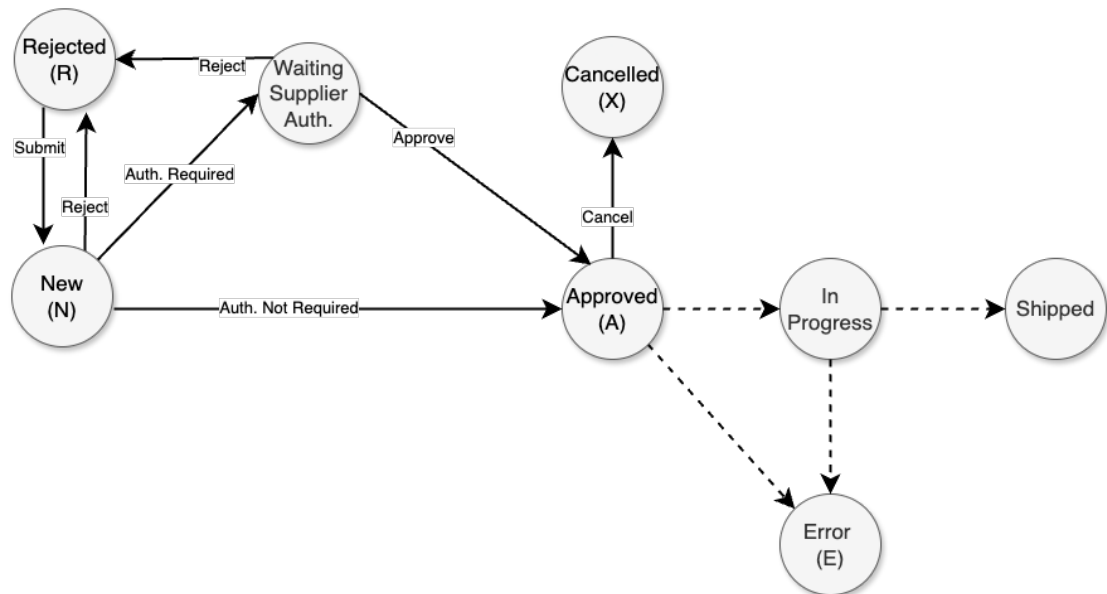
Return to Vendor application gives the possibility to create and maintain returns to vendors (RTV).

Functionality

The RTV's can be created, updated, rejected, approved or cancelled in Return to Vendor module. When RTV's are approved or cancelled they are published to MFCS. All RTV's, even those created in MFCS, can be cancelled in Return to Vendor module.

Status Workflow

Figure 14-1 Return to Vendor Status Workflow



Return to Vendor follows a workflow path for RTV, and the status can change depending on:

- The user permissions.
- The current status of the RTV.

RTV Status:

- New (N) – Initial Status after submitting a new RTV.
- Waiting Authorization Code (W) - Status that denotes that the RTV is missing the Supplier Authorization code.
- Rejected (R) - Status used when a RTV is Rejected.
- Approved (A) - Status that denotes that the RTV creation is complete and ready to be shipped.
- Shipped (S) - Status when RTV was shipped to destination.
- Cancelled (X) - Status when RTV is Cancelled.

Configuration

This section describes configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Schedule Overview

Figure 14-2 Figure 14: Return to Vendor Batch Schedule Overview



Other Cloud Services Configuration

This section describes implementation details that must be reviewed from an Oracle Retail Merchandising System perspective.

Reason Codes

The Reason Codes defined in MFCS CODE_HEAD and CODE_DETAIL tables with code type RTVR (Return to Vendor Reasons) are used to show LOVs on the screen and on S9T.

Batches

During the inventory phase of the MFCS batch execution, the integration of RTVs is not possible and Return To Vendor will not try to integrate during that phase.

Others

The supplier's configuration in table SUPS column RET_AUTH_REQ of MFCS defines if the RTV requires the Authorization Code as mandatory field to be Approved.

Other Implementation Considerations

This section describes Return to Vendor implementation considerations.

Custom Validations

The module performs validations by implementing Custom Validation Groups of the Custom Validations Framework described in the [Supply Chain Hub Frameworks](#) section. The groups are listed below.

SPRV_RTV_SUBMIT

Triggered when the Return to Vendor action is Submitted.

No custom validations exist by default for these custom validations group but can be added.

SPRV_RTV_REJECT

Triggered when the Return to Vendor action is Rejected.

No custom validations exist by default for these custom validations group but can be added.

SPRV_RTV_APPROVE

Triggered when the Return to Vendor action is Approved.

No custom validations exist by default for these custom validations group but can be added.

SPRV_RTV_CANCEL

Triggered when the Return to Vendor action is Canceled.

No custom validations exist by default for these custom validations group but can be added.

Materialized Views

This application requires some results to be materialized in tables using the Materialized View Framework described in the [Supply Chain Hub Frameworks](#) section, below is the list for this application. Materialized views used in Cost Change Management are listed in this section.

ORCV_MV_SPRV_RTV_STAT_APPR

The materialized view has the IMCS RTVs in status 10 - Approved

Pre-Dependencies: N/A

Post-Dependencies: N/A

Frequency: Ad-hoc

ORCV_MV_SPRV_RTV_STAT_SHPD

The materialized view has the IMCS RTVs in 15 - Shipped status.

Pre-Dependencies: N/A

Post-Dependencies: N/A

Frequency: Ad-hoc

ORCV_MV_SPRV_RTV_BY_DEPT

The materialized view has the total RTV Order Amount by department from the last 30 days.

Pre-Dependencies: N/A

Post-Dependencies: N/A

Frequency: Ad-hoc

ORCV_MV_SPRV_RTV_BY_LOC

The materialized view has the Total RTV Order Amount by location from the last 30 days.

Pre-Dependencies: N/A

Post-Dependencies: N/A

Frequency: Ad-hoc

ORCV_MV_SPRV_DASH_RTV_PENDNG

Returns the number of RTVs by supplier without Authorization code. Used in the Supply Chain Hub homepage.

Pre-Dependencies: N/A

Post-Dependencies: N/A

Frequency: Ad-hoc

Codes

Certain operations require codes to define the options available. Below you can find the list of code types available. The codes are defined in tables ORCV_CODE_HEAD and ORCV_CODE_DETAIL.

CODE TYPE	CODE TYPE DESCRIPTION
RTVC	Reason Codes to Cancel an RTV
RTVR	Reason Codes to Reject an RTV
RVAI	Available levels for Add Items to RTVs
RVAR	Return to Vendor Action Reason Code
RVDL	Return to Vendor Dashboard Labels
RVEL	Return to Vendor error levels to be used in the Issues List.
RVES	Return to Vendor error sources to be used in the error screen.

Templates

Return to Vendor uses a spreadsheet download process, developed as part of the Retail Supply Chain Hub Frameworks for extraction of data. To support those functions, several different configurations should be reviewed as part of the implementation.

Template Return to Vendor

Table 14-1 Template key: ORCV_SPRV_RTV

ACTION TYPE	ROLE NAME
Download	ORCV_SPRV_RTV_APPR_JOB
Download	ORCV_SPRV_RTV_EDIT_JOB
Download	ORCV_SPRV_RTV_VIEW_JOB
Download Blank	ORCV_SPRV_APP_JOB
Upload	ORCV_SPRV_RTV_APPR_JOB

Table 14-2 Worksheet RTV Header

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Action RTV Order Location Supplier	Creates new RTV Order.
Update	Action RTV Order Location Supplier	Update RTV Order.
Approve	Action RTV Order Location Supplier Authorization Code	Approve RTV Order.
Cancel	Action RTV Order Location Supplier	Cancel RTV Order.
Reject	Action RTV Order Location Supplier	Reject RTV Order.

Table 14-3 Worksheet RTV Details

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Action RTV Order Item or Barcode Quantity Unit Cost Reason Inventory Type (If Reason is Unavailable)	Add a new RTV Order Item.
Update	Action RTV Order Item or Barcode Quantity Unit Cost Reason Inventory Type (If Reason is Unavailable)	Update RTV Order Item.
Delete	Action RTV Order Item or Barcode Quantity Unit Cost Reason Inventory Type (If Reason is Unavailable)	Delete RTV Order Item.

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_S9T_CODE_HEAD and ORCV_S9T_CODE_DETAIL tables.

Code Type	Code Type Description
RTVD	Return To Vendor - Action Detail
RTVH	Return To Vendor - Action Header
RTVR	Return to Vendor - RTV Reason Codes

References

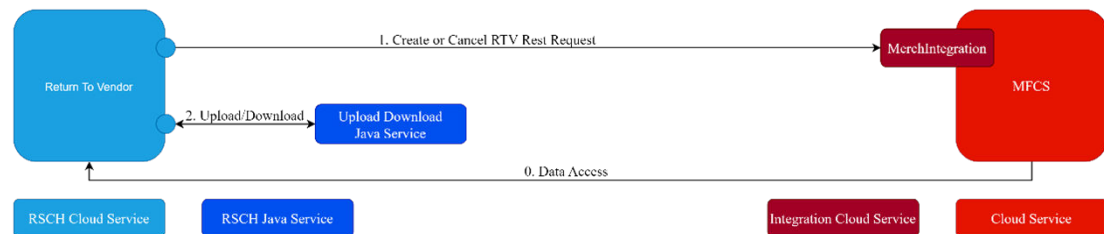
Return to Vendor also uses Queries module to enable communication between retailer and supplier users, details about Queries are available in the [Supply Chain Hub Frameworks](#) section.

Operations

This section describes available Return to Vendor operations.

Integration by Module

Figure 14-3 Return to Vendor Integration by Module



Step	Description
0	RSCH Data Access of MFCs data
1	Return to Vendor publishes Approved or Cancelled RTVs to MFCs MerchIntegrations Rest Service.
2	Return to Vendor calls Upload Download Java Service to transform files, depending on the request details, Upload Download Java Service might need to callback to Return to Vendor to get more details.

Materialized Views Refresh Details

This section describes Materialized Views Refresh Details.

ORCV_MV_SPRV_RTV_STAT_APPR

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPRV_RTV_STAT_APPR
Result Table A	Name of _A result table.	ORCV_SPRV_RTV_STAT_APPR_A_R
Result Table B	Name of _B result table.	ORCV_SPRV_RTV_STAT_APPR_B_R
Source View	The View contains business logic to be used to populate the result table.	ORCV_V_SPRV_RTV_STAT_APPR_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPRV_RTV_STAT_APPR_RC
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH . COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA

COLUMN	DESCRIPTION	VALUE
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	SUPPLIER

ORCV_MV_SPRV_RTV_STAT_SHPD

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPRV_RTV_STAT_SHPD
Result Table A	Name of _A result table.	ORCV_SPRV_RTV_STAT_SHPD_A_R
Result Table B	Name of _B result table.	ORCV_SPRV_RTV_STAT_SHPD_B_R
Source View	The View contains business logic to be used to populate the result table.	ORCV_V_SPRV_RTV_STAT_SHPD_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPRV_RTV_STAT_SHPD_RC
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH . COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	SUPPLIER

ORCV_MV_SPRV_RTV_BY_DEPT

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPRV_RTV_BY_DEPT
Result Table A	Name of _A result table.	ORCV_SPRV_RTV_BY_DEPT_A_R
Result Table B	Name of _B result table.	ORCV_SPRV_RTV_BY_DEPT_B_R

COLUMN	DESCRIPTION	VALUE
Source View	The View contains business logic to be used to populate the result table.	ORCV_V_SPRV_RTV_BY_DEPT_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPRV_RTV_BY_DEPT_RC
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH. COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	SUPPLIER

ORCV_MV_SPRV_RTV_BY_LOC

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPRV_RTV_BY_LOC
Result Table A	Name of _A result table.	ORCV_SPRV_RTV_BY_LOC_A_R
Result Table B	Name of _B result table.	ORCV_SPRV_RTV_BY_LOC_B_R
Source View	The View contains business logic to be used to populate the result table.	ORCV_V_SPRV_RTV_BY_LOC_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPRV_RTV_BY_LOC_RC
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH. COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	SUPPLIER

ORCV_MV_SPRV_DASH_RTV_PENDNG

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPRV_DASH_RTV_PENDNG
Result Table A	Name of _A result table.	ORCV_SPRV_DASH_RTV_PENDNG_A_R
Result Table B	Name of _B result table.	ORCV_SPRV_DASH_RTV_PENDNG_B_R
Source View	The View contains business logic to be used to populate the result table.	ORCV_V_SPRV_DASH_RTV_PENDNG_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPRV_DASH_RTV_PENDNG_RC
Pre- Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH. COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	SUPPLIER

15

Inventory and Sales

The Inventory and Sales module is an application available in Supply Chain Hub. This chapter describes the implementation considerations and the operation details for this application.

Functionalities

Inventory and Sales is an application that shows the stock and sales details of items per location. The homepage of Inventory and Sales has the top selling departments and the low stock, and no stock items counts.

Configuration

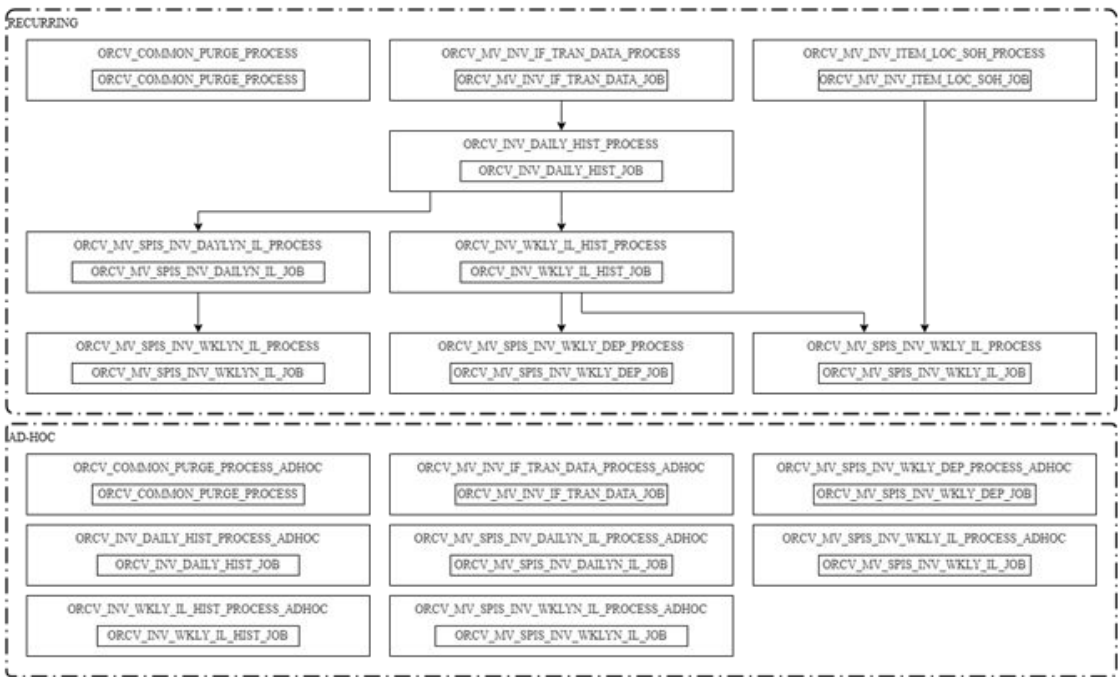
This section describes the configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Schedule Overview

Figure 15-1 Inventory and Sales Batch Schedule Overview



Other Implementation Considerations

This section describes Inventory and Sales implementation considerations.

Materialized Views

This application requires some results to be materialized in tables using the Materialized View Framework described in the [Supply Chain Hub Frameworks](#) section and below is the list for this application.

ORCV_MV_SPIS_INV_DAILY_IL

Contains the item-location stock on hand and sales by units' values in a daily basis.

Pre-Dependencies: ORCV_INV_DAILY_HIST_JOB

Post-Dependencies: ORCV_MV_SPIS_INV_WKLYN_IL_JOB

Frequency: Ad-hoc, Nightly

ORCV_MV_SPIS_INV_WKLYN_IL

Contains the item-location stock on hand and sales by unit's values from the n weeks.

Pre-Dependencies: ORCV_MV_SPIS_INV_DAILYN_IL_JOB

Post-Dependencies: None

Frequency: Ad-hoc, Nightly

ORCV_MV_SPIS_INV_WKLY_DEP

Departments with most sales by units from the last week. It will contain 5 rows (the top 4 performers plus one row for all the others).

Pre-Dependencies: ORCV_INV_WKLY_IL_HIST_JOB

Post-Dependencies: None

Frequency: Ad-hoc, Nightly

ORCV_MV_SPIS_INV_WKLY_IL

Contains item-location stock on hand and sales by unit's values from the last weeks.

Pre-Dependencies: ORCV_INV_WKLY_IL_HIST_JOB,
ORCV_MV_INV_ITEM_LOC_SOH_JOB

Post-Dependencies: None

Frequency: Ad-hoc, Nightly

References

Inventory and Sales has dependencies on Inventory batch jobs that are listed below. The details of those dependencies are available in the [Supply Chain Hub Frameworks](#) section.

MODULE	PROGRAM NAME
Inventory	ORCV_INV_DAILY_IL_HIST_JOB
Inventory	ORCV_INV_WKLY_IL_HIST_JOB
Inventory	ORCV_MV_INV_ITEM_LOC_SOH_JOB

Operations

This section describes Inventory and Sales operations.

Materialized Views Refresh Details

This section describes Materialized Views Refresh Details.

ORCV_MV_SPIS_INV_DAILYN_IL

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPIS_INV_DAILYN_IL
Result Table A	Name of _A result table.	ORCV_SPIS_INV_DAILYN_IL_A_R
Result Table B	Name of _B result table.	ORCV_SPIS_INV_DAILYN_IL_B_R
Source View	The View that contains business logic to be used to populate the result table.	ORCV_V_SPIS_INV_DAILYN_IL_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPIS_INV_DAILYN_IL_RC
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH. COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	LOC

ORCV_MV_SPIS_INV_WKLYN_IL

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPIS_INV_WKLYN_IL
Result Table A	Name of _A result table.	ORCV_SPIS_INV_WKLYN_IL_A_R

COLUMN	DESCRIPTION	VALUE
Result Table B	Name of _B result table.	ORCV_SPIS_INV_WKLYN_IL_B_R
Source View	The View that contains business logic to be used to populate the result table.	ORCV_V_SPIS_INV_WKLYN_IL_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPIS_INV_WKLYN_IL_RC
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH. COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	LOC

ORCV_MV_SPIS_INV_WKLY_DEP

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPIS_INV_WKLY_DEP
Result Table A	Name of _A result table.	ORCV_SPIS_INV_WKLY_DEP_A_R
Result Table B	Name of _B result table.	ORCV_SPIS_INV_WKLY_DEP_B_R
Source View	The View that contains business logic to be used to populate the result table.	ORCV_V_SPIS_INV_WKLY_DEP_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPIS_INV_WKLY_DEP_RC
Pre Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH. COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	6
Filter Fields	Column name of the view to be used for threading.	DEPT

ORCV_MV_SPIS_INV_WKLY_IL

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPIS_INV_WKLY_IL
Result Table A	Name of _A result table.	ORCV_SPIS_INV_WKLY_IL_A_R
Result Table B	Name of _B result table.	ORCV_SPIS_INV_WKLY_IL_B_R
Source View	The View that contains business logic to be used to populate the result table.	ORCV_V_SPIS_INV_WKLY_IL_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPIS_INV_WKLY_IL_RC
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH. COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	LOC

16

Items Management

This section describes the common functionality and configurations available for Items Management.

Implementation Overview

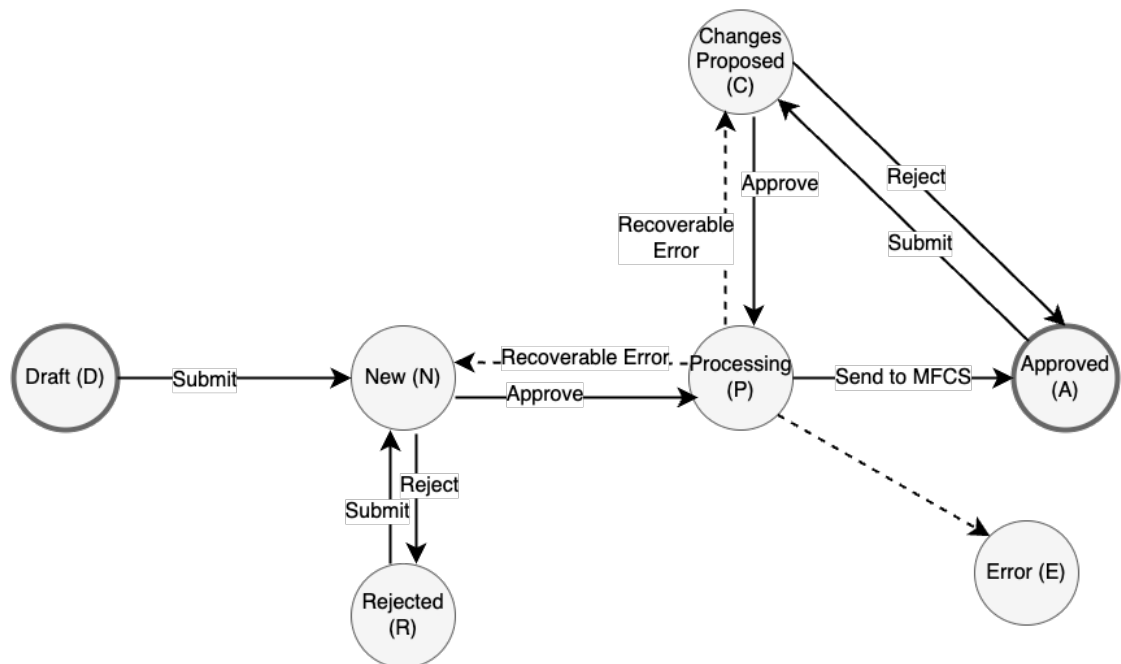
This section provides an implementation overview.

Functionality

The Items Management module gives the possibility to create and maintain Items (either by UI or by file export/import) and publishes them to MFCS.

Status Workflow

Figure 16-1 Items Management Status Workflow



Items Management follows a workflow path, and the status can change depending on:

- The user permissions
- The current Status of the Item

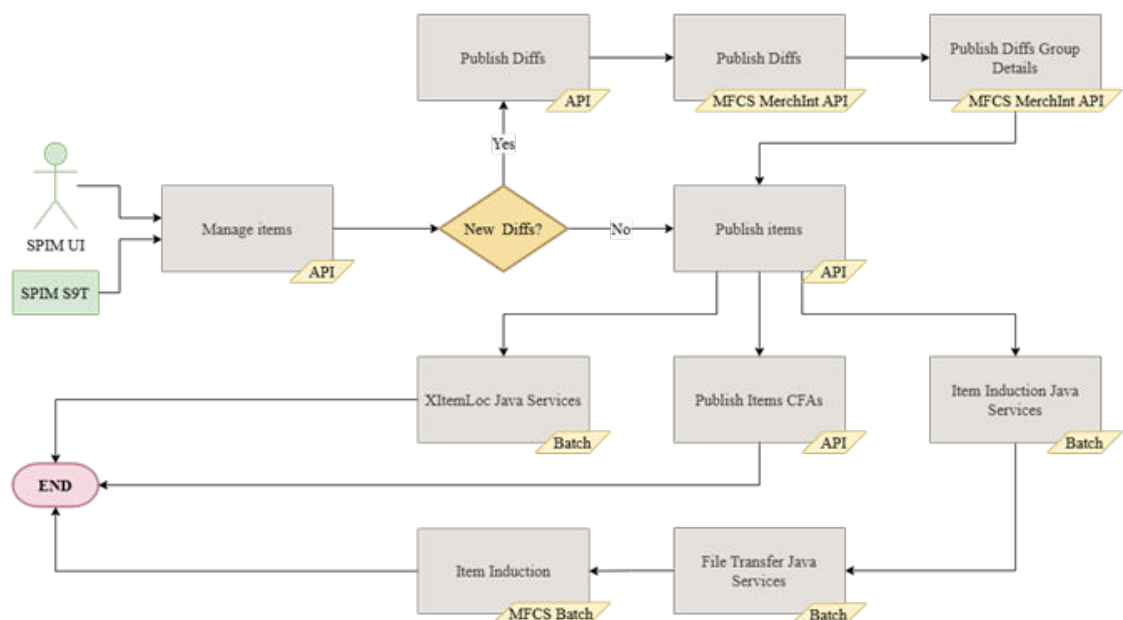
Item Status:

- **Draft** – Draft is the initial status of an item is when the creation process starts. At any time during in Draft a user can save and continue later, staying in draft status. When submitting if enough information is available to submit the Item Status can be changed to New, if the user does not have approval permissions, or directly to Approved Status (via Processing).
- **New** – New is the status of an item with a minimum amount of information to be created is in. In this status an Item can be Approved to send the item to MFCS or Rejected to change the status of the item to Rejected.
- **Rejected** – In Rejected Status an item can still be updated and submitted back to New Status, to restart the approval process if needed.
- **Approved** – An Approved Item is an Item that exists in MFCS and is complete to be used on all processes in SPCH.
- **Changes Proposed** – Changes Proposed is a status used when a user without approve permissions updates an Approved Item. During this Status the changes can be approved and then sent to MFCS or Rejected and the changes are discarded.
- **Processing** – Processing Status is an interim status while an item is being interfaced with MFCS. Since the integration is not near real-time. The item is not editable whilst on this status. Once the integration is successful, the item will move to Approved Status. If during Processing an issue is encountered that is a recoverable error (data validations for example), the Item Status will be moved to the previous Status to allow the Item data to be fixed before processing the item again, if the issue found is non recoverable the Item will move to Error Status instead.
- **Error** – Error Status is used when an issue is encountered while interfacing the Item into MFCS and when in this status technical support will be required to understand what caused the error.

Integration High Level

SPIM Integration Overview

Figure 16-2 Items Management Integration High Level



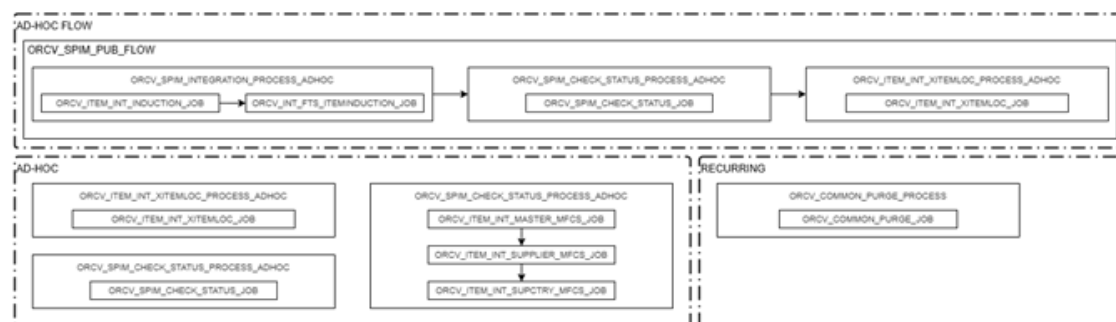
Configuration

This section describes the configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Figure 16-3 Items Management Batch Schedule Overview



Similar Items

Items Management enables the possibility of adding views configurations so they can return possible similar the new item being created and stop the item approval process until the changes are merged in the existing item, or the data that makes the new item look similar is changed.

To support it the System Parameter **NEW_ITEM_MERGE_RECOM** can be configured as Y and during the submit process the views configured in table **ORCV_SPIM_SIMILAR_ITEMS_VIEWS** are executed to insert possible matches in table **ORCV_SPIM_SIMILAR_ITEMS**. After executing the views with success and any rows are inserted, the process does not continue to approval and a new validation issue is raised to notify the user that they may have duplicates and must act.

When the user enters the new or existing item with possible duplicates a new button is displayed that opens a popup for the user to choose which of the existing duplicate items to merge the changes to. After confirming the new item is deleted, the changes merged into existing item and the existing item is sent for approval.

Database Objects

Views

The views that can be created must return the following two fields so that the process can use them to filter and insert in similar items tables. The views must be added to **ORCV_SPIM_SIMILAR_ITEMS_VIEWS**.

Column	Data Type	Notes
EXISTING_ITEM	VARCHAR2(25)	This fields contains the approved MFCS item that has a duplicated barcode in a SPIM unapproved item.
NEW_ITEM	VARCHAR2(25)	This fields contains the not approved SPIM item that has a duplicated barcode in a MFCS approved item.

Other Cloud Services Configuration

This section describes implementation details that must be reviewed from Oracle Retail Merchandising System perspective.

System Options

To ensure that Items can correctly integrate with MFCS some configurations in the MFCS configuration table CORESVC_ITEM_CONFIG must be made. To allow Items Management to explode Item level 1 information to lower levels the following parameters must be set to N in CORESVC_ITEM_CONFIG.

PARAMETER	VALUE
CASCADE_UDA_DETAILS	N
CASCADE_IUD_ITEM_SUPP_COUNTRY	N
CASCADE_IUD_ISC_DIMENSIONS	N
CASCADE_IUD_ISMC	N
ISC_UPDATE_ALL_LOCS	N
ISC_UPDATE_ALL_CHILD_LOCS	N
CASCADE_VAT_ITEM	N
CASCADE_IIM_DETAILS	N
CASCADE_IUD_ITEM_SUPPLIER	N
CASCADE_ITEM HTS	N
CASCADE_ITEM HTS_ASSESS	N
CASCADE_ITEM_EXPENSES	N
CASCADE_ITEM_TICKET	N
CASCADE_ITEM_SEASONS	N
CASCADE_ITEM_UPCHRG	N

Furthermore, two of the configurations must be aligned with their corresponding configurations in the Items Management System Parameters in table ORCV_SYSTEM_PARAMETERS with area CONFIG and module Items Management.

MFCS CORESVC_ITEM_CONFIG	ORCV ORCV_SYSTEM_PARAMETERS
MAX_ITEM_RESV_QTY	RESERVE_ITEM_MAX_AMOUNT
MAX_ITEM_EXPIRY_DAYS	RESERVE_ITEM_MAX_EXPIRY

**Note:**

MAX_ITEM_RESV_QTY must be configured with at least value 10.

Reason Codes

Whenever a user updates an Item that is already approved, they need to specify a reason for that update. These reasons are identified by reason codes which are defined at system level. Below you can find the list of the codes available, the sequence and their description.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

Code Head

CODE TYPE	CODE TYPE DESCRIPTION
IMRR	SPIM revision reason code, to be used on the ORCV_SPIM_ITEM_REV_CTRL table.

Code Detail

CODE	CODE DESCRIPTION
D	Discontinue
C	Change Availability
U	UI Update Details
S	S9T Update Details
R	Reject
A	Approve

UDAs

Items Management uses some attributes that are specific to it but still need to be integrated with MFCS, to do so specific user-defined attributes need to be created in MFCS. The user attribute identifier on the table below is an implementation example.

These attributes are specified below.

UDA ID	UDA DESCRIPTION	UDA TYPE	UDA VALUE	UDA VALUE DESCRIPTION	SINGLE VALUE INDICATOR
9001	ORCV Item Integration Revision	FF (Free Form)	NA	NA	Y
9002	On Sale Date	DT (Date)	NA	NA	Y
9003	Off Sale Date	DT (Date)	NA	NA	Y
9004	Stock Available	LV (List of Values)	1	Y	Y
9004	Stock Available	LV (List of Values)	2	N	Y

**Note:**

Important to note is that these UDAs need to be aligned with the following System Parameters in ORCV_SYSTEM_PARAMETERS, if they need to have a different ID.

AREA	PARAMETER	DESCRIPTION	VALUE 1	VALUE 2
CONFIG	ITEM_INT_REVISION_UDA	UDA that must be used exclusively in the integration process.	9001	NA
CONFIG	EXCLUDED_UDA	UDAs that should be excluded from the user assigning to an item.	ON_SALE_DATE	9002
CONFIG	EXCLUDED_UDA	UDAs that should be excluded from the user assigning to an item.	OFF_SALE_DATE	9003
CONFIG	EXCLUDED_UDA	UDAs that should be excluded from the user assigning to an item.	STOCK_AVAIL	9004

Batches

The MFCS integration item induction job, IINDUCT_JOB, should be configured to execute with high frequency to assure that items changes made in Items Management are integrated into MFCS as soon as possible with final target the merchandising tables.

Other Implementation Considerations

This section describes Item Management implementation considerations.

Notifications and Alerts

Items Management uses Events Framework to inform users about events occurred by generating notifications and alerts. Below is the list of Notifications and Alerts available. The notifications are implemented through the Rule Engine Framework documented in the [Supply Chain Hub Frameworks](#) section.

Notifications

This section describes the notifications for Items Management.

ORCV_SPIM_EVENT_ITEM_RECOVERY

Notification triggered when user approves an item and the item fails the approval.

Property	Value
Event Name	SPIM_EVENT_ITEM_RECOVERY
Event Type	ORCV_SPIM_EVENT_ITEM_RECOVERY
Description	An error(s) occurred during the item(s) approval process.
Trigger	UI action
Target Users	User that approved the item.

ORCV_SPIM_EVENT_PROP_CHANG_EXIST_ITEM

Notification triggered when a change propose is submitted.

Property	Value
Event Name	SPIM_EVENT_PROP_CHANG_EXIST_ITEM
Event Type	ORCV_SPIM_EVENT_PROP_CHANG_EXIST_ITEM
Description	Items with new changes proposed by Retailer
Trigger	UI/S9T action
Target Users	All Retailer users with permissions for the item department or primary supplier will receive the notification.

ORCV_SPIM_EVENT_SUBM_NEW_ITEM

Notification triggered when a new item is submitted.

Property	Value
Event Name	SPIM_EVENT_SUBM_NEW_ITEM
Event Type	ORCV_SPIM_EVENT_SUBM_NEW_ITEM
Description	New Items submitted
Trigger	UI/S9T action
Target Users	All Retailer users with permissions for the item department or primary supplier will receive the notification.

Alerts

ORCV_SPIM_EVENT_ITEMS_PENDING

Email sent with the List of Items that were submitted 'x' days ago and still in 'New' or 'Changes Proposed'.

Property	Value
Event Name	SPIM_EVENT_ITEMS_PENDING_ACTION
Event Type	ORCV_SPIM_EVENT_ITEMS_PENDING
Alert Type	EMAIL
Description	Items with actions pending
Trigger	Rules Engine Batch with key: SPIM_ITEM_EVT_PND
Target Users	Retail Contact for the item Suppliers

Custom Validations

The application performs validations by implementing Custom Validation Groups of the Custom Validations Framework described in the [Supply Chain Hub Frameworks](#) section. The groups are listed below.

SPIM_ITEM_APPROVE

Triggered when the user approves an Item by UI or S9T.

All validation functions for this Custom Validation Group are available in ORCV_SPIM_MGMT_ITEM_VAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_HEADER	Validate item header data.	1
VALIDATE_IMPORT	Validate item import data.	2
VALIDATE_IMAGE	Validate item image data.	3
VALIDATE_PACKITEM	Validate pack item data.	4
VALIDATE_PRICING	Validate item pricing data.	5
VALIDATE_RANGING	Validate item ranging data.	6
VALIDATE_SUPPLIER	Validate item supplier data.	7
VALIDATE_BUY_ATTR	Validate item buy attributes.	8
VALIDATE_DIFFS	Validate item diffs.	9

SPIM_ITEM_SUBMIT

Triggered when the user submits an Item by UI or S9T.

All validation functions for this Custom Validation Group are available in ORCV_SPIM_MGMT_ITEM_VAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_HEADER	Validate item header data.	1
VALIDATE_IMPORT	Validate item import data.	2
VALIDATE_IMAGE	Validate item image data.	3
VALIDATE_PACKITEM	Validate pack item data.	4
VALIDATE_PRICING	Validate item pricing data.	5
VALIDATE_RANGING	Validate item ranging data.	6
VALIDATE_SUPPLIER	Validate item supplier data.	7
VALIDATE_DIFFS	Validate item diffs data	8
VALIDATE_BUY_ATTR	Validate item buy attributes.	9

Custom Flex Attributes

This application supports three different types of Custom Flex Attributes (CFA): Item Master, Item Supplier and Item Supplier Country. For each CFA there is a corresponding System Parameter to allow their use. For S9T, the worksheet must also be enabled.

Before importing CFA attribute configuration from MFCS, make sure the entities defined in ORCV_CFA_EXT_ENTITY match the entities on the MFCS side. To import the CFAs run the import batch. Afterwards the attributes will be synchronized every time a user accesses that CFA group in the UI. The import only needs to be run if there are any changes in the group sets or groups and the details are available in the [Supply Chain Hub Frameworks](#) section.

Specific Batch Configuration

This section describes specific batch configurations.

Check Status Batch

The Check Status batch is used to validate if an item was successfully published into MFCS.

- Job: ORCV_SPIM_CHECK_STATUS_JOB
- Frequency: Nightly
- Dependency: None
- Inter Scheduler Dependency: None

Batch configuration table: ORCV_BATCH_CONFIG

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_SPIM_CHECK_STATUS_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
IDTF	SPIM Display Type field code, to be used on the ORCV_SPIM_SELL_ATTR table.
IMCB	SPIM item Location Calculation Basis, to be used on the Add Locations popup screen.
IMEL	SPIM error level. To be used in the ORCV_SPIM_ITEM_ERRORS table.
IMIT	SPIM item type, to be used on the dashboard screen.
IMLT	SPIM item Location Type, to be used on the Add Locations popup screen.
IMRR	SPIM revision reason code, to be used on the ORCV_SPIM_ITEM_REV_CTRL table.
IMRS	SPIM revision status, to be used on the ORCV_SPIM_ITEM_REV_CTRL table.
IMST	SPIM item status, to be used on the dashboard screen.

Templates

Items Management uses a spreadsheet download and upload process, developed as part of the Retail Supply Chain Hub Frameworks, for creation, maintenance, and extraction of data. To support these processes there are some templates which must be configured, and several configurations should be reviewed as part of the implementation.

Template SPIM Item Data

Table 16-1 Template key: SPIM_ITEM_DATA

ACTION TYPE	ROLE NAME
Download	ORCV_SPIM_ITEM_APPR_JOB
Download	ORCV_SPIM_ITEM_EDIT_JOB
Download	ORCV_SPIM_ITEM_VIEW_JOB
Download Blank	ORCV_SPIM_APP_JOB
Upload	ORCV_SPIM_ITEM_APPR_JOB
Upload	ORCV_SPIM_ITEM_EDIT_JOB

Table 16-2 Worksheet Item Header

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Item or VPN, Item type, Supplier, Vendor Description, Item Description, Inventory, Orderable, Sellable, Department ID, Class ID, Subclass ID, Standard UOM	Creates a new Item. When the user has Edit permissions, it creates the Item with status New. When the user has Approve permissions, it creates the Item with status Approved. If other column or columns are mandatory, the enrichment process will populate the proper default values.
Update	Item or VPN Updatable: vendor description, Item description, brand ID, season, on sale date, off sale date, recommend retail price	Updates the Item with the new details. When a user with Edit permissions does the upload, if the previous status was New or Reject, the Item status will be New. If the previous status was Approved or Change Proposed, the status will be Change Proposed. When a user with Approve permissions does the upload, the Item status will be Approved.
Approve	Item or VPN	Approves items, if in Change Proposed approves the latest changes.
Reject	Item or VPN	Rejects all changes.

Table 16-3 Worksheet Item SKU

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Item parent or VPN parent, Item SKU or Primary VPN SKU, Diff Value 1 Diff Value 2 Diff Value 3 Diff Value 4	Adds item skus to the style (item parent). The diff values are mandatory if exists diff groups or diff ids associated to the item parent.
Update	Item parent or VPN parent, Item SKU Updatable: primary VPN SKU, primary barcode, primary barcode type	Updates the item SKU of the Item parent associated.

Table 16-4 Worksheet Item Barcodes

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Item SKU or VPN SKU, Barcode, Barcode Type, Primary Barcode	Adds item barcode to the SKU.
Update	Item SKU or VPN SKU Updatable: barcode, barcode type, primary barcode	Updates the barcode of the associated SKU item.
Delete	Item SKU or VPN SKU	Deletes the barcode of the associated SKU item.

Table 16-5 Worksheet Pack Components

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Item Parent or VPN Parent, Component, VPN Component, Qty	Adds item component to pack.

Table 16-6 Worksheet Item Supplier Country

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Item or VPN, Supplier, Primary Supplier, Origin Country, Primary Country, Unit cost or Purchase rate, Purchase Type	Adds supplier country to the Item. Unit cost or Purchase rate are mandatory depending on Purchase type selected.
Update	Item or VPN Updatable: lead time, pickup lead time, supplier pack size, packing method, min order qty, max order qty	Updates information related to the supplier country associated.

Table 16-7 Worksheet Item Supplier Country Dim

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Item or VPN, Supplier, Origin Country, Dim Object	Adds supplier country dimension to the Item.
Update	Item or VPN Updatable: item length, item width, item height, LWH UOM, weight, WEIGHT UOM	Updates information related to the supplier country dimensions associated.

Table 16-8 Worksheet Item Buying Attributes

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Item or VPN, UDA Type, UDA ID, UDA Date or UDA FF/LOV	Adds buying attributes to the Item. UDA Date or UDA FF/LOV depends on UDA Type selected.

Table 16-9 Worksheet Item Selling Attributes

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Item or VPN, Attribute, Value	Adds selling attributes to the Item.

Table 16-9 (Cont.) Worksheet Item Selling Attributes

ACTION	MANDATORY CELLS / FIELDS	NOTES
Update	Item or VPN Updatable: value	Updates the value of the attribute associated.

Table 16-10 Worksheet Item Import

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Item or VPN, Origin country, Import country, HTS	Adds HTS to the Item.

Table 16-11 Worksheet Item Range

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Item or VPN, Location, Status	Adds range to the Item. In case of the Purchase Type exceptions, it's necessary to fill the purchase type, if different from owned, then must fill the cost per unit or purchase rate (only one can be filled).
Update	Item or VPN Updatable: status	Updates the status of the location.

Table 16-12 Worksheet Item Retail by Zone

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Item or VPN, Zone ID, Retail price	Adds retail price per zone ID to the Item.

Table 16-13 Worksheet Item Image

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Item or VPN, Image name, Image address, Primary image, Display priority	Adds image to the Item.
Update	Item or VPN Updatable: primary image	Updates the primary image indicator.

Template SPIM Supplier Default

Table 16-14 Template Key: SPIM_SUPP_DFLT

ACTION TYPE	ROLE NAME
Download	ORCV_SPIM_ITEM_APPR_JOB
Download	ORCV_SPIM_ITEM_EDIT_JOB
Download	ORCV_SPIM_ITEM_VIEW_JOB
Download Blank	ORCV_SPIM_APP_JOB
Upload	ORCV_SPIM_ITEM_APPR_JOB

Table 16-15 Worksheet Supplier Defaults

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Supplier, Entity – Attribute, Value 1, Default Value	Creates a new attribute (selected from the entity-attribute list) for the supplier. Value 2, value 3 and value 4 can be filled according to selected entity attribute.
Update	Supplier, Entity – Attribute Updatable: value 1, value 2, value 3, value 4, default value	Updates the values and default value indicator for the entity-attribute to supplier related.
Delete	Supplier, Entity – Attribute, Value 1, Value 2, Value 3, Value 4	Deletes the entity-attribute for the supplier.

Table 16-16 Worksheet Supplier Merch Mappings

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Supplier, At least one of the following fields: Brand or Ref 1, Dept	Creates a new mapping between brand and/or supplier references (1 to 5) to the Merch Hierarchy.
Update	Supplier, Brand and/or Ref (1 to 5)	Updates the Merch Hierarchy mapping (department, class, subclass).
Delete	Supplier, Brand and/or Ref (1 to 5)	Deletes the Merch Hierarchy mapping.

Table 16-17 Worksheet Supplier Selling Attributes

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Supplier, Selling attribute ID, Dept, Value, Required	Adds selling attributes values to the supplier.
Update	Supplier, Selling attribute ID, Dept Updatable: value, required	Updates the supplier selling attributes.
Delete	Supplier, Selling attribute ID, Dept	Deletes the supplier selling attributes.

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_S9T_CODE_HEAD and ORCV_S9T_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
IHAC	Item Management Header - Item Data Action
IHIT	Item Management - Item Type
IDAC	Item Management - Item Data Action
IDUT	Item Management - Item UDA Type
IDID	Item Management - Item Detail Action
IPAC	Item Management - Pack Component Action
IHIY	Item Management - Indicator LOV (Yes)
SDAC	Supplier Defaults - Action
SDEA	Supplier Defaults - Entity Attribute

References

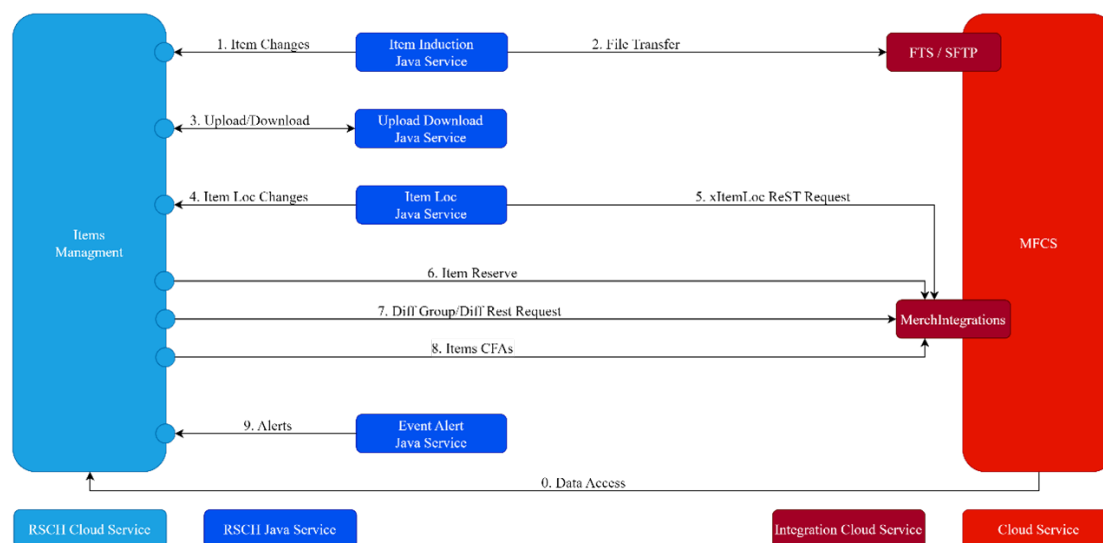
Items Management also uses the Queries module to enable communication between retailer and supplier users, details about Queries are available in the [Supply Chain Hub Frameworks](#) section. It also has the ability to attach documents to Items, if the user type is Retailer the attachments will be available for all Retailer and Supplier users of the Item and if the user is Supplier the attachments will be available for Retailer users and Supplier users with access to the same supplier.

Operations

This section describes Inventory and Sales operations.

Integration by Module

Figure 16-4 Items Management Integration by Module



STEP	DESCRIPTION
0	RSCH Data Access of MFCS data
1	Item Induction Java Services checks if Item Management has Items to be published and produces ODS files to be consumed by MFCS Item Induction Job.
2	Item Induction Java Services sends produced files to Integration File Transfer.
3	Integration File Transfer publishes files to MFCS File Transfer Services (FTS) or MFCS shared folder (depending on configuration).
4	Items Management calls Upload Download Java Service to transform files, depending on the request details, Upload Download Java Service might need to callback to Item Management to get more details.
5	Item Loc Java Services checks if Item Management has Items to be published and generates xItemLoc RIB messages to be integrated with MFCS.
6	Item Loc Java Services calls RIB-EXT Rest with generated xItemLoc messages.
7	Item Management calls Items MFCS ReST Merch Integration request for Item IDs.
8	Item Reserve Java Service calls MFCS ItemManagementService with a Rest request to request Item IDs.
9	Items Integration calling Diff Group and Diff Rest endpoint of MFCS ReST Merch Integration to create Diffs and Diff Details.
10	Event Alert Java Service checks if there are alerts to be triggered.

Batch by Module

This section describes the batch by module.

Check Item Status Batch

Authentication Method: OAuth2

BaseURL: https://<BASE_URL>/ords/mfcs/orcv/merch/itemsmanagement/batch/checkstatus/<URL_PATH>

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_SPIM_CHECK_STATUS_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPIM_ITEM_RESERVE_XREF ORCV_SPIM_ITEM_HEADER
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPIM_CHECK_STATUS_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPIM_ITEM_RESERVE_XREF ORCV_SPIM_ITEM_HEADER
log	GET	URL query parameters: /log? jobName=ORCV_SPIM_CHECK_STATUS_JOB &executionId=<EXEC_ID>&logDate=YYYY- DD-MM	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPIM_CHECK_STATUS_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPIM_ITEM_RESERVE_XREF ORCV_SPIM_ITEM_HEADER

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
status	GET	URL query parameters: /status? jobName=ORCV_SPIM_CHECK_STATUS_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A
status/all	GET	URL query parameters: /status?<EXEC_IDS>	N/A
config	GET	Empty	N/A

Alerts

Items Management sends alerts using the Rules Engine Framework documented in the [Supply Chain Hub Frameworks](#) section, below is the list of rules batch-based rules required to be executed in batch scheduler.

EVENT NAME	RULE	JOB
ORCV_SPIM_EVENT_ITEMS_PENDING	SPIM_ITEM_EVT_PND	ORCV_SPIM_ITEM_EVT_PND_JOB

Service by Module

This section describes the Service by Module.

Recalculate Defaults

This service will be used to update the default weights for the default mappings between the defined entities based on configuration defined in ORCV_SPIM_MERCH_MAP_CFG_WEIGHT.

Rest URL: /ords/mfcs/orcv/merch/itemsmanagement/recalculate/defaults

Weight	
Url Path	<Rest URL>/weight
Request Method	PUT
Sample Request Body	NA
Sample Success Message	{ "msg": "Success" } Response Code: 200 (Success)

Weight	
Sample Failure Message	{ "statusMsg": "Processed with error.", "errorMsg": "Invalid Column" } Response Code: 400 (Bad Request)
Affected Tables	ORCV_SPIM_MERCH_MAP_CFG_WEIGHT

Reference

Please refer to the [Supply Chain Hub Frameworks](#) section for operational details on integration and item reserve services.

Cost Changes Management

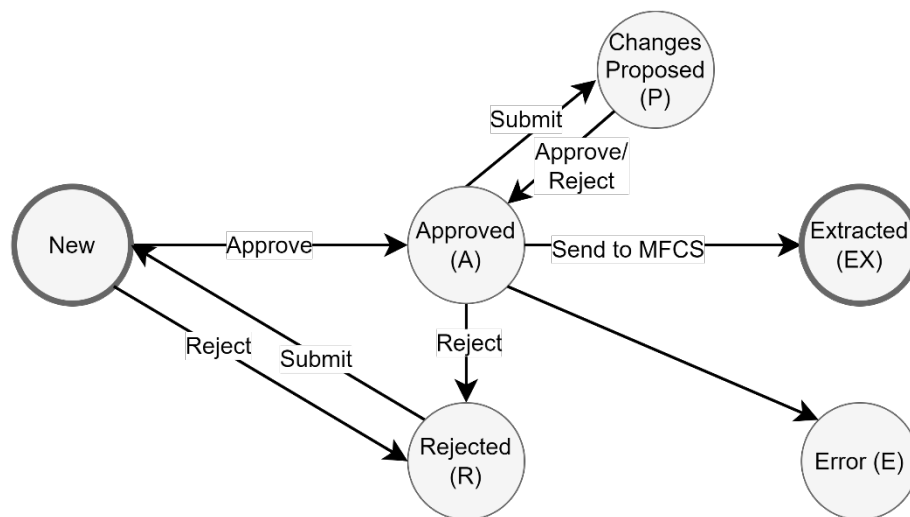
The Cost Changes Management module is an application available in Supply Chain Hub. This chapter describes the implementation considerations and the operation details for this application.

Implementation Overview

Cost Changes Management gives the possibility to create and maintain Cost Changes and before their effective date publish them to MFCS.

Status Workflow

Figure 17-1 Cost Changes Management Status Workflow



Cost Changes Management follows a workflow path, and the status can change depending on:

- The user permissions
- The status of the Cost Change

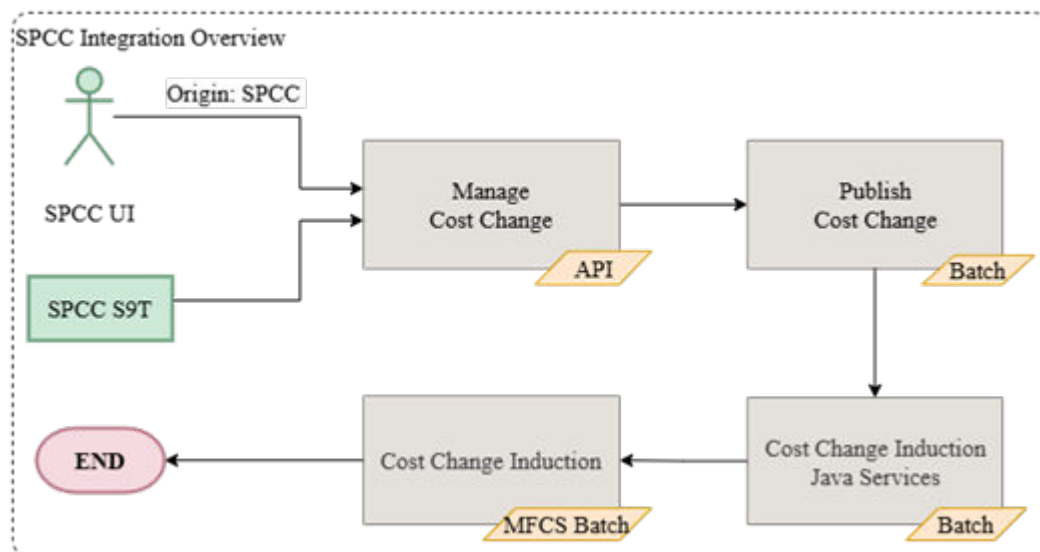
Cost Change Status:

- **New** – New status is the initial status of a Cost Change and the prior status to being approved, if a user has approval permissions and they submit a Cost Change in New Status the Cost Change is automatically approved.
- **Approved** – Approved status is Cost Changes that have been recorded on SPCC and are waiting to be exported into MFCS at a later date, while a Cost Change is in Approved Status it can be updated or rejected. If a user with approval permissions updates an Approved Cost Change the status remains as Approved.

- **Rejected** – The Rejected Status is for Cost Changes that were either submitted or approved previously and a user rejected them. Rejected Cost Changes can be updated and submitted again and changed into New Status.
- **Changes Proposed** – Cost Changes with Changes Proposed status were previously approved and were updated by a user without Approve permissions, in this status a user with Approve permissions can make further updates, and approve, or reject the changes made. If the user rejects the changes the Cost Change goes back to Approved Status with the previously approved data.
- **Extracted** – Extracted Status is displayed when a previously approved Cost Change was sent to MFCS and can no longer be updated.
- **Error** – Error Status is used when an issue is encountered while extracting the Cost Change into MFCS and when in this status technical support will be required to understand what caused the error.

Integration High Level

Figure 17-2 Cost Changes Management Integration Overview



The integration of Cost Changes from Cost Changes Management to MFCS described above is done in the following order:

- Cost Changes are created and managed in Cost Changes Management, either through the UI or via upload functionality.
- After being approved, and when the publish date is reached, Cost Changes are published by a batch process.
- Cost Change Induction Java Services adds Cost Change to an MFCS induction file, which is transferred to MFCS SFTP/FTS.
- Finally, MFCS induction batch for Cost Changes is executed and integrates Cost Changes to MFCS.

Configuration

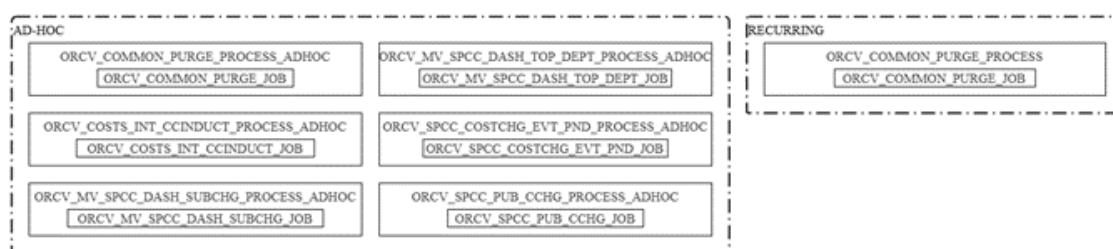
This section describes configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Schedule Overview

Figure 17-3 Cost Changes Management Batch Schedule Overview



Other Cloud Services Configuration

This section describes implementation details that must be reviewed from Oracle Retail Merchandising System perspective.

Cost Change ids used in Cost Changes Management module should not overlap with Cost Change ids created through other modes in MFCS such as Induction, Rest Services or MFCS UI.

System Options

The following MFCS System Option is used by the publish cost change batch to determine when to publish Cost Changes from Cost Changes Management to MFCS.

TABLE	PARAMETER	DESCRIPTION
FOUNDATION_UNIT_OPTIONS	COST_PRIOR_CREATE_DAYS	Indicates the minimum number of days prior to the cost change effective date that a cost change can be entered.

Reason Codes

Cost Changes Reason Codes available in Cost Changes Management are the reason codes available in MFCS table COST_CHG_REASON.

Other Implementation Considerations

This section describes Cost Changes Management implementation considerations.

Notifications and Alerts

Cost Changes Management uses the Events Framework to inform users about some actions with notifications and alerts. Below is the list of notifications available. The notifications are implemented via the Rule Engine Framework documented in the [Supply Chain Hub Frameworks](#) section.

Notifications

SPCC_EVENT_PEND_APPR_COSTCHG

Submitted Cost Change is pending approval.

Property	Value
Event Name	SPCC_EVENT_PEND_APPR_COSTCHG
Event Type	ORCV_SPCC_EVENT_PEND_APPR_COSTCHG
Description	Notification triggered when: • New Cost Change is Submitted by a User without Approval Permissions • New Cost Change is Submitted by Supplier and Cost Change has a Significant Change • Approved Cost Change is Updated by a User without Approval Permissions
Trigger	UI/S9T
Target Users	User has access to Application • User has Approval Permission • User has access to Supplier • User has access to at least 1 Item of the Cost Change

Alerts

SPAP_EVENT_APPTS_PENDING_ACTION

List of future Appointments pending approval.

Property	Value
Event Name	SPCC_EVENT_COSTCHGS_PENDING
Event Type	ORCV_SPCC_EVENT_COSTCHGS_PENDING
Alert Type	EMAIL
Description	Email with the list of cost changes pending approval.
Trigger	Batch ORCV_SPCC_COSTCHG_EVT_PND_JOB
Target Users	• User has access to Application • User has Approval Permission • User has access to Supplier • User has access to at least 1 Item of the Cost Change

Specific Batch Configuration

This section describes specific batch configurations.

Cost Changes Events Pending Batch

Batch job to check if there are Cost Changes pending user actions and creates Alerts.

- Job: ORCV_RLEG_SPCC_COSTCHG_EVT_PND
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_RLEG_SPCC_COSTCHG_EVT_PND
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Publish Cost Change Batch

Publish Cost Change batch publishes Cost Changes from Cost Changes Management to the integration layer when the publish date is reached. It uses the System Option COST_PRIOR_CREATE_DAYS from MFCS Foundation Unit Option to determine how many days before the effective date they should be published. Approved Cost Changes are considered and Cost Changes that were previously Approved and are in Changes Proposed are considered in their previous Approved version. The Cost Changes Integration API is called with the objects containing the cost changes to be integrated.

- Job: ORCV_SPCC_PUB_CCHG_JOB
- Frequency: Nightly
- Dependency: None
- Inter Scheduler Dependency: None

Batch configuration table: ORCV_BATCH_CONFIG

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_SPCC_PUB_CCHG_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3

COLUMN NAME	DEFAULT VALUES
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Materialized Views

This application requires some results to be materialized in tables using the Materialized View Framework described in the [Supply Chain Hub Frameworks](#) section, below is the list for this application. Materialized views used in Cost Change Management are listed in this section.

ORCV_MV_SPCC_DASH_TOP_DEPT

Number of cost changes items by department displayed in Cost Changes Management Home page.

- **Pre-Dependencies:** N/A
- **Post-Dependencies:** N/A
- **Frequency:** Ad-hoc.

ORCV_MV_SPCC_DASH_SUBCHG

Number of cost changes that have status substantial change displayed in Cost Changes Management Home page.

- **Pre-Dependencies:** N/A
- **Post-Dependencies:** N/A
- **Frequency:** Ad-hoc.

Codes

Certain operations require codes to define the options available. Below you can find the list of code types available. The codes are defined in tables ORCV_CODE_HEAD and ORCV_CODE_DETAIL.

CODE TYPE	CODE TYPE DESCRIPTION
CCST	Cost Change status

Custom Validations

The module performs validations by implementing Custom Validation Groups of the Custom Validations Framework described in the [Supply Chain Hub Frameworks](#) section. The groups listed below can be used to add Custom Validations for Cost Changes.

Group Id	Executed When
SPCC_COSTCHG_APPROVE	Cost Change is approved or Approved Cost Change is updated.
SPCC_COSTCHG_REJECT	Cost changes is rejected.

Group Id	Executed When
SPCC_COSTCHG_SUBMIT	Cost Change is submitted or when updating an existing Cost Change.

Templates

Cost Changes Management uses a spreadsheet download and upload process, developed as part of the Retail Supply Chain Hub Frameworks, for creation, maintenance, and extraction of data. To support those functions, several different configurations should be reviewed as part of the implementation.

Template SPCC Cost Change

Table 17-1 Template Key: SPCC_COST_CHANGE_DATA

ACTION TYPE	ROLE NAME
Download	ORCV_SPCC_COSTCHG_APPR_JOB
Download	ORCV_SPCC_COSTCHG_EDIT_JOB
Download	ORCV_SPCC_COSTCHG_VIEW_JOB
Download Blank	ORCV_SPCC_APP_JOB
Upload	ORCV_SPCC_COSTCHG_APPR_JOB
Upload	ORCV_SPCC_COSTCHG_EDIT_JOB

Table 17-2 Worksheet Cost Changes Head

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Cost change id, supplier, origin country, effective date, cost change description, reason code	Creates a new Cost Change. When the user has Edit permissions, it creates the Cost Change with status New. When the user has Approve permissions, it creates the Cost Change with status Approved.
Update	Cost change id Updatable: effective date, cost change description	Updates the Cost Change with the new details. When the upload is done by a user with Edit permissions, if the previous status was New or Reject, the Cost Change status will be New. If the previous status was Approved or Change Proposed, the status will be Change Proposed. When the upload is done by a user with Approve permissions, the Cost Change status will be Approved.
Approve, Reject	Cost change id	Approves or Rejects the Cost Change.

Table 17-3 Worksheet Cost Changes Detail

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Cost change id, Item or VPN, New unit cost	Adds items to the Cost Change. When the Cost Change is not present in Cost Changes Head worksheet it is considered an update of the Cost Change.
Update	Cost change id Updatable: new unit cost, recalculate order indicator	Updates the item details of the Cost Change.

Interactive Grid Additional Columns

This application supports adding additional fields to the Interactive Grids by using the Interactive Grid Additional Columns functionality described in the [Supply Chain Hub Frameworks](#) section.

The Interactive Grids that support this functionality are listed below:

Interactive Grid	Grid ID	User Types	Page	Non Based View	Entity Primary Key
Cost Changes Dashboard Grid	P100_IG_COST CHG	Retail Administrator, Retailer User, Supplier Administrator, Supplier User	Home	ORCV_V_SPCC_COSTCHG_HE AD_NB	COST_CHANG E_ID
Cost Change Items Grid	P200_IG_ITEM S	Retail Administrator, Retailer User, Supplier Administrator, Supplier User	Create or Edit Cost Change	ORCV_V_SPCC_ITEM_NB	ITEM, SUPPLIER, COUNTRY

References

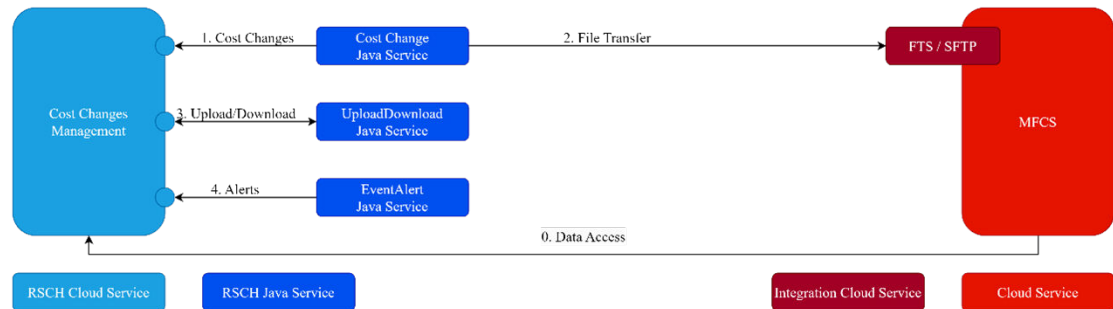
Cost Changes Management also uses the Queries module to enable communication between retailer and supplier users, details about Queries are available in the [Supply Chain Hub Frameworks](#) section.

Operations

This section describes Cost Changes Management operations.

Integration by Module

Figure 17-4 Cost Changes Management Integration by Module



STEP	DESCRIPTION
0	RSCH Data Access of MFCs data
1	Cost Change Java Service checks if there are Cost Changes from Cost Changes Management to be published to MFCs and generates ODS files for MFCs Induction.
2	Cost Change Java Service sends generated files to Integration File Transfer Java Service to send to MFCs
3	Integration File Transfer publishes files to MFCs File Transfer Services (FTS) or MFCs shared folder (depending on configuration).
4	Cost Changes Management calls Upload Download Java Service to transform files, depending on the request details, Upload Download Java Service might need to callback to Cost Changes Management to get more details.
5	Event Alert Java Service checks if there are alerts to be triggered.

Batch by Module

This section describes batches by module.

Cost Changes Events Pending Batch

The Cost Changes Events Pending Batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section. To execute this batch call Rule Engine Web Service with Rule Key SPCC_COSTCHG_EVT_PND.

Publish Cost Changes

Authentication Method: OAuth2

BaseURL: `https://<BASE_URL>/ords/mfcs/orcv/merch/costchangesmanagement/publish/costchange/`

`batch/<URL_PATH>`

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_SPCC_PUB_CCHG_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPCC_COSTCHG_HEAD ORCV_SPCC_COSTCHG_DTL ORCV_COSTS_INT_HEAD_SVC ORCV_COSTS_INT_DETAIL_SVC ORCV_COSTS_INT_DETAIL_LOC_SVC ORCV_COSTS_INT_ERR_SVC
/restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "<EXEC_ID>", "jobName": "ORCV_SPCC_PUB_CCHG_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPCC_COSTCHG_HEAD ORCV_SPCC_COSTCHG_DTL ORCV_COSTS_INT_HEAD_SVC ORCV_COSTS_INT_DETAIL_SVC ORCV_COSTS_INT_DETAIL_LOC_SVC ORCV_COSTS_INT_ERR_SVC
/log	GET	URL query parameters: /log?jobName=ORCV_SPCC_PUB_CCHG_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A
/kill	POST	Empty	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
/status	GET	URL query parameters: /status? jobName=ORCV_SPCC_PUB_CCHG_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A
/ping	GET	Empty	N/A

Materialized Views Refresh Details

This section describes Materialized Views Refresh Details.

ORCV_MV_SPCC_DASH_SUBCHG

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPCC_DASH_SUBCHG

COLUMN	DESCRIPTION	VALUE
Result Table A	Name of _A result table.	ORCV_SPCC_DASH_SUBCHG_A_R
Result Table B	Name of _B result table.	ORCV_SPCC_DASH_SUBCHG_B_R
Source View	The View contains business logic to be used to populate the result table.	ORCV_V_SPCC_DASH_SUBCHG_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPCC_DASH_SUBCHG_RC
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH . COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	SUPPLIER

ORCV_MV_SPCC_DASH_TOP_DEPT

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPCC_DASH_TOP_DEPT
Result Table A	Name of _A result table.	ORCV_SPCC_DASH_TOP_DEPT_A_R
Result Table B	Name of _B result table.	ORCV_SPCC_DASH_TOP_DEPT_B_R
Source View	The View contains business logic to be used to populate the result table.	ORCV_V_SPCC_DASH_TOP_DEPT_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPCC_DASH_TOP_DEPT_RC
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH . COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4

COLUMN	DESCRIPTION	VALUE
Filter Fields	Column name of the view to be used for threading.	DEPT

Supplier Performance

This section describes the common functionality and configurations available for Supplier Performance.

Implementation Overview

This section provides an implementation overview.

Functionality

The Supplier Performance module is an application designed to monitor the relationship and performance of a Supplier.

The application was designed for Retailer users only and provides:

- Automatic calculation of major KPIs
- Maintain and record questionnaires for non-automatic KPIs

Configuration

This section describes Supplier Performance configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Schedule Overview

Figure 18-1 Supplier Performance Batch Schedule Overview



Other Implementation Considerations

This section contains Supplier Performance implementation considerations.

Specific Batch Configuration

This section describes specific batch configurations.

On Time Delivery

This batch will compare a Shipment Estimated Arrival Date (or PO Not after date if no Estimated Arrival Date exists) with the first receipt date of an ASN.

- Job: ORCV_SPPF_ON_TIME_DLV_JOB
- Frequency: Nightly
- Dependency: None
- Inter Scheduler Dependency: MFCS.SQLSTAGE_JOB

Batch configuration table: ORCV_BATCH_CONFIG

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_SPPF_ON_TIME_DLV_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Quantity Compliance

This batch will compare the Shipment Expected Quantity vs Received Quantity.

- Job: ORCV_SPPF_QTY_COMP_JOB
- Frequency: Nightly
- Dependency: None
- Inter Scheduler Dependency: None

Batch configuration table: ORCV_BATCH_CONFIG

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_SPPF_QTY_COMP_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Invoice Compliance

This batch will calculate the number of Merchandise Invoices with discrepancies vs the total amount of Merchandise Invoices from the Supplier.

- Job: ORCV_SPPF_INVC_COMP_JOB
- Frequency: Nightly
- Dependency: None
- Inter Scheduler Dependency: None

Batch configuration table: ORCV_SPPF_INVC_COMP_JOB

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_SPPF_INVC_COMP_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4

COLUMN NAME	DEFAULT VALUES
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Return Percentage

This batch will calculate the % of RTVs vs the total PO intake on the last SCORE_AVG_DAYS at the time the refresh is performed.

- Job: ORCV_SPPF_RETURN_PCT_JOB
- Frequency: Nightly
- Dependency: None
- Inter Scheduler Dependency: None

Batch configuration table: ORCV_BATCH_CONFIG

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_SPPF_RETURN_PCT_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Defect Percentage

This batch will calculate the % of Defects vs the total PO intake on the last SCORE_AVG_DAYS at the time the refresh is performed.

- Job: ORCV_SPPF_DEFECT_PCT_JOB
- Frequency: Nightly
- Dependency: None
- Inter Scheduler Dependency: None

Batch configuration table: ORCV_BATCH_CONFIG

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_SPPF_DEFECT_PCT_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3

COLUMN NAME	DEFAULT VALUES
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Overall Compliance

This batch will calculate the score for each Supplier of each metric along with an overall score of the last days based on SCORE_AVG_DAYS System Parameter.

- Job: ORCV_SPPF_OVERALL_COMP_JOB
- Frequency: Nightly
- Dependency: ORCV_SPPF_ON_TIME_DLV_JOB, ORCV_SPPF_QTY_COMP_JOB, ORCV_SPPF_INVC_COMP_JOB, ORCV_SPPF_RETURN_PCT_JOB, ORCV_SPPF_DEFECT_PCT_JOB
- Inter Scheduler Dependency: None

Batch configuration table: ORCV_BATCH_CONFIG

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_SPPF_OVERALL_COMP_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Materialized Views

This application requires some results to be materialized in tables using the Materialized View Framework described in the [Supply Chain Hub Frameworks](#) section, below is the list for this application. Materialized views used in Supplier Performance are listed in this section.

ORCV_MV_SPPF_SUP_T_ORD

Returns all Total count, value and last creation date of completed Orders with status C and Total count, value and last creation date of open Orders status A, by supplier. Value currency is on MFCS system configuration options reference currency. Used by the Supplier Performance homepage.

Pre-Dependencies: ORCV_SPPF_OVERALL_COMP_JOB

Post-Dependencies: ORCV_MV_SPPF_SUP_ORDERS_JOB

Frequency: Ad-hoc, Nightly

ORCV_MV_SPPF_SUP_ORDERS

Returns the order number, status, supplier, currency code, received and not after date, total value, quantity ordered and received for all supplier orders. Used on Supplier Performance page.

Pre-Dependencies: ORCV_MV_SPPF_SUP_T_ORD_JOB

Post-Dependencies: None

Frequency: Ad-hod, Nightly

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
SPCL	Supplier Performance Classification, to be used on the dashboard screen.
SPMT	Supplier Performance Metrics, to be used on the dashboard screen.
SPRT	Supplier Performance Ratings.

Templates

Supplier Performance uses a spreadsheet download and upload process, developed as part of the Retail Supply Chain Hub Frameworks, for creation, maintenance, and extraction of data. To support these processes there are some templates which must be configured, and several configurations should be reviewed as part of the implementation.

Template SPPF Questionnaire

Table 18-1 Template key: ORCV_SPPF_QUESTN

ACTION TYPE	ROLE NAME
Download Blank	ORCV_SPPF_APP_JOB
Upload	ORCV_SPPF_SCORE_EDIT_JOB

Table 18-2 Worksheet Questionnaire - SPPF_QUESTN_HEADER

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Template Name Template ID	Create new questionnaire template. Template ID is a placeholder ID while creating the questionnaire template to be referenced by the Questionnaire Detail sheet.
Update	Template ID	Update an existing questionnaire template.
Delete	Template ID	Delete an existing questionnaire template.

Table 18-3 Worksheet Questionnaire Detail - SPPF_QUESTION Detail

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Template ID Mandatory Indicator Criteria ID Criteria Text	Create new questionnaire template criteria. When creating new criteria the Criteria ID is mandatory, to be used as placeholder ID and the template ID needs to keep the reference ID to the header sheet.
Update	Template ID Criteria Text Mandatory Indicator	Update existing template criteria.
Delete	Template ID Criteria Text	Delete criteria detail from existing questionnaire template.

Template SPPF Questionnaire Reply

Table 18-4 Template key: ORCV_SPPF_QUESTION_REPLY

ACTION TYPE	ROLE NAME
Download	ORCV_SPPF_APP_JOB
Upload	ORCV_SPPF_SCORE_EDIT_JOB

Table 18-5 Worksheet Questionnaire Reply - SPPF_QUESTION_REPLY

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Criteria ID Criteria Text Mandatory Indicator Supplier	Creates a new questionnaire reply for the supplier. An upload file can only have the reply for a single supplier.

Operations

This section describes Supplier Performance Operations.

Batch by Module

This section describes batches by module.

On Time Delivery

Authentication Method: OAuth2

BaseURL: https://<BASE_URL>/ords/mfcs/orcv/merch/supplierperformance/ontimedelivery/batch/<URL_PATH>

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_SPPF_ON_TIME_DLV_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPF_ON_TIME_DLV
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPF_ON_TIME_DLV_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPF_ON_TIME_DLV
log	GET	URL query parameters: /log? jobName=ORCV_SPPF_ON_TIME_DLV_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPF_ON_TIME_DLV_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
status	GET	URL query parameters: /status? jobName=ORCV_SPPF_ON_TIME_DLV_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A

Quantity Compliance

Authentication Method: OAuth2

BaseURL: https://<BASE_URL>/ords/mfcs/orcv/merch/supplierperformance/quantitycompliance/batch/<URL_PATH>

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_SPPF_QTY_COMP_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPF_QTY_COMP
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPF_QTY_COMP_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPF_QTY_COMP
log	GET	URL query parameters: /log? jobName=ORCV_SPPF_QTY_COMP_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPF_QTY_COMP_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
status	GET	URL query parameters: /status? jobName=ORCV_SPPF_QTY_COMP_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A

Invoice Compliance

Authentication Method: OAuth2

BaseURL: https://<BASE_URL>/ords/mfcs/orcv/merch/supplierperformance/
invoicecompliance/batch/<URL_PATH>

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_SPPF_INVC_COMP_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPF_INVC_COMP
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPF_INVC_COMP_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPF_INVC_COMP
log	GET	URL query parameters: /log? jobName=ORCV_SPPF_INVC_COMP_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPF_INVC_COMP_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
status	GET	URL query parameters: /status? jobName=ORCV_SPPF_INVC_COMP_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A

Return Percentage

Authentication Method: OAuth2

BaseURL: https://<BASE_URL>/ords/mfcs/orcv/merch/supplierperformance/returnpercentage/batch/<URL_PATH>

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_SPPF_RETURN_PCT_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPF_RETURN_PCT

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPF_RETURN_PCT_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPF_RETURN_PCT
log	GET	URL query parameters: /log? jobName=ORCV_SPPF_RETURN_PCT_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPF_RETURN_PCT_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
status	GET	URL query parameters: /status? jobName=ORCV_SPPF_RETURN_PCT_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A

Defect Percentage

Authentication Method: OAuth2

BaseURL: https://<BASE_URL>/ords/mfcs/orcv/merch/supplierperformance/defectpercentage/batch/<URL_PATH>

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_SPPF_DEFECT_PCT", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPF_RETURN_PCT
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPF_DEFECT_PCT", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPF_RETURN_PCT
log	GET	URL query parameters: /log?jobName=ORCV_SPPF_DEFECT_PCT &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPF_DEFECT_PCT", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
status	GET	URL query parameters: /status? jobName=ORCV_SPPF_DEFECT_PCT &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A

Overall Compliance

Authentication Method: OAuth2

BaseURL: https://<BASE_URL>/ords/mfcs/orcv/merch/supplierperformance/defectpercentage/batch/<URL_PATH>

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_SPPF_OVERALL_COMP_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPF_SUPP_SCORE_HISTORY ORCV_SPPF_SUPP_SCORE
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPF_OVERALL_COMP_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPF_SUPP_SCORE_HISTORY ORCV_SPPF_SUPP_SCORE
log	GET	URL query parameters: /log? jobName=ORCV_SPPF_OVERALL_COMP_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPF_OVERALL_COMP_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
status	GET	URL query parameters: /status? jobName=ORCV_SPPF_OVERALL_COMP_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A

Materialized Views Refresh Details

This section describes the Materialized Views Refresh Details.

ORCV_MV_SPPF_SUP_T_ORD

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPPF_SUP_T_ORD
Result Table A	Name of _A result table.	ORCV_SPPF_SUP_T_ORD_A_R
Result Table B	Name of _B result table.	ORCV_SPPF_SUP_T_ORD_B_R
Source View	The View contains business logic to be used to populate the result table.	ORCV_V_SPPF_SUP_T_ORD_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPPF_SUP_T_ORD_RC
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH. COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NULL
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	SUPPLIER

ORCV_MV_SPPF_SUP_ORDERS

COLUMN	DESCRIPTION	VALUE
Result Synonym	Name of the synonym pointing to the result table.	ORCV_MV_SPPF_SUP_ORDERS

COLUMN	DESCRIPTION	VALUE
Result Table A	Name of _A result table.	ORCV_SPPF_SUP_ORDERS_A_R
Result Table B	Name of _B result table.	ORCV_SPPF_SUP_ORDERS_B_R
Source View	The View contains business logic to be used to populate the result table.	ORCV_V_SPPF_SUP_ORDERS_R
Driving View	This field contains the view to distribute the refresh process.	ORCV_V_SPPF_SUP_ORDERS_RC
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_COMMON_MV_REFRESH. COMMON_PRE_PROCESSOR
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NULL
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	ORDER_NUMBER

19

Quality Control

This section describes the common functionality and configurations available for Quality Control.

Implementation Overview

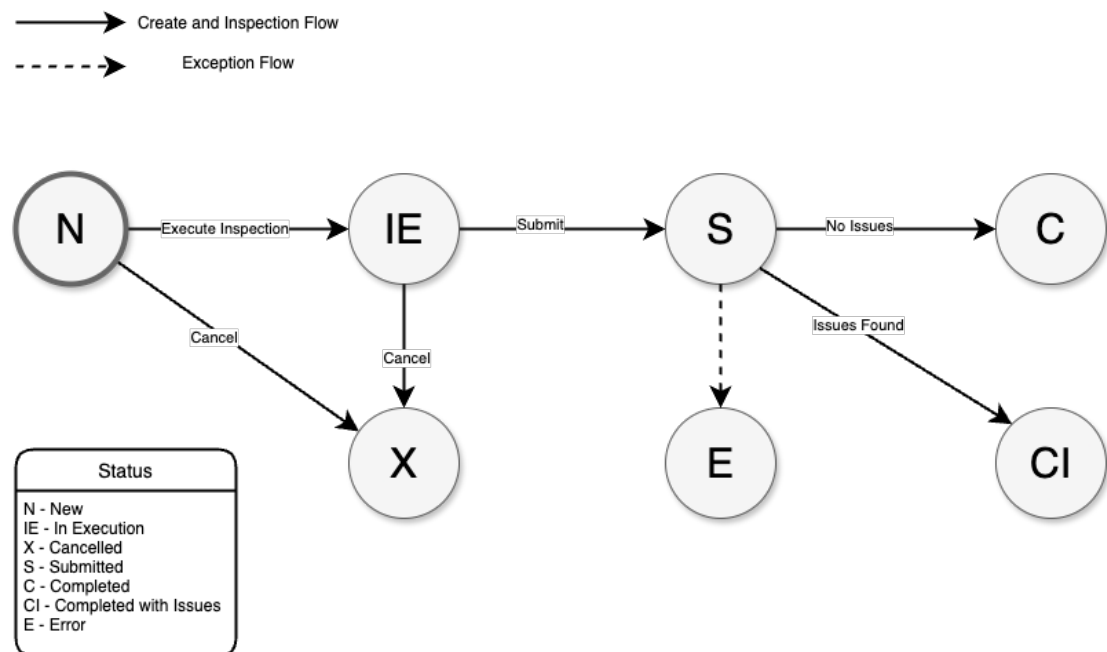
This section provides an implementation overview.

Functionality

Quality Control module gives the possibility to create and maintain Jobs for Quality Control checks (either by UI or by file export/import).

Status Workflow

Figure 19-1 Quality Control Status Workflow



Quality Control follows a workflow path, and the status can change depending on:

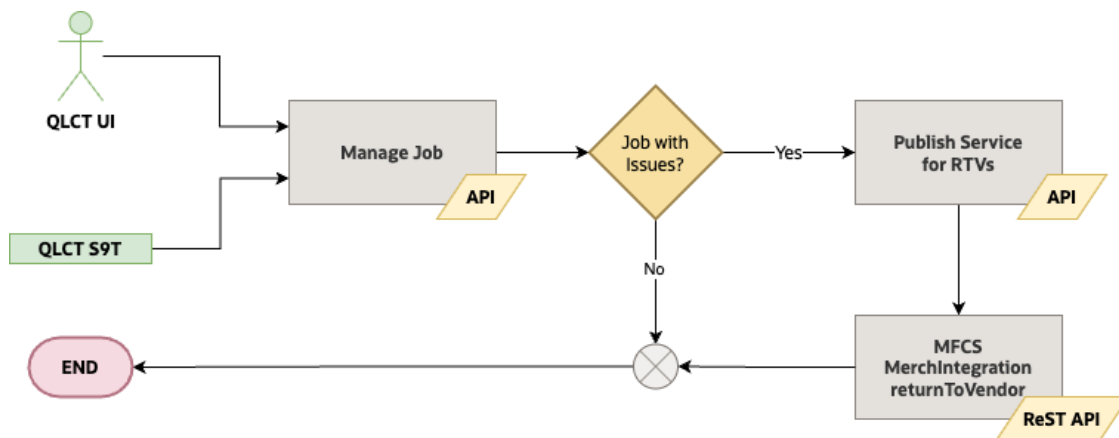
- The user permissions
- The current Status of the Job

Job Status:

- New (N) – New is the status of a Job when the user creates it. In this status a Job can be Cancelled or, when a user starts execution, can be set to In Execution.
- In Execution (IE) – In Execution status a Job can be updated and Submitted/Completed or can be Cancelled.
- Submitted (S) – Submitted is a status used when a user without approved permission tried to complete a Job. A Job can be set to this status when integration fails, to integrate RTV again.
- Completed (C) – A Completed Job is a Job completed without issues. The Job is not editable whilst on this status.
- Completed with Issues (CI) – A Completed with Issues Job is a Job with issues. The Job is not editable whilst on this status.
- Cancelled (X) – In Cancelled Status a Job cannot be updated.

Integration High Level

Figure 19-2 Quality Control Integration High Level



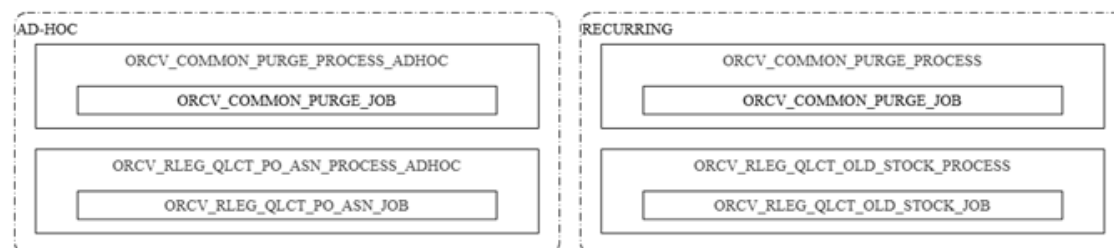
Configuration

This section describes Quality Control configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Figure 19-3 Quality Control Batch Schedule Overview



Other Implementation Considerations

This section describes Quality Control implementation considerations.

Notifications and Alerts

Quality Control uses the Events Framework to inform users about events occurred in scope of Quality Control with notifications and alerts. Below is the list of notifications available. The notifications are implemented via the Rule Engine Framework documented in the [Supply Chain Hub Frameworks](#) section.

Notifications

QLCT_EVENT_NEW_JOB_LOC

Job id %s1 was created.

Property	Value
Event Name	QLCT_EVENT_NEW_JOB_LOC
Event Type	ORCV_QLCT_EVENT_NEW_JOB_LOC
Description	Triggered for new Jobs being created.
Trigger	UI/S9T
Target Users	The retailer users that are assigned to the specified location will receive a notification.

QLCT_EVENT_COMP_JOB_SUP

Job id %s1 was completed with issues.

Property	Value
Event Name	QLCT_EVENT_COMP_JOB_SUP
Event Type	ORCV_QLCT_EVENT_COMP_JOB_SUP
Description	A Job was completed with issues.
Trigger	UI/S9T
Target Users	The supplier users with access to the supplier will receive a notification.

Alerts

QLCT_EVENT_NEW_JOB_EMAIL

Quality Control Job created.

Property	Value
Event Name	QLCT_EVENT_NEW_JOB_EMAIL
Event Type	ORCV_QLCT_EVENT_NEW_JOB_EMAIL
Alert Type	EMAIL
Description	A new Job was created.
Trigger	UI/S9T

Property	Value
Target Users	The retailer users that are assigned to the specified location will receive an e-mail.

Custom Validations

The application performs validations by implementing Custom Validation Groups of the Custom Validations Framework described in the [Supply Chain Hub Frameworks](#) section. The groups are listed below.

QLCT_JOB_CREATE

Triggered when the user creates a Job by UI or S9T.

All validation functions for this Custom Validation Group are available in ORCV_QLCT_JOB_VAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_SUPPLIER	Supplier exists and user has access.	1
VALIDATE_LOCATION	Location exists and user has access.	2
VALIDATE_BASED_ON	Entity type (PO / ASN / ADHOC) exists.	3
VALIDATE_ENTITY	Entity (PO / ASN) is valid.	4
VALIDATE_TEMPLATE	Template exists.	5
VALIDATE_ITEM_LOC	Item exists and item / location range is valid.	6

QLCT_JOB_MODIFY

Triggered when the user updates a Job detail by UI or S9T.

All validation functions for this Custom Validation Group are available in ORCV_QLCT_JOB_VAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_EXIST_JOB	Job exists.	1
VALIDATE_COUNTED_QTY	Counted quantity is ≥ 0 and is filled in whenever required.	2
VALIDATE_RESULT	Job / item result is valid. When NOK, a valid reason is provided.	3
VALIDATE_ISSUES_QTY	Issues quantity is ≥ 0 and $\leq$ expected quantity. When result is not "No action", it must be > 0 .	4

QLCT_JOB_SUBMIT

Triggered when the user submits a Job by UI or S9T.

All validation functions for this Custom Validation Group are available in ORCV_QLCT_JOB_VAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_EXIST_JOB	Job exists.	1

FUNCTION	VALIDATION	SEQ NO
VALIDATE_ALL_CHECKS	All checks were done.	2
VALIDATE_INSP_STATUS	Job inspection status and action are valid.	3

QLCT_JOB_CANCEL

Triggered when the user cancels a Job by UI or S9T.

All validation functions for this Custom Validation Group are available in ORCV_QLCT_JOB_VAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_EXIST_JOB	Job exists.	1
VALIDATE_INSP_STATUS	Job inspection status and action are valid.	2
VALIDATE_ITEMS_RELEASE D	Items are unreleased.	3

QLCT_JOB_COMPLETE

Triggered when the user completes a Job with or without issues by UI.

All validation functions for this Custom Validation Group are available in ORCV_QLCT_JOB_VAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_EXIST_JOB	Job exists.	1
VALIDATE_ALL_CHECKS	All checks were done.	2
VALIDATE_REASON_CODE	Reason code exists.	3
VALIDATE_SUPPLIER_RTV	Supplier allows RTVs.	4
VALIDATE_AUTH_CODE_RTV	Supplier requires authorization code for RTVs.	5
VALIDATE_INSP_STATUS	Job inspection status and action are valid.	6
VALIDATE_ITEM_RTYP_RCO D	Item / Reason Type / Reason Code are unique.	7

QLCT_JOB_RELEASE_ITEM

Triggered when the user releases one or more items by UI.

All validation functions for this Custom Validation Group are available in ORCV_QLCT_JOB_VAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_EXIST_JOB	Job exists.	1
VALIDATE_ALL_CHECKS	All checks were done.	2
VALIDATE_ITEMS_RELEASE D	Items are unreleased.	3
VALIDATE_INSP_STATUS	Job inspection status and action are valid.	4

Specific Batch Configuration

This section describes specific batch configurations.

PO ASN Received Batch

The PO ASN Received batch generates Quality Control jobs for received PO ASNs. For all PO ASNs that are in received status and were not yet previously processed, this batch calls the manage function of QLCT, providing the required information for the job creation. This batch uses the Rule Engine framework.

- Job: ORCV_RLEG_QLCT_PO_ASN_JOB
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

This batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section with the following configurations.

Rule default configuration (ORCV_RLEG_RULE):

RULE KEY	RULE NAME	PROGRAM_NAME	APP ALIAS	RULE MODE
QLCT_PO_ASN	PO ASN Received	ORCV_RLEG_QLCT_PO_ASN	QLCT	BATCH

Rule default parameters (ORCV_RLEG_PARAMETERS):

PARAMETER	DESCRIPTION	VALUE 1	VALUE 2	VALUE 3	VALUE 4
APPLY_TEMPLATE_TYPE	Indicates if it is to apply template Type PO/ASN in the Job Header.	Y	NA	NA	NA

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_RLEG_QLCT_PO_ASN
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Old Stock Batch

The Old Stock Batch generates Jobs for items without transactions for a period of time. It takes users the last transaction date, stock level and last inspection, this batch calls Manage Job API for each considered location, providing the required information for the job creation.

- Job: ORCV_RLEG_QLCT_OLD_STOCK_JOB
- Frequency: Daily
- Dependency: None
- Inter Scheduler Dependency: None

This batch uses Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section with the following configurations.

Rule default configuration (ORCV_RLEG_RULE):

RULE KEY	RULE NAME	PROGRAM_NAME	APP ALIAS	RULE MODE
QLCT_OLD_STOCK	Old Stock	ORCV_RLEG_QLCT_OLD_STOCK	QLCT	BATCH

Rule default parameters (ORCV_RLEG_PARAMETERS):

PARAMETER	DESCRIPTION	VALUE 1	VALUE 2	VALUE 3	VALUE 4
APPLY_ADHOC_TEMPLATE	Ad-Hoc template key to apply to job's items.	NULL	NA	NA	NA
DAYS_SINCE_LAST_TRANSACTION	Number of days since item's last transaction.	60	NA	NA	NA
SOH_THRESHOLD	Maximum SOH for inspection.	NULL	NA	NA	NA
DAYS_SINCE_LAST_CHECK	Number of days since item's last inspection.	30	NA	NA	NA
APPLY_TO_LOCATION	Location for which the rule should run.	NULL	NA	NA	NA

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_RLEG_QLCT_OLD_STOCK
MAX_CHUNK_SIZE	1
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
CKST	Check status
EXEC	Checks execute
INVA	Inventory adjustment reason
JCST	Job check status
JHET	Entity type (based on)
JHST	Job Status
JIRT	Job item/check result
JRRT	Job result type
NARE	No action reason
QCEL	QLCT error level. To be used in the error screen.
QCES	QLCT error source. To be used in the error screen.
REPR	Repair type
RLSI	Release indicator
RSRE	Result reason
THST	Template Status
THTY	Template Type

Templates

Quality Control uses a spreadsheet download and upload process, developed as part of the Retail Supply Chain Hub Frameworks, for creation, maintenance, and extraction of data. To support these processes there are some templates which must be configured, and several configurations should be reviewed as part of the implementation.

Template QLCT Job

Table 19-1 Template key: QLCT_JOB

ACTION TYPE	ROLE NAME
Download	ORCV_QLCT_QCJOB_CREATE_JOB
Download	ORCV_QLCT_QCJOB_VIEW_JOB
Download	ORCV_QLCT_QCINSPECT_APPR_JOB
Download	ORCV_QLCT_QCINSPECT_EDIT_JOB
Download	ORCV_QLCT_QCINSPECT_VIEW_JOB
Download Blank	ORCV_QLCT_APP_JOB
Upload	ORCV_QLCT_QCJOB_CREATE_JOB
Upload	ORCV_QLCT_QCINSPECT_APPR_JOB
Upload	ORCV_QLCT_QCINSPECT_EDIT_JOB

Table 19-2 Worksheet Job Header

ACTION	MANDATORY CELLS / FIELDS	NOTES
CREATE	Job ID Job Description Status Location Supplier Entity ID Entity Type Template	Creates a new Job with status New.
SUBMIT	Job ID	Submit Job.
CANCEL	Job ID	Cancel Job.

Table 19-3 Worksheet Job Detail

ACTION	MANDATORY CELLS / FIELDS	NOTES
CREATE	Job ID Check ID Item/VPN	Creates a new mapping between Job and Item.
UPDATE	Job ID Check ID Item/VPN Quantity Counted Quantity Issues Result Reason Comments	Updates Job and Items Checks.

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_S9T_CODE_HEAD and ORCV_S9T_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
QLJA	Quality Control – Job Data Action

Interactive Grid Additional Columns

This application supports adding additional fields to the Interactive Grids by using the Interactive Grid Additional Columns functionality described in the [Supply Chain Hub Frameworks](#) section.

The Interactive Grids that support this functionality are listed below:

Interactive Grid	Grid ID	User Types	Page	Non Based View	Entity Primary Key
Jobs Dashboard Grid	P100_IG_JOBS	Retail Administrator, Retailer User, Supplier Administrator, Supplier User	Home	ORCV_V_QLCT - JOB_NB	JOB_ID
Job Items Grid	P200_IG_ITEM S_ TO_INSPECT	Retail Administrator, Retailer User, Supplier Administrator, Supplier User	Create or Edit Cost Job	ORCV_V_QLCT - ITEM_NB	ITEM

References

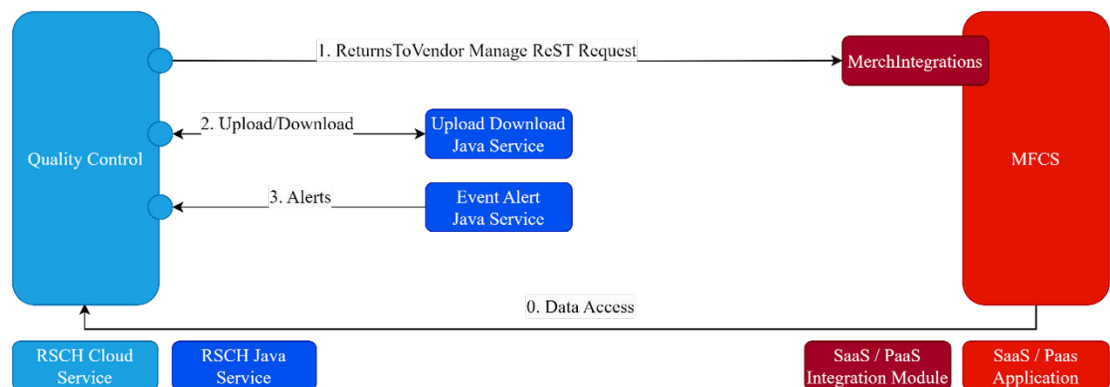
Quality Control also uses the Queries module to enable communication between retailer and supplier users, details about Queries are available in the [Supply Chain Hub Frameworks](#) section.

Operations

This section contains Quality Control operations.

Integration by Module

Figure 19-4 Quality Control Integration by Module



STEP	DESCRIPTION
0	RSCH Data Access of MFCS data
1	Quality Control calls MFCS Return To Vendor Rest endpoint in MerchIntegrations to publish new RTVs.

STEP	DESCRIPTION
2	Quality Control calls Upload Download Java Service to transform files, depending on the request details, Upload Download Java Service might need to callback to Quality Control to get more details.
3	Event Alert Java Service checks if there are alerts to be triggered.

Batch by Module

This section describes batches by module.

PO ASN Received Batch

The PO ASN Received batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section. To execute this batch call Rule Engine Web Service with Rule Key QLCT_PO_ASN.

Old Stock Batch

The Old Stock batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section. To execute this batch call Rule Engine Web Service with Rule Key QLCT_OLD_STOCK.

Deals Management

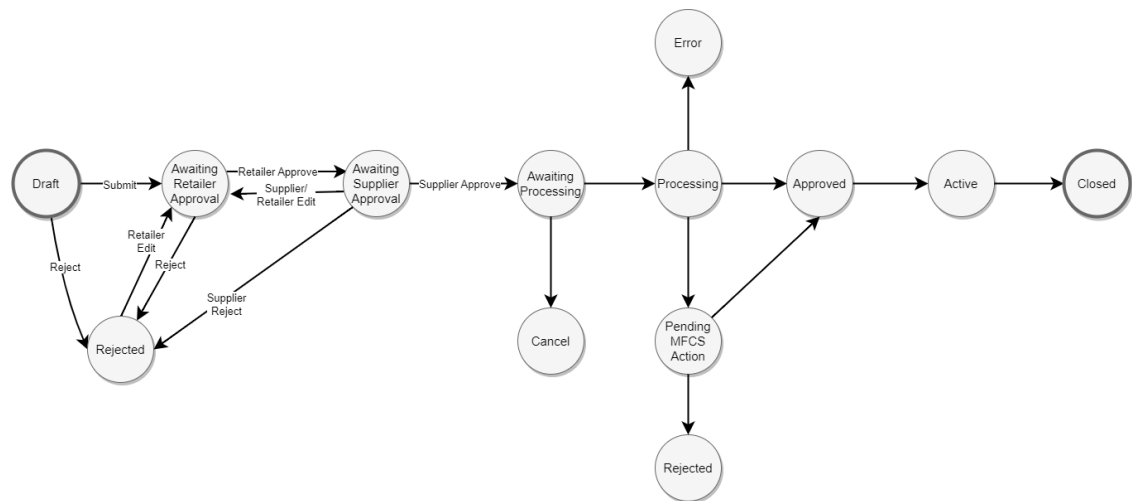
The Deals Management module is an application available in Supply Chain Hub. This chapter describes the implementation considerations and the operation details for this application.

Implementation Overview

Deals Management gives the possibility to create and maintain Deals and publish the Deals to MFCS before the start date.

Status Workflow

Figure 20-1 Deals Management Status Workflow



Deals Management follows a workflow path, and the status can change depending on:

- The user permissions
- The current status of the Deal

Deal Status

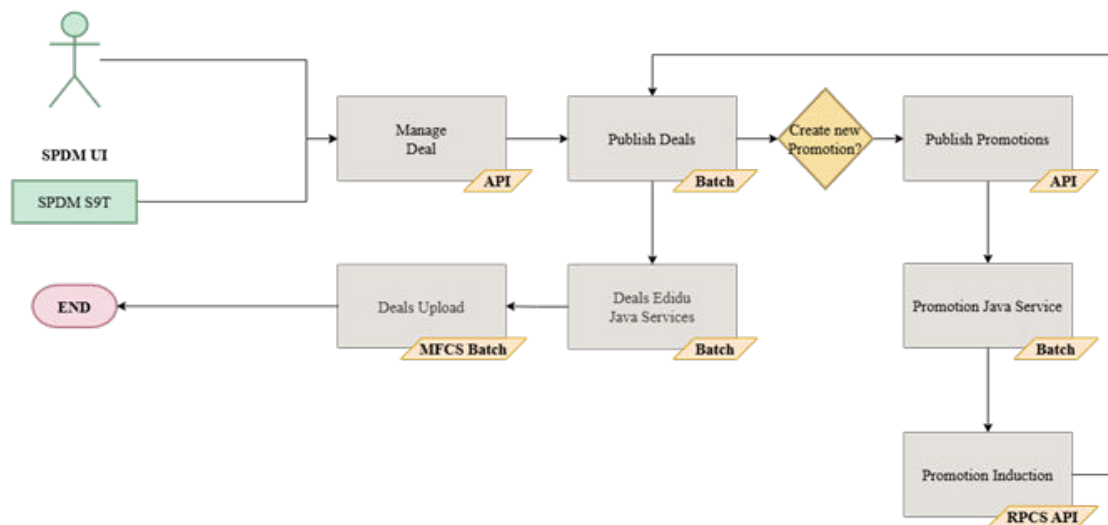
- **Draft** – Draft status is the initial status of a Deal and the prior status to being submitted. When the deal is submitted, the status is automatically updated to New. If the user rejects a Deal in Draft status, then new status is Rejected.
- **Awaiting Retailer Approval** – Awaiting Retailer Approval status is the prior status to being approved. If a Retailer user has approval permission and approve a Deal in Draft status the new status is Awaiting Retailer Approval (after running validations). If any user rejects the Deal, the new status is Rejected.
- **Awaiting Supplier Approval** – Awaiting Approved status is the new Deal status after the Deal has been approved by a Retailer user. While a Deal is in Awaiting Approval status it can be edited by a Supplier and Retailer users and the status goes back to New status,

can be approved by a Supplier user, with approval permission, and the new status is Awaiting Processing or rejected and the new status is Rejected.

- **Rejected** – The Rejected status is for Deals that were either created (Draft status), submitted (New status) or approved by a Retailer user (Awaiting Approval status) previously and a user rejected them. Deal in Reject status can be updated and submitted again by a Retailer user and changes to New status. If a Retailer user approves a rejected Deal the new status is Awaiting Approval.
- **Awaiting Processing** – Awaiting Processing status is the new Deal status after the Deal has been approved by a Supplier user when the Deal is in Awaiting Approval status. While a Deal is in Awaiting Processing status it can be processed by the Publish Batch Process or canceled.
- **Canceled** – Canceled status is the new Deal status after the Deal has been canceled when the Deal is in Awaiting Processing status. After the Deal is in Canceled status no more actions can be applied to the Deal.
- **Processing** – This status is used while the Deal is being interfaced with MFCS, once the Deal is successfully processed by MFCS the status will change to either Approved or Pending MFCS (if the deal in MFCS is in Worksheet or Submitted).
- **Active** – This status represents when the Deal is Active in MFCS. The deal will be displayed with Active Status for Approved Deals between the Start and End Date of the Deal.
- **Closed** – The deal will move to this status once it has reached the End Date and is Closed by MFCS.
- **Pending MFCS** – This status is used when a Deal was interfaced to MFCS but is not Approved. This could be due to validations failing in MFCS, but the Deal has still been created, or if the Supplier Configuration in MFCS is set to always interface the Deals to a non Approved Status. In this Status users are expected to go to MFCS to finish the approval workflow process.
- **Error** – This status is used to indicate if issues were found while interfacing the deal to MFCS and should contact support to check.

Integration High Level

Figure 20-2 Deals Management Integration High Level



Configuration

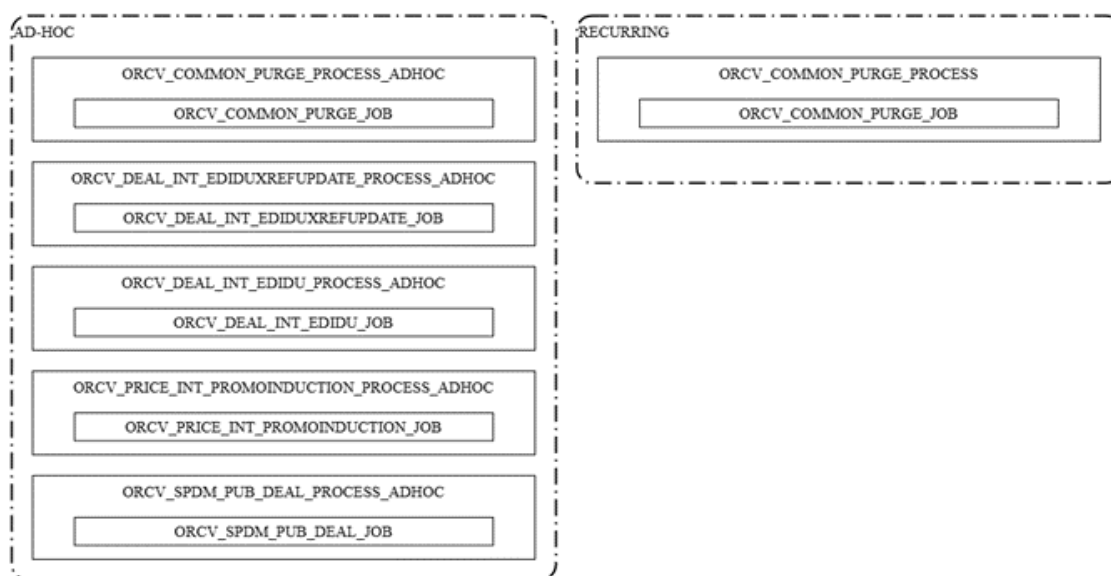
This section describes Deals Management configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Schedule Overview

Figure 20-3 Deals Management Batch Schedule Overview



Other Cloud Services Configuration

This section describes implementation details that must be reviewed from Oracle Retail Merchandising System perspective.

Deal ids used in Deals Management module should not overlap with Deal ids created through other modes in MFCS such as Induction, Rest Services or MFCS UI.

System Options

The following MFCS System Option are being used in Deals Management.

TABLE	PARAMETER	DESCRIPTION
FUNCTIONAL_CONFIG_OPTIONS	ORG_UNIT_IND	This parameter will be used to determine which locations will be available for Deals creation.
PROCUREMENT_UNIT_OPTIONS	DEAL_LEAD_DAYS	Number of days from today until deals can be integrated from Deals Management to MFCS.

The following MFCS System Options need to be reviewed to ensure Deals integration works properly.

TABLE	PARAMETER	DESCRIPTION
FUNCTIONAL_CONFIG_OPTIONS	DEAL_IND	This column controls whether deals will be used within the system. It needs to be 'Y'.
PROCUREMENT_UNIT_OPTIONS	CALC_NEGATIVE_INCOME	This field will determine whether income will be calculated or not when the income of the entire period is negative in deals. Recommendation is 'Y'.
PROCUREMENT_UNIT_OPTIONS	CREDIT_MEMO_LEVEL	This indicator decides at what level the credit/debit memos will be posted to MFCS. Recommendation is 'D' Deal level.
PROCUREMENT_UNIT_OPTIONS	DEAL_LEAD_DAYS	Number of days from today until deals can become active. For example, if this number is 2, deals can have a start date no earlier than 2 days from today's date. Recommendation at least 1.
PURGE_CONFIG_OPTIONS	DEAL_HISTORY_MONTHS	Indicates the number of months that closed deals will be retained in RMS. Recommended to be reviewed with Deals Management purge configurations.

Batches

The MFCS deals upload job, DEALUPLD_JOB, should be configured to execute with high frequency to assure that Deals are integrated into MFCS frequently.

Other Implementation Considerations

This section describes Deals Management implementation consideration.

Custom Validations

The module performs validations by implementing Custom Validation Groups of the Custom Validations Framework described in the [Supply Chain Hub Frameworks](#) section. The groups are listed below.

SPDM_DEAL_DRAFT

Triggered when user creates or saves a Deal.

All validation functions for this Custom Validation Group are available in ORCV_SPDM_CVAL_DEAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_SUPPLIER	Validate if It's a valid supplier.	1
VALIDATE_DEPARTMENT	Validate if it's a valid department.	2
VALIDATE_START_DATE	Validate if deal start date is valid.	3
VALIDATE_END_DATE	Validate if deal end date is valid.	4

FUNCTION	VALIDATION	SEQ NO
VALIDATE_BILLING_TYPE	Validate if billing type is valid against system parameter SUPPORTED_BILLING_TYPE.	5
VALIDATE_BILLBACK_METHOD	Validate if bill-back method is valid against code type DLBM.	6
VALIDATE_PURCHASE_SALES_BASED	Validate if purchase-sales indicator is valid against code type DMBO.	7
VALIDATE_BASED_ON	Validate if based on value is valid against code type DMBO.	8
VALIDATE_REPORTING_LEVEL	Validate if reporting level is valid against code type DMRL.	9
VALIDATE_MULTIPLE_THRSHLD_IND	Validate if multiple threshold indicator is valid for the billing type.	10
VALIDATE_DEFAULT_CONTRIBUTION	Validate default contribution type against code type DMCT.	11
VALIDATE_DEFAULT_DISCOUNT	Validate default contribution type against code type DMDI.	12
VALIDATE_STATUS	Validate if deal current status is valid to be Saved.	13
VALIDATE_ITEM_LOCATIONS	Validate if all items belong to the supplier of the deal and if the locations are valid.	14
VALIDATE_ITEM_LOCATIONS_EXCEPTION	Validate if Items Locations Exceptions are valid.	15
VALIDATE_PROMOTIONS	If is a VFP deal validate if there are promotions associated.	16
VALIDATE_THRESHOLDS	Validate threshold values are valid.	17

SPDM_DEAL_SUBMIT

Triggered when user submits a Deal for approval.

All validation functions for this Custom Validation Group are available in ORCV_SPDM_CVAL_DEAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_STATUS	Validate if deal current status is valid to be submitted	1
VALIDATE_ITEM_LOCATIONS	Validate if all items belong to the supplier of the deal and if the locations are valid.	2
VALIDATE_ITEM_LOCATIONS_EXCEPTION	Validate if Items Locations Exceptions are valid are valid to be submitted,	3
VALIDATE_PROMOTIONS	If is a VFP deal validate if there are promotions associated.	4
VALIDATE_THRESHOLDS	Validate threshold values are valid.	5
VALIDATE_START_DATE	Validate Deal start date.	6
VALIDATE_END_DATE	Validate Deal end data	7

SPDM_DEAL_APPROVE

Triggered when user approves a Deal.

All validation functions for this Custom Validation Group are available in ORCV_SPDM_CVAL_DEAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_SUPPLIER	Validate if It's a valid supplier.	1
VALIDATE_DEPARTMENT	Validate if it's a valid department.	2
VALIDATE_START_DATE	Validate if deal start date is valid.	3
VALIDATE_END_DATE	Validate if deal end date is valid.	4
VALIDATE_BILLING_TYPE	Validate if billing type is valid against system parameter SUPPORTED_BILLING_TYPE.	5
VALIDATE_BILLBACK_METHOD	Validate if bill-back method is valid against code type DLBM.	6
VALIDATE_PURCHASE_SALES_BASED	Validate if purchase-sales indicator is valid against code type DMBO.	7
VALIDATE_BASED_ON	Validate if based on value is valid against code type DMBO.	8
VALIDATE_REPORTING_LEVEL	Validate if reporting level is valid against code type DMRL.	9
VALIDATE_MULTIPLE_THRSHLD_IND	Validate if multiple threshold indicator is valid for the billing type.	10
VALIDATE_DEFAULT_CONTRIBUTION	Validate default contribution type against code type DMCT.	11
VALIDATE_DEFAULT_DISCOUNT	Validate default contribution type against code type DMDI.	12
VALIDATE_STATUS	Validate if deal current status is valid to be Saved.	13
VALIDATE_ITEM_LOCATIONS_EXCEP	Validate if all items belong to the supplier of the deal and if the locations are valid.	14
VALIDATE_ITEM_LOCATIONS	Validate if Items Locations Exceptions are valid.	15
VALIDATE_PROMOTIONS	If is a VFP deal validate if there are promotions associated.	16
VALIDATE_THRESHOLDS	Validate threshold values are valid.	17

SPDM_DEAL_REJECT

Triggered when user rejects a Deal.

All validation functions for this Custom Validation Group are available in ORCV_SPDM_CVAL_DEAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_STATUS	Validate if deal current status is valid to be submitted	1

SPDM_DEAL_CANCEL

Triggered when user cancels a Deal.

All validation functions for this Custom Validation Group are available in ORCV_SPDM_CVAL_DEAL package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_STATUS	Validate if deal current status is valid to be submitted	1

Specific Batch Configuration

This section describes specific batch configurations.

Publish Deal Batch

Publish Deals Batch publishes Deals in status “Awaiting Processing” to Integration API when the VDATE is between start date - INT_DAYS_BEFORE_START and start date.

If the Deal is of type VFP the batch will check if Promotions are required to be created before the deal and will publish to the Integration API the Promotions. Only after Promotions are successfully create in MFCS/RPCS it will publish the Deal.

The Bill Back Deals use BILL_BACK_PERIOD defined at Supplier configuration level when publishing the Deals to MFCS.

- Job: ORCV_SPDM_PUB_DEAL_JOB
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_SPDM_PUB_DEAL_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Codes

Certain operations require codes to define the options available. Below you can find the list of code types available. The codes are defined in tables ORCV_CODE_HEAD and ORCV_CODE_DETAIL.

CODE TYPE	CODE TYPE DESCRIPTION
DMBO	Threshold Based On
DMBP	Bill Back Period
DMCT	Contribution Types
DMDI	Discount Type
DMEL	SPDM error level. To be used in the error screen.

CODE TYPE	CODE TYPE DESCRIPTION
DMES	SPDM error source. To be used in the error screen.
DMIR	Add Item List - Retailer
DMIS	Add Item List - Supplier
DMLI	Item Types
DMLL	Location Types
DMLR	Add Location - Retailer
DMLS	Add Location - Supplier
DMLT	Threshold Limit Types
DMRC	Reason Codes
DMRL	Deal Reporting Level
DMSR	Add Location for VFP deals - Retailer
DMSS	Add Location for VFP deals - Supplier
DMST	Deal status

Templates

Deals Management uses a spreadsheet download and upload process, developed as part of the Retail Supply Chain Frameworks, for creation, maintenance, and extraction of data. To support these processes there are some templates which must be configured, and several configurations should be reviewed as part of the implementation.

Template Deal

Template key: SPDM_DEAL

The required permissions per action for this template are the following:

Table 20-1 Template key: SPDM_DEAL

ACTION TYPE	ROLE NAME
Download	ORCV_SPDM_DEAL_APPR_JOB
Download	ORCV_SPDM_DEAL_EDIT_JOB
Download	ORCV_SPDM_DEAL_VIEW_JOB
Download Blank	ORCV_SPDM_APP_JOB
Upload	ORCV_SPDM_DEAL_APPR_JOB
Upload	ORCV_SPDM_DEAL_EDIT_JOB

Table 20-2 Worksheet Deal

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Deal, Supplier, Start Date, End Date, Billing Type, Bill Back Method	Creates a new Deal with status Draft/New depending on Status column.
Modify	Deal, Supplier, Start Date, End Date, Billing Type, Bill Back Method	Modifies an existing Deal on Status Draft/New. If the user type is Retailer and the Status Deal is "Awaiting Approval" the Deal returns to "New" Status.

Table 20-2 (Cont.) Worksheet Deal

ACTION	MANDATORY CELLS / FIELDS	NOTES
Submit	Deal	Submit Deal, it means that Deal Status goes to "Submitted" Status.
Approve	Deal	Approve the Deal. If the user type is Retailer the Deal Status goes to "Awaiting Approval". If the user type is Supplier the Deal Status goes to "Awaiting Processing".
Reject	Deal	Rejects the Deal.
Cancel	Deal	Cancels the Deal.

Table 20-3 Worksheet Item Location

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Deal, Item type, Department	Creates a new itemloc for the department.
Create	Deal, Item type, Department, Class	Creates a new itemloc for the Class.
Create	Deal, Item type, Department, Class, Subclass	Creates a new itemloc for the Subclass.
Create	Deal, Item type, Department, Class, Subclass, Item	Creates a new itemloc for the Item.
Create	Deal, Item type, Department, Class, Subclass VPN	Creates a new itemloc for the corresponding VPN item.
Create	Deal, Location type, Item list	Creates a new itemloc for the Item List.
Create	Deal, Location type, Location	Creates a new itemloc for the Location.
Create	Deal, Location type, Location list	Creates a new itemloc for the Location List.
Create	Deal, Location type, Price zone	Creates a new itemloc for the Price Zone.
Delete	Deal, Item Loc id	Deletes the itemloc.

Table 20-4 Worksheet Exclusions

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Deal, Item Loc id	Creates a new exclusion.
Create	Department	Creates a new exclusion for the Department.
Create	Department, Class	Creates a new exclusion for the Class.
Create	Department, Class, Subclass	Creates a new exclusion for the Subclass.

Table 20-4 (Cont.) Worksheet Exclusions

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Location Type Location	Creates a new exclusion for the Location.
Delete	Deal, Item Loc id, Department	Deletes the Department exclusion.
Delete	Deal, Item Loc id, Department, Class	Deletes the Class exclusion.
Delete	Deal, Item Loc id, Department, Class, Subclass	Deletes the Subclass exclusion.
Delete	Deal, Item Loc id, Department, Class, Subclass, Item	Deletes the Item exclusion.
Delete	Location Type, Location	Deletes the Location exclusion.

Table 20-5 Worksheet Promotions

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Deal, Promotion id, Offer id	Creates a new promotion.
Delete	Deal, Promotion id, Offer id	Deletes an existing promotion.

Table 20-6 Worksheet Thresholds

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Deal, Lower Limit, Upper Limit, Value	Creates a new threshold.
Delete	Deal, Lower Limit, Upper Limit	Deletes an existing threshold.

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_S9T_CODE_HEAD and ORCV_S9T_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
DMHA	Deals Management - Header Action
DMLI	Deals Management - Item Type
DMLL	Deals Management - Location Type
DMST	Deals Management - Status

References

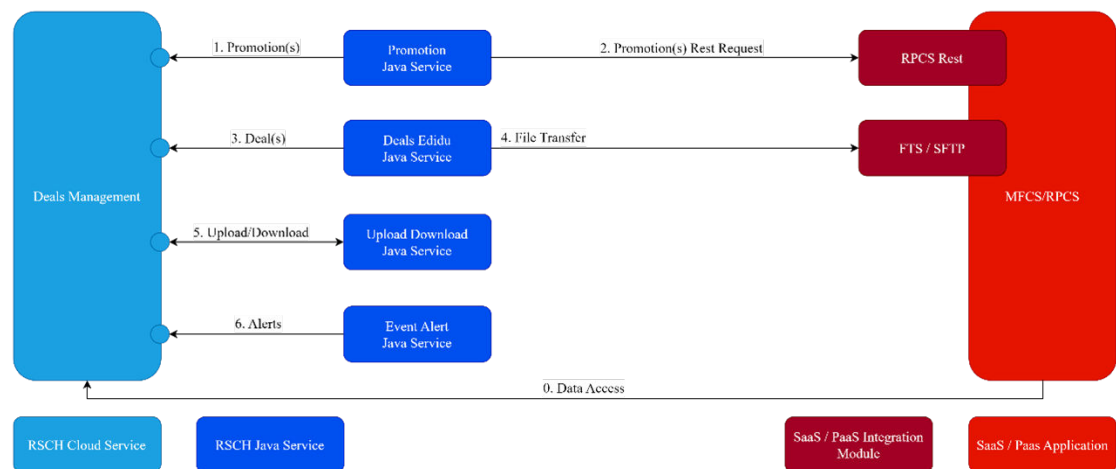
Deals Management also uses the Queries module to enable communication between retailer and supplier users, details about Queries are available in the [Supply Chain Hub Frameworks](#) section.

Operations

This section describes Deals Management operations.

Integration by Module

Figure 20-4 Deals Management Integration by Module



STEP	DESCRIPTION
0	RSCH Data Access of MFCS data
1	Promotion Java Service gets Promotions to be integrated to RPCS.
2	Promotion Java Service publishes Promotions to RPCS via Rest.
3	Deals Edidu Java Service gets Deals to be integrated to MFCS.
4	Deals Edidu Java Service sends produced files to Integration File Transfer.
5	Integration File Transfer publishes files to MFCS File Transfer Services (FTS) or MFCS shared folder (depending on configuration).
6	Deals Management calls Upload Download Java Service to transform files, depending on the request details, Upload Download Java Service might need to callback to Deals Management to get more details.
7	Event Alert Java Service checks if there are alerts to be triggered.

Batch by Module

This section describes batches by module.

Publish Deal Batch

Authentication Method: OAuth2

BaseURL: https://<BASE_URL>/ords/mfcs/orcv/merch/dealsmanagement/publish/deal/<URL_PATH>

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_SPDM_PUB_DEAL_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPDM_DEAL_HEAD ORCV_SPDM_DEAL_PROMO ORCV_DEAL_INT_ERR_SVC ORCV_DEAL_INT_HEAD_SVC ORCV_DEAL_INT_DETAIL_SVC ORCV_DEAL_IL_ITEM_SVC ORCV_DEAL_INT_COMP_PRM_SVC ORCV_DEAL_INT_THRESHOLD_SVC ORCV_PRICE_INT_ERR_SVC ORCV_PRICE_INT_PRM_HEAD_SVC ORCV_PRICE_INT_PRM_OF_SVC ORCV_PRICE_INT_PRM_OFR_SVC ORCV_PRICE_INT_PRM_OFR_ITM_SVC ORCV_PRICE_INT_PRM_OFLOC_SVC

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPDM_PUB_DEAL_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPDM_DEAL_HEAD ORCV_SPDM_DEAL_PROMO ORCV_DEAL_INT_ERR_SVC ORCV_DEAL_INT_HEAD_SVC ORCV_DEAL_INT_DETAIL_SVC ORCV_DEAL_IL_ITEM_SVC ORCV_DEAL_INT_COMP_PRM_SVC ORCV_DEAL_INT_THRESHOLD_SVC ORCV_PRICE_INT_ERR_SVC ORCV_PRICE_INT_PRM_HEAD_SVC ORCV_PRICE_INT_PRM_OF_SVC ORCV_PRICE_INT_PRM_OFR_SVC ORCV_PRICE_INT_PRM_OFR_ITM_SVC ORCV_PRICE_INT_PRM_OFLOC_SVC
log	GET	URL query parameters: /log? jobName=ORCV_SPDM_PUB_DEAL_JOB &executionId=<EXEC_ID>&logDate=YYY Y-DD-MM	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPDM_PUB_DEAL_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
status	GET	URL query parameters: /status? jobName=ORCV_SPDM_PUB_DEAL_JOB &executionId=<EXEC_ID>&logDate=YYY Y-DD-MM	N/A

21

Supplier Penalties

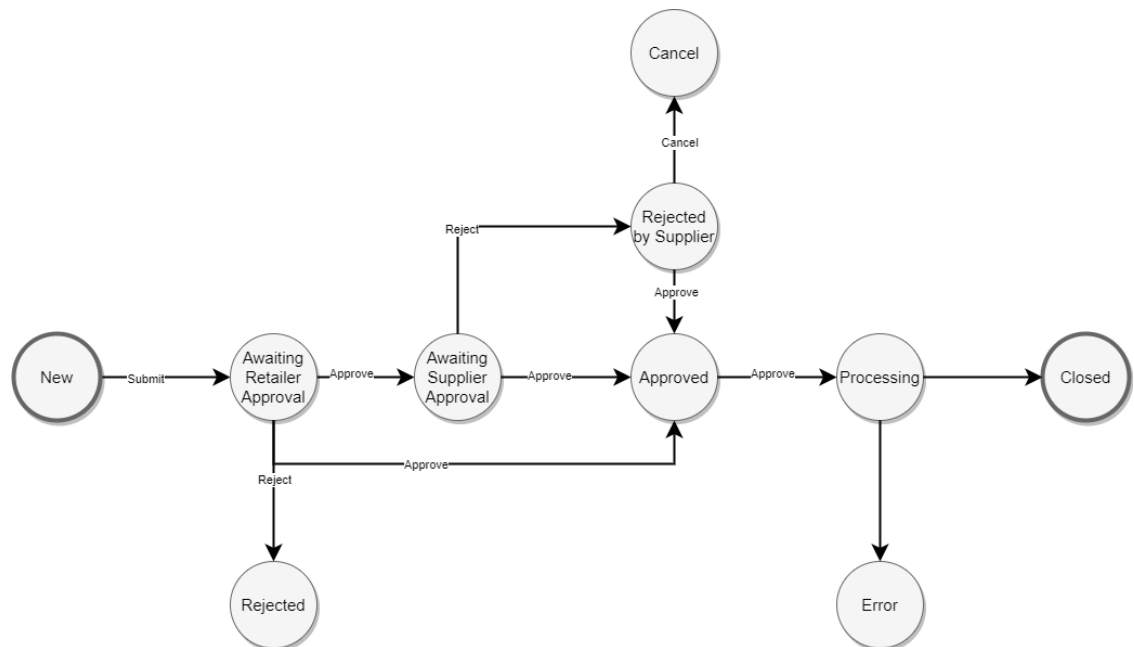
The Supplier Penalties module is an application available in Supply Chain Hub. This chapter describes the implementation considerations and the operation details for this application.

Implementation Overview

Supplier Penalties gives the possibility to automate the creation of penalties that will translate into a Debit Memo/Credit Note Request, generated to the supplier.

Status Workflow

Figure 21-1 Supplier Penalties Status Workflow



Supplier Penalties follows a workflow path, and the status can change depending on:

- The user permissions
- The current status of the Penalty

Penalty Status

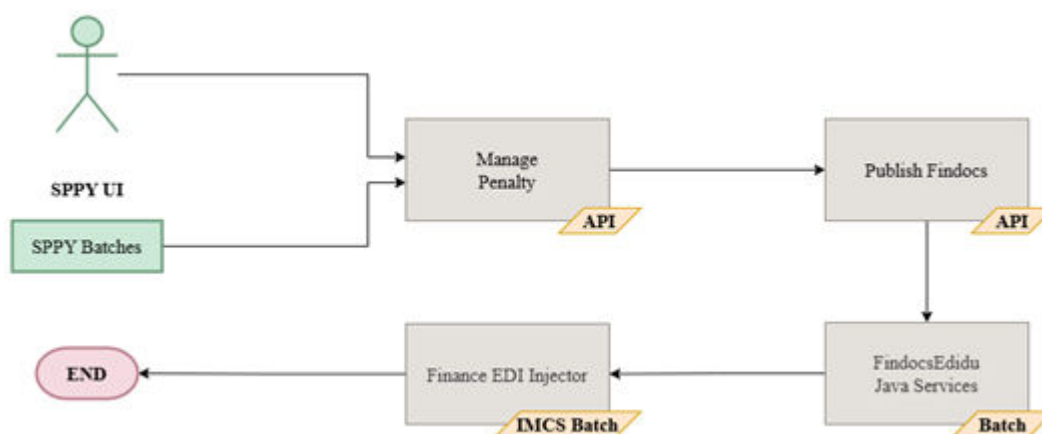
- New Status is a temporary status while the Penalty is being created, once submitted this Status will not be saved.
- Awaiting Retailer Approval – Awaiting Retailer Approval status is the new Penalty status after the Penalty has been created by a Retailer user or the Batch. While a Penalty is in Awaiting Approval status it can be edited by Retailer users and the status stays the same or can be approved by a Retailer user, with approval permission, and the new status is

Awaiting Supplier Approval or Approved if there is no need for Supplier Approval, or rejected and the new status is Rejected.

- Awaiting Supplier Approval – Awaiting Supplier Approval status is the Penalty status after the Penalty has been approved by a Retailer user. While a Penalty is in Awaiting Supplier Approval status it can be approved by a Supplier user, with approval permission, and the new status is Approved or rejected, and the new status is Rejected by Supplier.
- Approved – The Approved status is for Penalties that were approved by a Retailer user (Awaiting Retailer Approval status) previously or approved by a Supplier user (Awaiting Supplier Approval status).
- Rejected – The Rejected status is for Penalties that were either created (Awaiting Retailer Approval status) previously and a retailer user rejected them. After the Penalty is in Rejected status no more actions can be applied to the Penalty.
- Canceled – Canceled status is the new Penalty status after the Penalty has been Rejected by Supplier when the Penalty is in Awaiting Supplier Approval status and the Retailer accepts the rejection. After the Penalty is in Canceled status no more actions can be applied to the Penalty.
- Closed – The penalty will move to this status once it been integrated in MFCS.
- Error – This status is used to indicate if issues were found while interfacing the penalty to MFCS and should contact support to check.

Integration High Level

Figure 21-2 Supplier Penalties Integration High Level



Configuration

This describes Supplier Penalties configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Schedule Overview

Figure 21-3 Supplier Penalties Batch Schedule Overview



Other Cloud Services Configuration

This section describes cloud services configurations.

System Options

The following MFCS System Option are being used in Supplier Penalties.

TABLE	PARAMETER	DESCRIPTION
IM_SUPPLIER_OPTIONS	SEND_DEBIT_MEMO	This parameter will be used to determine which type of document should be generated in a penalty.
IM_SUPPLIER_OPTIONS	EXCL_PRFX_ON_EXTL_SRCD_DOCS	Indicates whether the system will include a prefix on the document identifier for externally generated documents. Before making any changes all pending integrations must be completed.

Batches

The IMCS integration EDI Injector job, REIM_EDI_INJECTOR_JOB, should be configured to execute with high frequency to assure that Finance Documents create in Supplier Penalties are integrated into IMCS as soon as possible.

Other Implementation Considerations

This section describes Supplier Penalties implementation considerations.

Notifications and Alerts

Supplier Penalties uses the Events Framework to inform users about events occurred in scope of Supplier Penalties with notifications and alerts. Below is the list of notifications available. The notifications are implemented via the Rule Engine Framework documented in the [Supply Chain Hub Frameworks](#) section.

Notifications

SPPY_EVENT_NEW_PENALTY

New Penalty created.

Property	Value
Event Name	SPPY_EVENT_NEW_PENALTY
Event Type	ORCV_SPPY_EVENT_NEW_PENALTY
Description	Triggered for new Penalties being created.
Trigger	UI/Rule Engine Batch
Target Users	The retailer and supplier users will receive a notification.

SPPY_EVENT_REJ_PENALTY

Penalty rejected.

Property	Value
Event Name	SPPY_EVENT_REJ_PENALTY
Event Type	ORCV_SPPY_EVENT_REJ_PENALTY
Description	Triggered for Penalties being rejected.
Trigger	UI/Rule Engine Batch
Target Users	The retailer and supplier users will receive a notification.

Alerts

SPPY_EVENT_PENALTIES_CHANGING

New penalties have been created or changed status.

Property	Value
Event Name	SPPY_EVENT_PENALTIES_CHANGING
Event Type	ORCV_SPPY_EVENT_PENALTIES_CHANGING
Alert Type	EMAIL
Description	A Penalty was created or changed status.
Trigger	UI/S9T
Target Users	The retailer users will receive an e-mail.

Custom Validations

The module performs validations by implementing Custom Validation Groups of the Custom Validations Framework described in the [Supply Chain Hub Frameworks](#) section. The groups listed below can be used to add Custom Validations for Penalties.

Group Id	Executed When
SPPY_PENALTY_APPROVE	Supplier Penalties approve action.
SPPY_PENALTY_CANCEL	Supplier Penalties cancel action.

Group Id	Executed When
SPPY_PENALTY_CREATE	Supplier Penalties create action.
SPPY_PENALTY_EDIT	Supplier Penalties edit action.
SPPY_PENALTY_REJECT	Supplier Penalties reject action.

SPPY_PENALTY_CREATE

This custom validation group includes by default the following validations in ORCV_SPPY_CVAL_PENALTY package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_DOC_TYPE	Document type are configured for the supplier.	1

SPPY_PENALTY_APPROVE

This custom validation group includes by default the following validations in ORCV_SPPY_CVAL_PENALTY package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_DOC_TYPE	Document type are configured for the supplier.	1

Specific Batch Configuration

This section describes specific batch configurations.

Early Late Shipment Batch

The Early Late Shipment batch generates Penalties for received shipments. For all shipments that are in received status and were not yet previously processed, this batch calls the manage function of SPPY, providing the required information for the penalty creation. This batch uses the Rule Engine framework.

- Job: ORCV_RLEG_SPPY_EARLY_LATE_SHIP_JOB
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

This batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section with the following configurations.

Rule default configuration (ORCV_RLEG_RULE):

RULE KEY	RULE NAME	PROGRAM_NAME	APP ALIAS	RULE MODE
SPPY_EARLY_LATE_SHIP	Early Late Shipment	ORCV_SPPY_RLEG_EARLY_LATE_SHIP	SPPY	BATCH

Rule default parameters (ORCV_RLEG_PARAMETERS):

PARAMETER	DESCRIPTION	VALUE 1	VALUE 2	VALUE 3	VALUE 4
PROCESS_AFTER_DAYS	Indicates the number of days after receive data to process.	10	NA	NA	NA
REQUIRES_RETAILER_APPROVAL	Indicates if Retailer Approval is Needed.	Y	NA	NA	NA

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_RLEG_SPPY_EARLY_LATE_SHIP
MAX_CHUNK_SIZE	1
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

QA Defect Batch

The QA Defect batch generates Penalties for jobs created in the Quality Control module. For all jobs that are in complete with issues status and were not yet previously processed, this batch calls the manage function of SPPY, providing the required information for the penalty creation. This batch uses the Rule Engine framework.

- Job: ORCV_RLEG_SPPY_QA_DEFECT_JOB
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

This batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section with the following configurations.

Rule default configuration (ORCV_RLEG_RULE):

RULE KEY	RULE NAME	PROGRAM_NAME	APP ALIAS	RULE MODE
SPPY_QA_DEFECT	QA Defect	ORCV_SPPY_RLEG_QA_DEFECT	SPPY	BATCH

Rule default parameters (ORCV_RLEG_PARAMETERS):

PARAMETER	DESCRIPTION	VALUE 1	VALUE 2	VALUE 3	VALUE 4
REQUIRES_RETAILER_APPROVAL	Indicates if Retailer Approval is Needed.	Y	NA	NA	NA

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_RLEG_SPPY_QA_DEFECT
MAX_CHUNK_SIZE	1
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Cancel PO Batch

The Cancel PO batch generates Penalties for POs that have quantity canceled. For all POs that have quantity canceled and were not yet previously processed, this batch calls the manage function of SPPY, providing the required information for the penalty creation. This batch uses the Rule Engine framework.

- Job: ORCV_RLEG_SPPY_CANCEL_PO_JOB
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

This batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section with the following configurations.

Rule default configuration (ORCV_RLEG_RULE):

RULE KEY	RULE NAME	PROGRAM_NAME	APP ALIAS	RULE MODE
SPPY_CANCEL_PO	PO Cancelled	ORCV_SPPY_RLEG_CANCEL_PO	SPPY	BATCH

Rule default parameters (ORCV_RLEG_PARAMETERS):

PARAMETER	DESCRIPTION	VALUE 1	VALUE 2	VALUE 3	VALUE 4
REQUIRES_RETAILER_APPROVAL	Indicates if Retailer Approval is Needed.	Y	NA	NA	NA
CANCELLED_CODE	Cancel Code configured to generate Cancelled POs Penalties.	V	NA	NA	NA

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_RLEG_SPPY_CANCEL_PO
MAX_CHUNK_SIZE	1
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3

COLUMN NAME	DEFAULT VALUES
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Penalties Update IMCS IDs

The Penalties Update IMCS IDs updates the IMCS ids in Supplier Penalties table ORCV_SPPY_PENALTY_HEAD.

- Job: ORCV_SPPY_UPD_IMCS_ID_JOB
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

Penalties Events Changes Batch

Batch job to check if there are Penalties pending user actions and creates Alerts.

- Job: ORCV_RLEG_SPPY_EVT_CHG_PENALTY
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_RLEG_SPPY_EVT_CHG_PENALTY
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	Y

Rule Attributes

Rule Attributes are necessary to define the attributes that need to generate penalties via batch. Listed below are the dynamic grid column mapping for each rule.

Early Late Shipment Rule

(ORCV_RLEG_MAP_ATTR_HEAD):

RULE KEY	VIEW NAME	VALIDATION FUNC
SPPY_EARLY_LATE_SHIP	ORCV_V_SPPY_RLEG_EARLY_LATE_SHIP	ORCV_SPPY_RLEG_RULE_ATTR_VAL.VALIDATE_RULE_ATTRIBUTES

(ORCV_RLEG_MAP_ATTR_DETAIL):

SEQ	COLUMN ATTR	label apex	type	list vals	required	default value	max size
1	NUMBER_1	Supplier	SELECTLIST	ORCV_V_SPPY_RLEG_SUPP_TL	0	NULL	NULL
2	VARCHAR_1	Order Context	SELECTLIST	ORCV_V_SPPY_RLEG_PO_TYPE_TL	0	NULL	NULL
3	NUMBER_2	Penalty %	NUMBER	NULL	1	NULL	NULL
4	VARCHAR_2	Early / Late	SELECTLIST	ORCV_V_SPPY_RLEG_EL_TYPE_TL	1	NULL	NULL
5	NUMBER_3	Number Days	NUMBER	NULL	1	NULL	NULL

QA Defect Rule

(ORCV_RLEG_MAP_ATTR_HEAD):

RULE KEY	VIEW NAME	VALIDATION FUNC
SPPY_QA_DEFECT	ORCV_V_SPPY_RLEG_QA_DEFECT	ORCV_SPPY_RLEG_RULE_ATTR_VAL.VALIDATE_RULE_ATTRIBUTES

(ORCV_RLEG_MAP_ATTR_DETAIL):

SEQ	COLUMN ATTR	label apex	type	list vals	required	default value	max size
1	NUMBER_1	Supplier	SELECTLIST	ORCV_V_SPPY_RLEG_SUPP_TL	0	NULL	NULL
2	VARCHAR_1	Order Context	SELECTLIST	ORCV_V_SPPY_RLEG_PO_TYPE_TL	0	NULL	NULL
3	NUMBER_2	Penalty %	NUMBER	NULL	1	NULL	NULL
4	NUMBER_3	Defect Threshold %	NUMBER	NULL	1	NULL	NULL

Cancel PO Rule

(ORCV_RLEG_MAP_ATTR_HEAD):

RULE KEY	VIEW NAME	VALIDATION FUNC
SPPY_CANCEL_PO	ORCV_V_SPPY_RLEG_PO_CANCEL	ORCV_SPPY_RLEG_RULE_ATTR_VAL.VALIDATE_RULE_ATTRIBUTES

(ORCV_RLEG_MAP_ATTR_DETAIL):

SEQ	COLUMN ATTR	label apex	type	list vals	required	default value	max size
1	NUMBER_1	Supplier	SELECTLIST	ORCV_V_SPPY_RLEG_SUPP_TL	0	NULL	NULL
2	VARCHAR_1	Order Context	SELECTLIST	ORCV_V_SPPY_RLEG_PO_TYPE_TL	0	NULL	NULL
3	NUMBER_2	Penalty %	NUMBER	NULL	1	NULL	NULL
4	NUMBER_3	Defect Threshold %	NUMBER	NULL	1	NULL	NULL

Codes

Certain operations require codes to define the options available. Below you can find the list of code types available. The codes are defined in tables ORCV_CODE_HEAD and ORCV_CODE_DETAIL.

CODE TYPE	CODE TYPE DESCRIPTION
PBOA	Penalty Based On - Automatic
PHBO	Penalty Based On
PHHR	Penalty Historic Reasons
PHIL	Add Item List
PHRS	Penalty Reason
PHST	Penalty Status
PRRA	Requires Retailer Approval
PRSA	Requires Supplier Approval
PSEL	Early Late Shipment Values

References

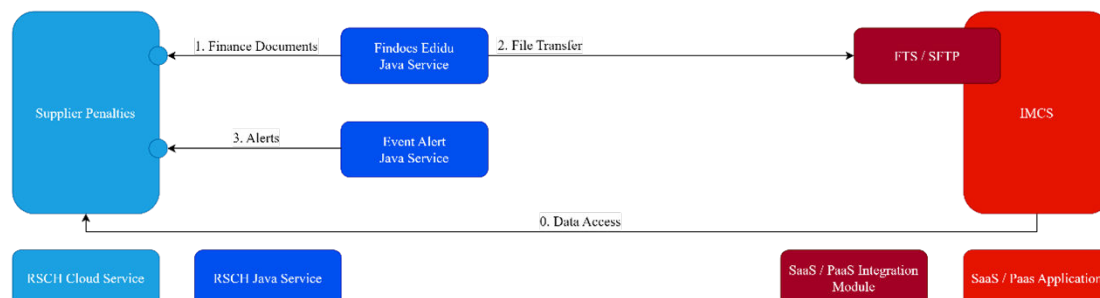
Supplier Penalties also uses the Queries module to enable communication between retailer and supplier users, details about Queries are available in the [Supply Chain Hub Frameworks](#) section.

Operations

This section describes Supplier Penalties operations.

Integration by Module

Figure 21-4 Supplier Penalties Integration by Module



STEP	DESCRIPTION
0	RSCH Data Access of MFCS data
1	Finance Edidu Java Services gets Finance Documents to be integrated to IMCS
2	Finance Edidu Java Services sends produced files to Integration File Transfer.
3	Integration File Transfer publishes files to IMCS File Transfer Services (FTS) or MFCS shared folder (depending on configuration).
4	Event Alert Java Service checks if there are alerts to be triggered.

Batch by Module

This section describes Supplier Penalties batches.

Early Late Shipment Batch

The Early Late Shipment batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section. To execute this batch call Rule Engine Web Service with Rule Key SPPY_EARLY_LATE_SHIP.

QA Defect Batch

The QA Defect batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section. To execute this batch call Rule Engine Web Service with Rule Key SPPY_QA_DEFECT.

Cancel PO Batch

The Cancel PO batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section. To execute this batch call Rule Engine Web Service with Rule Key SPPY_CANCEL_PO. Penalties Update IMCS IDs

Authentication Method: OAuth2

BaseURL: `https://<BASE_URL>/ords/mfcs/orcv/common/batch/<URL_PATH>`

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_SPPY_UPD_IMCS_ID_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTR OL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPY_PENALTY_HEAD ORCV_FIN_INT_DOC_XREF
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPY_UPD_IMCS_ID_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTR OL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SPPY_PENALTY_HEAD ORCV_FIN_INT_DOC_XREF
log	GET	URL query parameters: /log? jobName=ORCV_SPPY_UPD_IMCS_ID_JOB &executionId=<EXEC_ID>&logDate=YYYY -DD-MM	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPY_UPD_IMCS_ID_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTR OL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
status	GET	URL query parameters: /status?jobName= ORCV_SPPY_UPD_IMCS_ID_JOB &executionId=<EXEC_ID>&logDate=YYYY -DD-MM	N/A

Retail Price

The Retail Price module is an application available in Supply Chain Hub. This chapter describes the implementation considerations and the operation details.

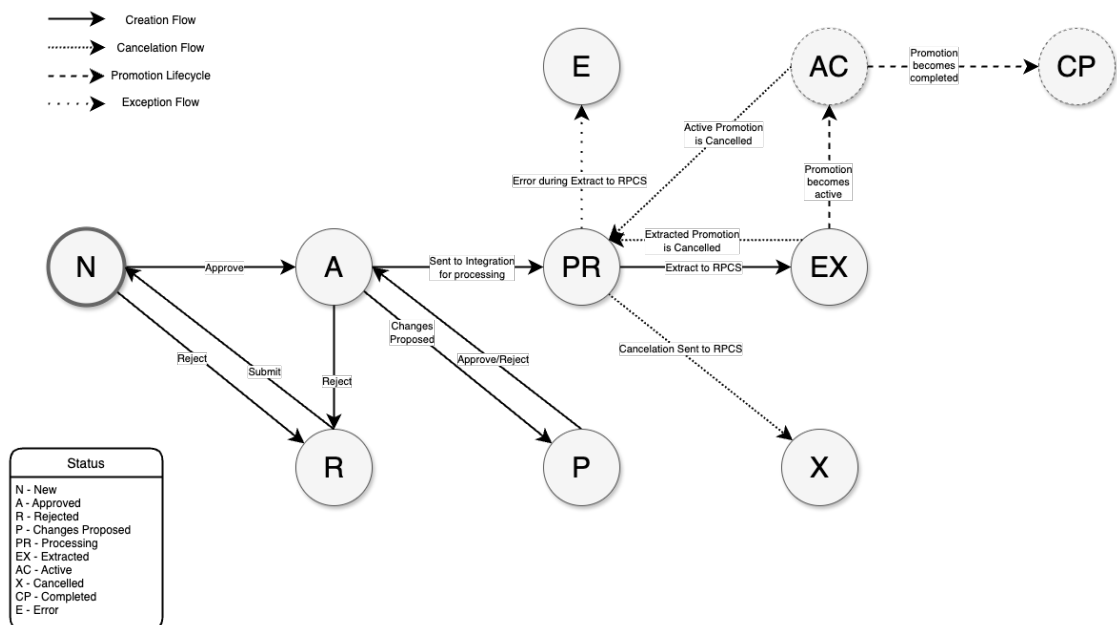
Implementation Overview

Retail Price provides the ability for retailer and supplier users to create, maintain and publish Price Events to RPCS.

Supported Price Event types are Price Changes and Simple Promotions.

Status Workflow

Figure 22-1 Retail Price Status Workflow



Retail Price follows a workflow path for Price Events, and the status can change depending on:

- The user permissions.
- The status of the Price Event.
- The type of the Price Event.

Price Event Status:

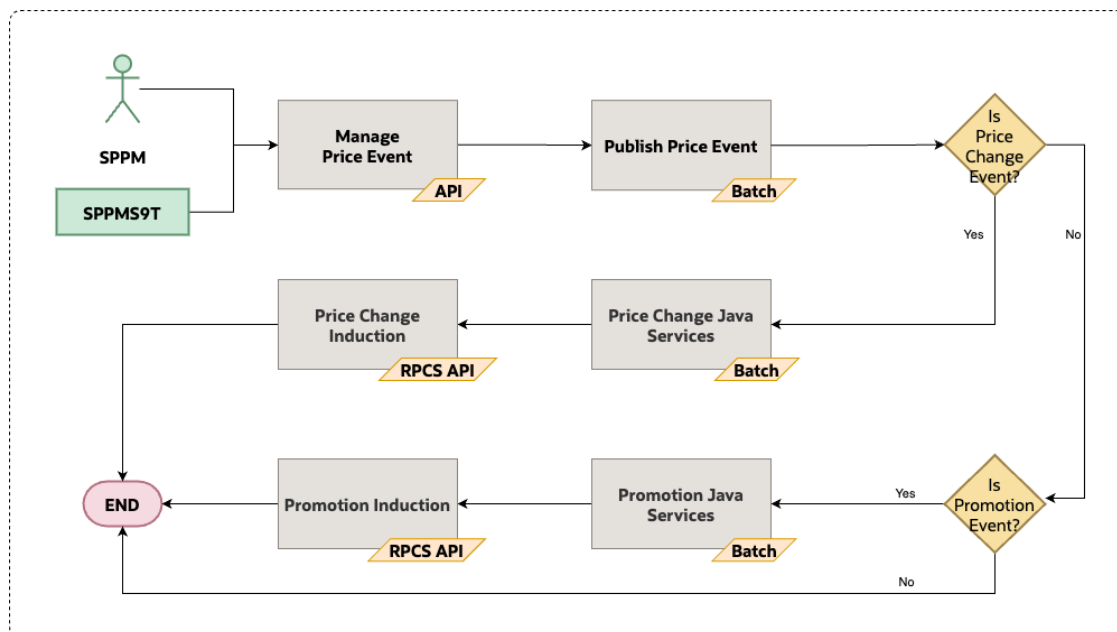
- **New (N)** – The initial status of a Price Event. While in this status, the Price Event can be moved to either Approved or Rejected status, by performing the appropriate actions.

Submitting a New Price Event will automatically approve it if the user has approval permission (the Price Event must not have substantial changes if the user is a supplier).

- **Approved (A)** – Price Event has been approved and is awaiting publishing. While in this status the Price Event can move on to three other status: Rejected if the user rejects it, Changes Proposed if the Price Event is edited or Processing if the Price Event has been picked up by the publish process (triggered either by a batch or automatically if it's an emergency Price Event).
- **Rejected (R)** – Price Event has been rejected and will no longer be published. While in this status the Price Event can be edited, and upon submission will move it back to New status.
- **Changes Proposed (P)** – Approved Price Event has been edited and is awaiting approval. While in this status the Price Event can move on to Approved status if the changes have been approved or rejected, the difference between these two actions is that approving will keep the changes proposed while reject will discard them. If the user editing the Approved Price Event has approval permission, the changes proposed will be automatically approved so this status may not be visible (the Price Event must not have substantial changes if the user is a supplier).
- **Extracted (EX)** – Price Event has been published and is available in RPCS. The final status for a Price Change. A Simple Promotion in this status can either be Cancelled by users with approval permission – which will move it to a Processing status (as the cancelation also needs to be published) – or become Active – which occurs automatically when the current date is between the Simple Promotion's start and end dates.
- **Active (AC)** – Virtual status (not reflected within the database tables) for an Extracted Simple Promotion whose start and end dates includes the current date. Active Simple Promotions can still be cancelled by users with approval permission.
- **Cancelled (X)** – Simple Promotion that has been Cancelled and the cancellation has already been published to RPCS. No actions can be performed on a Cancelled Simple Promotion.
- **Error (E)** – Price Event was picked up by the publish process but was not able to be correctly integrated within RPCS due to errors. No actions can be performed on these Price Events, they must be analyzed and manually recovered by the support team.
- **Processing (PR)** – Price Event has been picked up by the publish process and is awaiting integration into RPCS. No actions can be taken while the Price Event is in this status. Also used as a virtual status for Price Events which are undergoing custom validations asynchronously and cannot not be edited by the user.
- **Completed (CP)** – Virtual status (not reflected within the database tables) for an Extracted Simple Promotion whose end date surpass the current date. No actions can be performed on a Completed Simple Promotion

Integration High Level

Figure 22-2 Retail Price Integration High Level



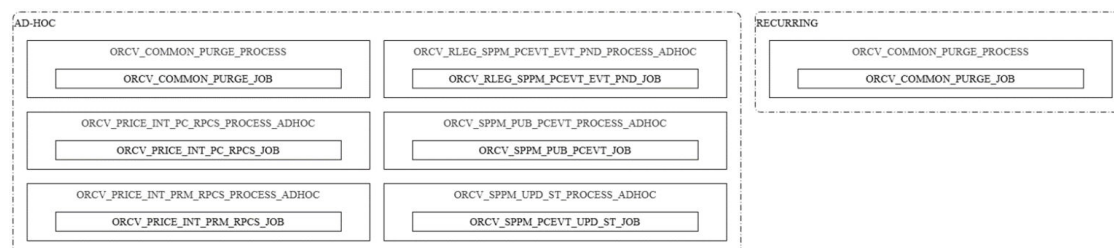
Configuration

This section describes Retail Price configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Figure 22-3 Retail Price Batch Schedule Overview



Promotion and Offer Aggregator

Retail Price can have the group of promotions to be generated with dynamic rules by using the Aggregator Framework documented in the [Supply Chain Hub Frameworks](#) section.

Promotion Aggregator

Configured with AGG_NAME “SPPM_PRM_OF” it generates a Dynamic ID to aggregate Price Events as RPCS Offers in the same Promotion ID based on below default configurations. In case ACTIVE_IND is set no ‘N’ it will be set as “SPPM_SUPPLIER’ + ‘_’ + SUPPLIER + ‘_’ + START_DATE + ‘_’ + EXCLUSIVE_DISCOUNT_IND”

Parameter	Value
APP_ALIAS	SPPM
AGG_NAME	SPPM_PRM
ENTITY_TABLE	ORCV_V_SPPM_PRM_AGG
ENTITY_KEY_1	PRICE_EVENT_ID
ENTITY_KEY_2	NA
ENTITY_KEY_3	NA
ENTITY_KEY_4	NA
ENTITY_KEY_5	NA
AGG_DELIMITER	_
AGG_FIELD_1	'SPPM_SUPPLIER'
AGG_FIELD_2	SUPPLIER
AGG_FIELD_3	START_DATE
AGG_FIELD_4	EXCLUSIVE_DISCOUNT_IND
AGG_FIELD_5	NA
ACTIVE_IND	N

Offer Aggregator

Configured with AGG_NAME “SPPM_PRM_OF” it generates a Dynamic ID to aggregate Price Events as RPCS Offers in the same Offer ID based on below default configurations. In case ACTIVE_IND is set no ‘N’ it will be set as “PRICE_EVENT_ID + ‘_’ + CHANGE_VALUE”

Parameter	Value
APP_ALIAS	SPPM
AGG_NAME	SPPM_PRM_OF
ENTITY_TABLE	ORCV_SPPM_PCEVT_ITEM
ENTITY_KEY_1	PRICE_EVENT_ID
ENTITY_KEY_2	ITEM
ENTITY_KEY_3	CHANGE_VALUE
ENTITY_KEY_4	NA
ENTITY_KEY_5	NA
AGG_DELIMITER	_
AGG_FIELD_1	PRICE_EVENT_ID
AGG_FIELD_2	CHANGE_VALUE
AGG_FIELD_3	NA
AGG_FIELD_4	NA
AGG_FIELD_5	NA
ACTIVE_IND	N

Other Cloud Services Configuration

This section describes implementation details that must be reviewed from an Oracle Retail Merchandising System perspective.

System Options

MFCS System Option CURRENCY_CODE defines the default currency used by Retail Price when creating a new Price Event.

Reason Codes

The Reason Codes defined in MFCS CODE_HEAD and CODE_DETAIL tables with code type PCRC are used to specify which reasons can be used for creating a Price Change.

Other Implementation Considerations

This section describes Retail Price implementation considerations.

Notifications and Alerts

The module uses the Events Framework to inform users about events occurred by generating notifications and alerts. Below is the list of Notifications and Alerts available. The notifications are implemented via the Rule Engine Framework documented in the [Supply Chain Hub Frameworks](#) section.

Notifications

ORCV_SPPM_EVENT_PEND_APPR_PRC

Price Event %s1 pending approval.

Property	Value
Event Name	SPPM_EVENT_PEND_APPR_PRC
Event Type	ORCV_SPPM_EVENT_PEND_APPR_PRC
Description	New Price Change submitted/updated by Supplier
Trigger	UI/S9T
Target Users	Retailer users that are assigned as the retailer contact of the supplier and have access to the Price Change in question.

ORCV_SPPM_EVENT_PEND_APPR_SPROMO

Price Event %s1 pending approval.

Property	Value
Event Name	SPPM_EVENT_PEND_APPR_SPROMO
Event Type	ORCV_SPPM_EVENT_PEND_APPR_SPROMO
Description	New Simple Promotion submitted/updated by Supplier.
Trigger	UI/S9T

Property	Value
Target Users	Retailer users that are assigned as the retailer contact of the supplier and have access to the Simple Promotion in question.

Alerts

ORCV_SPPM_EVENT_PCEVTS_PENDING

Price Events pending approval.

Property	Value
Event Name	SPPM_EVENT_PCEVTS_PENDING_ACTION
Event Type	ORCV_SPPM_EVENT_PCEVTS_PENDING
Alert Type	EMAIL
Description	Price Events with pending approval actions from retailer.
Trigger	Batch
Target Users	Retailer users that are assigned as the retailer contact of the supplier and have access to the Price Event in question.

ORCV_SPPM_EVENT_EMERGENCY_PCEVTS_PENDING

Emergency Price Events with actions pending.

Property	Value
Event Name	SPPM_EVENT_EMERGENCY_PCEVTS_PENDING
Event Type	ORCV_SPPM_EVENT_EMERGENCY_PCEVTS_PENDING
Alert Type	EMAIL
Description	Emergency Price Events with pending approval actions for retailer
Trigger	UI/S9T
Target Users	Retailer users that are assigned as the retailer contact of the supplier and have access to the Price Event in question.

Custom Validations

The module performs validations by implementing Custom Validation Groups of the Custom Validations Framework described in the [Supply Chain Hub Frameworks](#) section. The groups are listed below.

SPPM_PRC_SUBMIT

Triggered when the Price Change is Submitted.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_SUPPLIER	Validate if supplier is valid and user has access to it.	1
VALIDATE_DESCRIPTION	Validate if the description is valid.	2

FUNCTION	VALIDATION	SEQ NO
VALIDATE_PCEVT_TYPE	Validate if the price event type is valid.	3
VALIDATE_REASON_CODE	Validate if the reason code is valid.	4
VALIDATE_LOCATION_LEVEL	Validate if the location level is valid.	5
VALIDATE_STATUS	Validate if the status is valid as per the action to execute.	6
VALIDATE_CHANGE_TYPE	Validate if the change type is valid as per the price event type.	7
VALIDATE_CHANGE_VALUE	Validate if the change value is valid.	8
VALIDATE_EXCLUSIVE_DISCOUNT	Validate if the exclusive discount indicator is valid as per the price event type.	9
VALIDATE_START_DATE	Validate if the start date is valid as per the supplier's configurations.	10
VALIDATE_END_DATE	Validate if the end date is valid as per the price event type.	11
VALIDATE_ITEM	Validate if the items are valid for that price event and the user has access to them.	12
VALIDATE_LOCATION	Validate if the locations are valid for that price event according to the currency/location level and if the user has access to them.	13
VALIDATE_CHG_PROPOSAL_OVERLAP	Validates if the price event has any overlaps with other price events in changes proposed status.	14
VALIDATE_PCEVTS_OVERLAP	Validates if the price event has any overlaps with other approved price events or those in RPCS.	15
VALIDATE_CURRENCY	Validate if the currency is valid.	16

SPPM_PRC_APPROVE

Triggered when the Price Change is Approved.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_STATUS	Validate if the status is valid as per the action to execute.	1
VALIDATE_START_DATE	Validate if the start date is valid as per the supplier's configurations.	2
VALIDATE_CHG_PROPOSAL_OVERLAP	Validates if the price event has any overlaps with other price events in changes proposed status.	3
VALIDATE_PCEVTS_OVERLAP	Validates if the price event has any overlaps with other approved price events or those in RPCS.	4

SPPM_PRC_REJECT

Triggered when the Price Change is Rejected.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_STATUS	Validate if the status is valid as per the action to execute.	1
VALIDATE_START_DATE	Validate if the start date is valid as per the supplier's configurations.	2

SPPM_PRC_CHANGE_PROPOSAL

Triggered when the approved Price Change is Updated.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_STATUS	Validate if the status is valid as per the action to execute.	1
VALIDATE_START_DATE	Validate if the start date is valid as per the supplier's configurations.	2

SPPM_PRC_PUBLISH

Triggered when the Price Change is Published.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_CHANGE_VALUE	Validate if the change value is valid.	1
VALIDATE_CHG_PROPOSAL_OVERLAP	Validates if the price event has any overlaps with other price events in changes proposed status.	2
VALIDATE_PCEVTS_OVERLAP	Validates if the price event has any overlaps with other approved price events or those in RPCS.	3

SPPM_PRC_EDIT

Triggered when the Price Change is Edited.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_STATUS	Validate if the status is valid as per the action to execute.	1
VALIDATE_START_DATE	Validate if the start date is valid as per the supplier's configurations.	2

SPPM_PRC_SUB_CHANGES

Triggered when the Price Change is Submitted.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_SUB_CHANGES_IND	Validate if the price event has substantial changes at header level.	1

SPPM_SPROMO_SUBMIT

Triggered when the Simple Promotion is Submitted.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_SUPPLIER	Validate if supplier is valid and user has access to it.	1
VALIDATE_DESCRIPTION	Validate if the description is valid.	2
VALIDATE_PCEVT_TYPE	Validate if the price event type is valid.	3
VALIDATE_REASON_CODE	Validate if the reason code is valid.	4
VALIDATE_LOCATION_LEVEL	Validate if the location level is valid.	5
VALIDATE_STATUS	Validate if the status is valid as per the action to execute.	6
VALIDATE_CHANGE_TYPE	Validate if the change type is valid as per the price event type.	7
VALIDATE_CHANGE_VALUE	Validate if the change value is valid.	8
VALIDATE_EXCLUSIVE_DISCOUNT	Validate if the exclusive discount indicator is valid as per the price event type.	9
VALIDATE_START_DATE	Validate if the start date is valid as per the supplier's configurations.	10
VALIDATE_END_DATE	Validate if the end date is valid as per the price event type.	11
VALIDATE_ITEM	Validate if the items are valid for that price event and the user has access to them.	12
VALIDATE_LOCATION	Validate if the locations are valid for that price event according to the currency/location level and if the user has access to them.	13
VALIDATE_CHG_PROPOSAL_OVERLAP	Validates if the price event has any overlaps with other price events in changes proposed status.	14
VALIDATE_PCEVTS_OVERLAP	Validates if the price event has any overlaps with other approved price events or those in RPCS.	15
VALIDATE_CURRENCY	Validate if the currency is valid.	16

SPPM_SPROMO_APPROVE

Triggered when the Simple Promotion is Approved.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_STATUS	Validate if the status is valid as per the action to execute.	1
VALIDATE_START_DATE	Validate if the start date is valid as per the supplier's configurations.	2
VALIDATE_CHG_PROPOSAL_OVERLAP	Validates if the price event has any overlaps with other price events in changes proposed status.	3
VALIDATE_PCEVTS_OVERLAP	Validates if the price event has any overlaps with other approved price events or those in RPCS.	4

SPPM_SPROMO_REJECT

Triggered when the Simple Promotion is Rejected.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_STATUS	Validate if the status is valid as per the action to execute.	1
VALIDATE_START_DATE	Validate if the start date is valid as per the supplier's configurations.	2

SPPM_SPROMO_CANCEL

Triggered when the Simple Promotion is Cancelled.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_STATUS	Validate if the status is valid as per the action to execute.	1
VALIDATE_CANCEL_END_DATE	Validate if the end date is valid as per system's batch date.	2
VALIDATE_CANCEL_EVENT_TYPE	Validate if the price event type is valid for cancellation.	3

SPPM_SPROMO_CHANGE_PROPOSAL

Triggered when the approved Simple promotion is Updated.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_STATUS	Validate if the status is valid as per the action to execute.	1
VALIDATE_START_DATE	Validate if the start date is valid as per the supplier's configurations.	2

SPPM_SPROMO_PUBLISH

Triggered when the Simple Promotion is Published.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_CHANGE_VALUE	Validate if the change value is valid.	1
VALIDATE_CHG_PROPOSAL_OVERLAP	Validates if the price event has any overlaps with other price events in changes proposed status.	2
VALIDATE_PCEVTS_OVERLAP	Validates if the price event has any overlaps with other approved price events or those in RPCS.	3

SPPM_SPROMO_EDIT

Triggered when the Simple Promotion is Edited.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_STATUS	Validate if the status is valid as per the action to execute.	1
VALIDATE_START_DATE	Validate if the start date is valid as per the supplier's configurations.	2

SPPM_SPROMO_SUB_CHANGES

Triggered when the Simple Promotion is Submitted.

All validation functions for this Custom Validation Group are available in ORCV_SPPM_CVAL_PCEVT package.

FUNCTION	VALIDATION	SEQ NO
VALIDATE_SUB_CHANGES_IND	Validate if the price event has substantial changes at header level.	1

Specific Batch Configuration

This section describes Retail Price batch configurations.

Publish Price Event Batch

Publish Price Event Batch publishes Price Events in Processing status to Integration API when the start date is greater than the VDATE - PRICE_PRIOR_DEADLINE_DAYS.

- Job: ORCV_SPPM_PUB_PCEVT_JOB
- Frequency: Ad-hoc
- Dependency: None

- Inter Scheduler Dependency: None

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_SPPM_PUB_PCEVT_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Update Price Event Status Batch

Update Price Event Status Batch cross-references Price Events in Processing status with the integration tables and updates the Price Events' status depending on if they were updated successfully or not.

If there were any errors during the integration the batch will also insert the error records into the error tables used by Retail Price.

- Job: ORCV_SPPM_PCEVT_UPD_ST_JOB
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_SPPM_PCEVT_UPD_ST_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Price Event Emergency Alert Batch

Price Event Emergency Alert Batch generates alert emails for price events which are in New status but should now be picked up by the publish batch as their start date is greater than the VDATE - PRICE_PRIOR_DEADLINE_DAYS.

The emails will then be sent to the retailer contacts of the price event's suppliers.

- Job: ORCV_RLEG_SPPM_PCEVT_EVT_PND
- Frequency: Ad-hoc
- Dependency: None

- Inter Scheduler Dependency: None

This batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section with the following configurations.

Rule default configuration (ORCV_RLEG_RULE):

RULE KEY	RULE NAME	PROGRAM_NAME	AP P ALI AS	RU LE MO DE
SPPM_PCEVT_EVT_PND	ORCV_RLEG_SPPM_EVENT_PCEVT_JOB	ORCV_RLEG_SPPM_EVENT_PCEVT_JOB	SPPM	BATCH

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
PROGRAM_NAME	ORCV_RLEG_SPPM_PCEVT_EVT_PND
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
PCCT	Price Change - Change Type
PEPR	Price Event - Promotion Reason Codes
PERR	Price Event - Action Reason Codes
PMAI	Price Event - Add Items by
PMCT	Promotion - Change Type
PMEL	Price Event - Error Level
PMES	Price Event - Event Status
PMET	Price Event - Event Type
PMIT	Price Event - Search by
PMLL	Price Event - Location Level
PMLT	Price Event - Location Type
PMPC	Price Event - Price Control

Templates

Retail Price uses a spreadsheet download and upload process, developed as part of the Retail Supply Chain Frameworks, for creation, maintenance, and extraction of data. To support these processes there are some templates which must be configured, and several configurations should be reviewed as part of the implementation.

Template Price Events

Template key: SPPM_PCEVT_DATA

The required permissions per action for this template are the following:

Table 22-1 Template Key: SPPM_PCEVT_DATA

ACTION TYPE	ROLE NAME
Download Blank	ORCV_SPPM_APP_JOB
Download	ORCV_SPPM_PRC_APPR_JOB
Download	ORCV_SPPM_PRC_EDIT_JOB
Download	ORCV_SPPM_PRC_VIEW_JOB
Upload	ORCV_SPPM_PRC_APPR_JOB
Upload	ORCV_SPPM_PRC_EDIT_JOB
Download	ORCV_SPPM_SPROMO_APPR_JOB
Download	ORCV_SPPM_SPROMO_EDIT_JOB
Download	ORCV_SPPM_SPROMO_VIEW_JOB
Upload	ORCV_SPPM_SPROMO_APPR_JOB
Upload	ORCV_SPPM_SPROMO_EDIT_JOB

Table 22-2 Worksheet Price Event Header

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Price Event ID, Price Event Description, Price Event Type, Supplier, Start Date, Currency, Reason Code, Location Level, Change Type	Creates a new Price Event in New status, can also directly be Approved if the user has the necessary permissions and – if a supplier user – the Price Event has no substantial changes. Depending on the Price Event Type and configuration, certain other fields might also be mandatory, such as End Date and Exclusive Discount.
Update	Price Event ID	Updates an existing Price Event. Only New, Approved, Rejected and Changes Proposed Price Events can be updated. If Approved the Price Event will move to Changes Proposed status (can be directly Approved if the user has the necessary permissions and – if a supplier user – no substantial changes were added). If Rejected the Price Event will move to New status (same auto-approval rules apply). If New or Changes Proposed status the Price Event will maintain the same status (again, can be auto-approved following the rules above).

Table 22-2 (Cont.) Worksheet Price Event Header

ACTION	MANDATORY CELLS / FIELDS	NOTES
Approve	Price Event ID	Approves the Price Event. Only New and Changes Proposed Price Events can be Approved. In both cases the Price Event will move on to Approved status.
Reject	Price Event ID	Rejects the Price Event. Only New and Changes Proposed Price Events can be Rejected. If New the Price Event will move on to Rejected status. If Changes Proposed the Price Event will move back to Approved status with the prior approved data.
Cancel	Price Event ID	Cancels the Price Event. Only Extracted and Active Price Events can be Cancelled. In both cases the status moves on to Cancelled.

Table 22-3 Worksheet Price Event Detail

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Price Event ID, Item ID	Adds an item to the Price Event ID.
Create	Price Event ID, VPN	Adds an item (using the VPN) to the Price Event ID.
Update	Price Event ID, Item ID	Updates the item's Change Value for the Price Event ID.
Update	Price Event ID, VPN	Updates the item's (using the VPN) Change Value for the Price Event ID.
Delete	Price Event ID, Item ID	Removes the item from the Price Event ID.
Delete	Price Event ID, VPN	Removes the item (using the VPN) from the Price Event ID.

Table 22-4 Worksheet Price Event Detail Location

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Price Event ID, Location ID	Adds a store to the Price Event ID.
Create	Price Event ID, Zone ID	Adds a zone to the Price Event ID.
Update	Price Event ID, Location ID	Updates the store's Exclude Indicator.
Delete	Price Event ID, Zone ID	Removes the store from the Price Event ID.
Delete	Price Event ID, Location ID	Removes the zone from the Price Event ID.

Template Simplified Price Event Creation

Template key: SPPM_PCEVT_SIMPLE_DATA

The required permissions per action for this template are the following:

Table 22-5 Template Key: SPPM_PCEVT_SIMPLE_DATA

ACTION TYPE	ROLE NAME
Download Blank	ORCV_SPPM_APP_JOB
Upload	ORCV_SPPM_PRC_APPR_JOB
Upload	ORCV_SPPM_PRC_EDIT_JOB
Upload	ORCV_SPPM_SPROMO_APPR_JOB
Upload	ORCV_SPPM_SPROMO_EDIT_JOB

Table 22-6 Worksheet Price Event

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Price Event Description, Price Event Type, Supplier, Start Date, Currency, Reason Code, Change Type, Change Value, Location Type, Location ID, Item ID	Creates a new Price Event in New status, can also directly be Approved if the user has the necessary permissions and – if a supplier user – the Price Event has no substantial changes. Depending on the Price Event Type and configuration, certain other fields might also be mandatory, such as End Date and Exclusive Discount.
Create	Price Event Description, Price Event Type, Supplier, Start Date, Currency, Reason Code, Change Type, Change Value, Location Type, Location ID, VPN	Creates a new Price Event in New status, can also directly be Approved if the user has the necessary permissions and – if a supplier user – the Price Event has no substantial changes. Depending on the Price Event Type and configuration, certain other fields might also be mandatory, such as End Date and Exclusive Discount.

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_S9T_CODE_HEAD and ORCV_S9T_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
PEHA	Price Events Header - Price Event Data Action
PELT	Price Events Header - Price Event Data Location Type

References

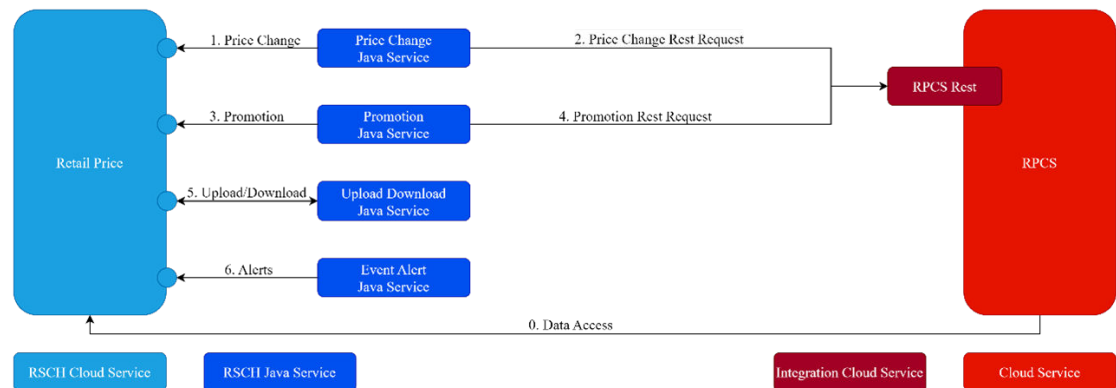
Retail Price also uses the Queries module to enable communication between retailer and supplier users, details about Queries are available in the [Supply Chain Hub Frameworks](#) section.

Operations

This section describes Retail Price operations.

Integration by Module

Figure 22-4 Retail Price Integration by Module



STEP	DESCRIPTION
0	RSCH Data Access of MFCS data
1	Price Change Java Service picks up published Price Changes to be integrated into RPCS.
2	Price Change Java Service calls RPCS Price Change Induction Rest Service.
3	Promotion Java Service picks up published Promotions to be integrated into RPCS.
4	Promotion Java Service calls RPCS Promotion Induction Rest Service.
5	Retail Price calls Upload Download Java Service to transform files, depending on the request details, Upload Download Java Service might need to callback to Retail Price to get more details.
6	Event Alert Java Service checks if there are alerts to be triggered.

Batch by Module

This section describes the Retail Price batches.

Publish Price Event Batch

The batch uses the Batch Framework.

Authentication Method: OAuth2

BaseURL: https://<BASE_URL>/ords/mfcs/orcv/common/batch/<URL_PATH>

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_SPPM_PUB_PCEVT_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CON TROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STAT US ORCV_PRICE_INT_PC_SVC ORCV_PRICE_INT_PRM_HE AD_SVC ORCV_SPPM_PCEVT_HEAD ORCV_SPPM_PCEVT_HEAD_ PREV ORCV_SPPM_PCEVT_ITEM ORCV_SPPM_PCEVT_ITEM_ PREV ORCV_SPPM_PCEVT_LOC ORCV_SPPM_PCEVT_LOC_P REV
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPM_PUB_PCEVT_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CON TROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STAT US ORCV_PRICE_INT_PC_SVC ORCV_PRICE_INT_PRM_HE AD_SVC ORCV_SPPM_PCEVT_HEAD ORCV_SPPM_PCEVT_HEAD_ PREV ORCV_SPPM_PCEVT_ITEM ORCV_SPPM_PCEVT_ITEM_ PREV ORCV_SPPM_PCEVT_LOC ORCV_SPPM_PCEVT_LOC_P REV
log	GET	URL query parameters: /log? jobName=ORCV_SPPM_PUB_PCEVT_JOB &executionId=<EXEC_ID>&logDate=YYY Y-DD-MM	N/A

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPM_PUB_PCEVT_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
status	GET	URL query parameters: /status? jobName=ORCV_SPPM_PUB_PCEVT_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A

Update Price Event Batch

The batch uses the Batch Framework.

Authentication Method: OAuth2

BaseURL: https://<BASE_URL>/ords/mfcs/orcv/common/batch/<URL_PATH>

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_SPPM_PCEVT_UPD_ST_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_PRICE_INT_PC ORCV_PRICE_INT_PC_GRP_XREF ORCV_PRICE_INT_PRM_XREF ORCV_SPPM_PCEVT_HEAD ORCV_SPPM_PCEVT_ITEM

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPM_PCEVT_UPD_ST_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CO NTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STAT US ORCV_PRICE_INT_PC ORCV_PRICE_INT_PC_GRP_ XREF ORCV_PRICE_INT_PRM_XR EF ORCV_SPPM_PCEVT_HEAD ORCV_SPPM_PCEVT_ITEM
log	GET	URL query parameters: /log? jobName=ORCV_SPPM_PCEVT_UPD_ST_J OB &executionId=<EXEC_ID>&logDate=YYYY -DD-MM	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SPPM_PCEVT_UPD_ST_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CO NTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STAT US
status	GET	URL query parameters: /status? jobName=ORCV_SPPM_PCEVT_UPD_ST_J OB &executionId=<EXEC_ID>&logDate=YYYY -DD-MM	N/A

Price Event Emergency Alert Batch

The Price Event Emergency Alert Bbatch uses Rule Engine Framework described in MCEFIO. To execute this batch call Rule Engine Web Service with Rule Key SPPM_PCEVT_EVT_PND.

23

Financial Summary

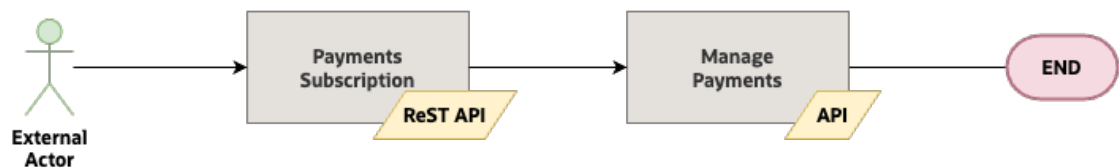
The SPFS module is an application available in Supply Chain Hub. This chapter describes the implementation considerations and the operation details.

Implementation Overview

Financial Summary gives the possibility to share with suppliers visibility of their payment's status.

Integration High Level

Figure 23-1 Financial Summary Integration High Level



Configuration

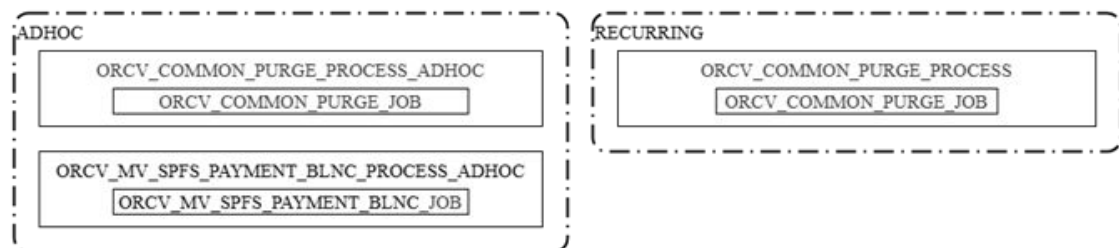
This section describes Financial Summary configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Schedule Overview

Figure 23-2 Financial Summary Batch Schedule Overview



Other Cloud Services Configuration

This section describes implementation details that must be reviewed from an Oracle Retail Merchandising System perspective.

Others

Certain operations require codes to define the options available at system level. Below you can find the list of code types required from MFCS/IMCS.

CODE TYPE	DESCRIPTION
IMTP	Document Type

Other Implementation Considerations

This section describes Financial Summary implementation considerations.

Materialized Views

This application requires some results to be materialized in tables using the Materialized View Framework described in the [Supply Chain Hub Frameworks](#) section, below is the list for this application. Materialized Views used in Financial Summary are listed in this section.

ORCV_MV_SPFS_PAYMENT_BLNC

Description: To calculate and populate current balance to be displayed in Financial Summary Home. The Materialized View records contain the sum of all merchandise invoices and credit notes not associated with any Payment.

Pre-Dependencies: N/A

Post-Dependencies: N/A

Frequency: Ad-hoc

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
FSDL	Dashboard Labels
FSDT	Document Type
FSPD	Payment Date

References

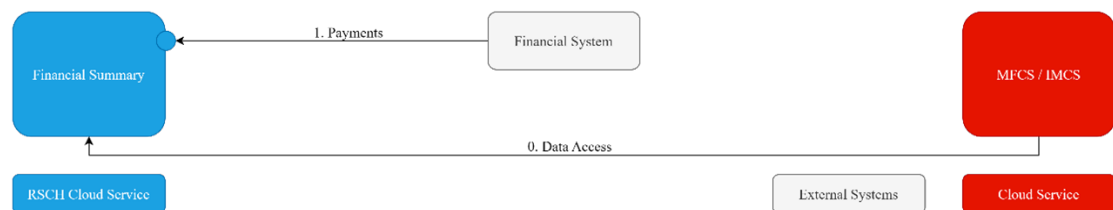
Financial Summary also uses the Queries module to enable communication between retailer and supplier users, details about Queries are available in the [Supply Chain Hub Frameworks](#) section.

Operations

This section describes Financial Summary operations.

Integration by Module

Figure 23-3 Financial Summary Integration by Module



ST EP	DESCRIPTION
0	RSCH Data Access of MFCS data
1	Integration of Payments from External Financial System with to Financial Summary via Rest Service.

Materialized Views Refresh Details

Materialized Views Refresh Details

ORCV_MV_SPFS_PAYMENT_BLNC

PARAMETER	VALUE
Result Synonym	ORCV_MV_SPFS_PAYMENT_BLNC
Result Table A	ORCV_SPFS_PAYMENT_BLNC_A_R
Result Table B	ORCV_SPFS_PAYMENT_BLNC_B_R
Source View	ORCV_V_SPFS_PAYMENT_BLNC_R
Driving View	ORCV_V_SPFS_PAYMENT_BLNC_RC
Pre-Processor	ORCV_COMMON_MV_REFRESH.COMMON_PRE_PROCESSOR
Check Function	NA
Preserve Data	Y
Restart Indicator	N
Max Concurrent Threads	4
Filter Fields	SUPPLIER

Service by Module

This section describes the Financial Summary services.

Payments Upload Service

This service uploads payments details of Financial Documents to Financial Summary. It will create the Payments and associated Documents in Financial Summary. In case the Payment ID already exists the service will overwrite the Payment and all its Documents.

Payment

Payment	
Url Path	<BASE URL>/ords/mfcs/orcv/merch/financialsummary/subscription/payment
Request Method	PUT

Payment	
Sample Request Body	<pre>{ "payments": [{ "payment_id": "6", "supplier": "6200", "payment_date": "2024-05-15", "payment_method": "Bankr", "payment_details": "Account 345678", "taxable_amount": "34899", "payment_status": "A", "total_tax": "200", "documents": [{ "doc_id": "1", "doc_date": "2024-05-10", "doc_currency": "USD", "exchange_rate": "1.23", "doc_amount": "629800", "payment_amount": "629800", "total_tax": "9000", "payment_currency": "USD", "total_payment": "629800" }, { "doc_id": "2", "doc_date": "2024-05-10", "doc_currency": "USD", "exchange_rate": "1.23", "doc_amount": "629800", "payment_amount": "629800", "total_tax": "9000", "payment_currency": "USD", "total_payment": "629800", "doc_type": "MRCHI" }, { "doc_id": "3", "doc_date": "2024-03-11", "doc_currency": "USD", "exchange_rate": "1.23", "doc_amount": "32000", "payment_amount": "32000", "total_tax": "1000", "payment_currency": "USD", "total_payment": "32000" }] }] }</pre>

Payment	
	<pre>] }] } </pre>
Sample Success Message	<i>Empty</i> Response Code: 200 (Success)
Sample Failure Message	<pre> { "statusMsg": "Processed with error.", "errorMsg": "An error occurred while processing some records." } </pre> Response Code: 400 (Bad Request)
Affected Tables	ORCV_SPFS_PAYMENT_DOC ORCV_SPFS_PAYMENT_HEAD ORCV_SPFS_SUB_PAYMENT_DOC ORCV_SPFS_SUB_PAYMENT_HEAD ORCV_SPFS_SUB_PAYMENT_HEAD_ERR

Request Payload Details

Element Name	Required	Data Type	Description
payments	Yes	Collection of Payment Object	List of payments.

Payment Object

Element Name	Required	Data Type	Description
payment_id	Yes	Number (10)	The Payment Id of the financial summary.
supplier	No	Number (10)	The Supplier id.
external_ref_id	No	Number (22)	External Reference Supplier ID.
payment_date	Yes	String (25)	Payment date in format defined in System Parameters DEFAULT_DATE_MASK.
payment_method	Yes	String (25)	The Payment payment method.
payment_details	Yes	String (256)	The Payment details.
taxable_amount	Yes	Number (20,4)	The taxable amount.
payment_status	Yes	String (1)	The payment status.
total_tax	Yes	Number (20,10)	The total tax value.
documents	No	Collection of Document Object	A list of documents associated with the Payment.

Document Object

Element Name	Required	Data Type	Description
doc_id	Yes	String (150)	The Document Id of the financial summary, doc_id must be unique.
doc_date	Yes	String (25)	Document date in format defined in System Parameters DEFAULT_DATE_MASK.
doc_type	Yes	String (6)	The Document Type, the valid values are defined in System Parameter SPFS_DOCT_TYPE
doc_currency	Yes	String (3)	The Document currency in MFCS format
exchange_rate	Yes	Number (20,10)	The Document Exchange rate.
doc_amount	Yes	Number (20,4)	The Document Amount.
payment_amount	Yes	Number (20,4)	The Payment Amount.
total_tax	Yes	Number (20,10)	The Document Total Tax.
payment_currency	Yes	String (3)	The Payment currency.
total_payment	Yes	Number (20,4)	The Payment total in Payment currency.

Appointments Management

The Appointments Management module is an application available in the Supply Chain Hub. This chapter describes the implementation considerations and the operation details.

Implementation Overview

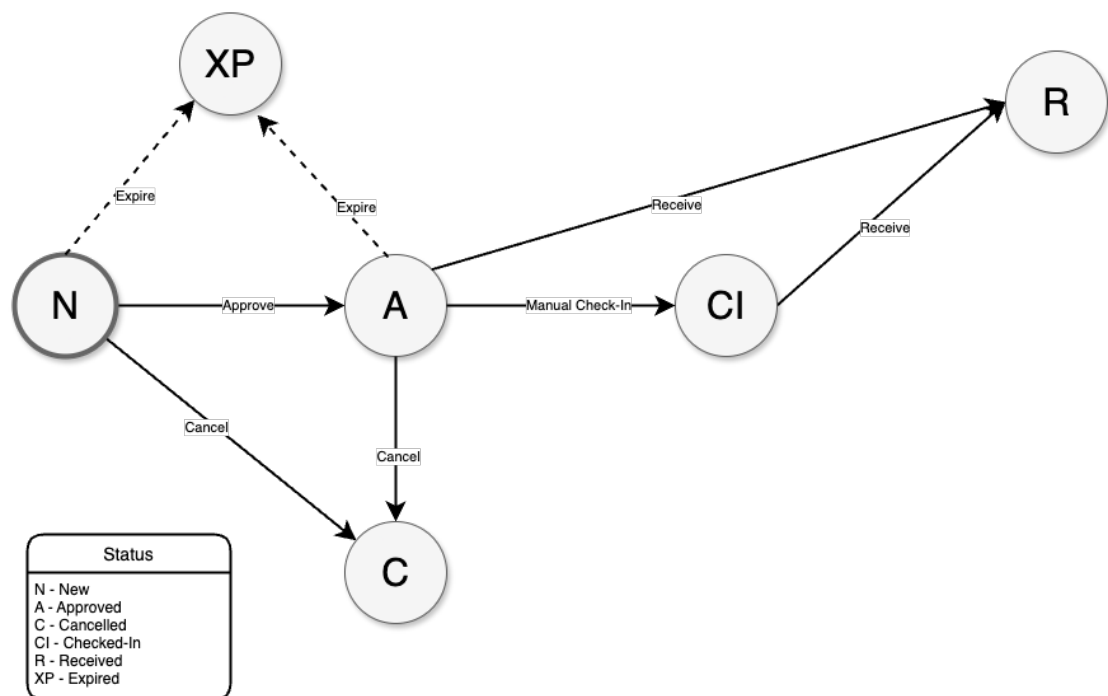
This section provides an implementation overview of Appointments Management.

Functionality

Appointments Management gives the possibility to create and maintain Appointments.

Status Workflow

Figure 24-1 Appointments Management Status Workflow



Appointments Management follows a workflow path for Appointment, and the status can change depending on:

- The user permissions.
- The current status of the Appointment.

Appointment Status:

- New (N) - This is the initial Appointment Status. From this status, it's possible to Cancel and Approve an Appointment.
- Approved (A) - This is the status of an Appointment that was approved. From this status, it's possible to Cancel, Check-in or Receive an Appointment.
- Canceled (X) - This is the status of an Appointment that was canceled. The appointment is not editable whilst on this status.
- Checked-In (CI) - This is the status of an Appointment that was checked in. From this status, it's possible to Receive an Appointment.
- Received (R) - This is the status of an Appointment that was received. The Appointment is not editable whilst on this status.
- Expired (XP) - This is the status of an Approved Appointment where the Appointment date has already passed. The appointment is not editable whilst on this status.

Configuration

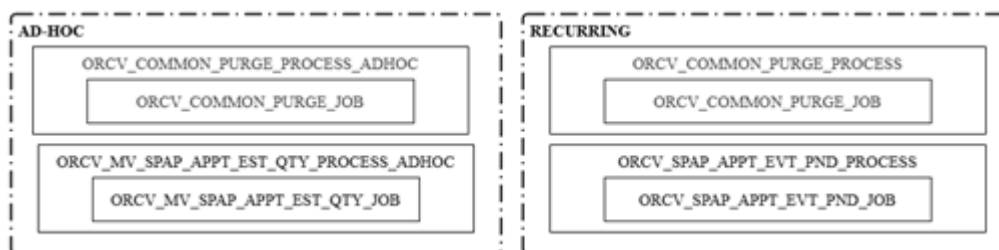
This section describes Appointments Management configuration.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Schedule Overview

Figure 24-2 Appointments Management Batch Schedule Overview



Other Implementation Considerations

This section describes the Appointment Management implementation considerations.

Notifications and Alerts

Appointments Management uses the Events Framework to inform users about some actions with notifications and alerts. Below is the list of notifications available. The notifications are implemented via the Rule Engine Framework documented in the [Supply Chain Hub Frameworks](#) section.

Notifications

SPAP_EVENT_PEND_APPR_APPT - Submitted Appointment is pending approval

Table 24-1 Notifications - Appointments Management

PROPERTY	VALUE
Event Name	SPAP_EVENT_PEND_APPR_APPT
Event Type	ORCV_SPAP_EVENT_PEND_APPR_APPT
Description	Notification triggered when an Appointment is Submitted by a Supplier User without approval permission.
Trigger	UI/S9T
Target Users	The Retailer Contacts of the Supplier with access to the Location of the Appointment.

SPAP_EVENT_CANCEL_APPT - Appointment canceled by the Retailer User

PROPERTY	VALUE
Event Name	SPAP_EVENT_CANCEL_APPT
Event Type	ORCV_SPAP_EVENT_CANCEL_APPT
Description	Notification triggered when an Appointment is Canceled by a Retailer.
Trigger	UI/S9T
Target Users	Supplier Users of the Supplier with access to the Location of the Appointment.

Alerts

SPAP_EVENT_APPTS_PENDING_ACTION

List of future Appointments pending approval.

PROPERTY	VALUE
Event Name	SPAP_EVENT_APPTS_PENDING_ACTION
Event Type	ORCV_SPAP_EVENT_APPTS_PENDING
Alert Type	EMAIL
Description	Email with the list of future Appointments pending approval.
Trigger	Batch ORCV_SPAP_APPT_EVT_PND_JOB
Target Users	The Retailer Contacts of the Supplier with access to the Location of the Appointment.

Custom Validations

The module performs validations by implementing Custom Validation Groups of the Custom Validations Framework described in the [Supply Chain Hub Frameworks](#) section. The groups listed below can be used to add Custom Validations for Appointments and Slots.

Group Id	Executed When
SPAP_APPT_APPROVE	Appointment is approved.
SPAP_APPT_CANCEL	Appointment is canceled.

Group Id	Executed When
SPAP_APPT_CHECK_IN	Appointment is checked in.
SPAP_APPT_RECEIVE	Appointment is received.
SPAP_APPT_SUBMIT	Appointment is submitted.
SPAP_SLOT_AVL_EXC_DELETE	Slot availability exception is deleted.
SPAP_SLOT_AVL_EXC_SUBMIT	Slot availability exception is submitted.

Specific Batch Configuration

Appointments Pending Action Alert Batch

Appointments Pending Action Alert checks if there are Appointments pending approval with Booking Date after Business Data (VDATE) and creates Event Alerts in Events Framework.

- Job: ORCV_RLEG_SPAP_EVENT_APPT_JOB
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

This batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section with the following configurations.

Rule default configuration (ORCV_RLEG_RULE):

RULE KEY	RULE NAME	PROGRAM NAME	APP ALIAS	RULE MODE
SPAP_APPT_EVT_PND	ORCV_RLEG_SPAP_EVENT_APPT_JOB	ORCV_RLEG_SPAP_APPT_EVT_PND	SPAP	Batch

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
Program Name	ORCV_RLEG_SPAP_APPT_EVT_PND
Maximum Chunk Size	1000
Maximum Concurrent Threads	4
Retry Lock Attempts	3
Retry Wait Time	3
Retry Maximum Attempts	3
Retry Ordered	N
Retry Indicator	N

Materialized Views

This application requires some results to be materialized in tables using the Materialized View Framework described in the [Supply Chain Hub Frameworks](#) section, below is the list for this application. Materialized Views used in SPAP are listed in this section.

ORCV_MV_SPAP_APPT_EST_QTY

Description: Returns Appointment IDs and respective estimated quantity. This MV will be used in Appointments Home page.

Pre-Dependencies: None

Post-Dependencies: None

Frequency: Ad-hoc

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
APAR	Appointment - Appointment Create Screen Document Type
APCL	Appointment - Appointment Calendar Day Labels
APDP	Appointment - Appointment Calendar Date Period
APDT	Appointment - Appointment Detail Document Type
APQI	Appointment - Appointment Query information
APRC	Appointment - Appointment Reason Codes
APST	Appointment - Appointment Status
SBRC	Appointment - Slot Blocks Reason Codes
SLAD	Appointment - All Slots Desc
SLAV	Appointment - Slot Availability
SLLT	Appointment - Slot Location Type
SLST	Appointment - Slot Status
SLTP	Appointment - Slot Type

Templates

Appointments uses a spreadsheet download and upload process, developed as part of the Retail Supply Chain Hub Frameworks, to create, maintain, and extract data. Some templates must be configured to support these processes, and several configurations should be reviewed as part of the implementation.

Template Appointments Data

Template key: SPAP_APPTS_DATA

The required permissions per action for this template are the following:

ACTION TYPE	ROLE NAME
Download	ORCV_SPAP_APPT_APPR_JOB
Download	ORCV_SPAP_APPT_EDIT_JOB
Download	ORCV_SPAP_APPT_VIEW_JOB

ACTION TYPE	ROLE NAME
Download Blank	ORCV_SPAP_APP_JOB
Upload	ORCV_SPAP_APPT_APPR_JOB
Upload	ORCV_SPAP_APPT_EDIT_JOB

Worksheet Appointments

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Appointment ID Booking Date Supplier Location Appointment Type Slot ID	NA
Approve	Appointment ID	Updates the appointment status.
Cancel	Appointment ID	Updates the appointment status.
Check-In	Appointment ID	Updates the appointment status.
Receive	Appointment ID	Updates the appointment status.

Worksheet Appointments Details

ACTION	MANDATORY CELLS / FIELDS	NOTES
Create	Appointment ID ASN ID or RTV ID	NA

Worksheet Appointments Items

Appointment Items Worksheet does not have mandatory upload data since it will only be used when downloading appointment data.

Template Slots Data

Template key: SPAP_SLOTS_DATA

The required permissions per action for this template are the following:

ACTION TYPE	ROLE NAME
Download	ORCV_SPAP_SLOT_EDIT_JOB
Download	ORCV_SPAP_SLOT_VIEW_JOB
Download Blank	ORCV_SPAP_SLOT_EDIT_JOB
Download Blank	ORCV_SPAP_SLOT_VIEW_JOB
Upload	ORCV_SPAP_SLOT_EDIT_JOB

Worksheet Slot

ACTI ON	MANDATORY CELLS / FIELDS	NOTES
Cre ate	Slot ID Slot Name Location Time Capacity Multiple Bookings Ind Dock Status Slot Type Availability ¹ Sunday Ind, Monday Ind, Tuesday Ind, Wednesday Ind, Thursday Ind, Friday Ind, Saturday Ind	¹ Daily Indicators are only mandatory if availability is set to "C- Custom Days".
Upd ate	Slot ID Updatable: Slot Name, Location, Time, Capacity, Limitations, Multiple Bookings Ind, Dock, Status, Slot Type, Availability and daily indicators.	Updates the slot with the new information.
Dele te	Slot ID	Deletes the slot.

Operations

This section describes Appointments Management operations.

Batch by Module

This section describes Appointments Management batches.

Appointments Pending Action Alert Batch

The Events Alerts batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section. To execute this batch call Rule Engine Web Service with Rule Key ORCV_RLEG_SPAP_EVENT_APPT_JOB.

Materialized Views Refresh Details

This section describes Materialized Views Refresh Details for Appointments Management.

ORCV_MV_SPAP_APPT_EST_QTY

PARAMETER	VALUE
Result Synonym	ORCV_MV_SPAP_APPT_EST_QTY
Result Table A	ORCV_SPAP_APPT_EST_QTY_A_R

PARAMETER	VALUE
Result Table B	ORCV_SPAP_APPT_EST_QTY_B_R
Source View	ORCV_V_SPAP_APPT_EST_QTY_R
Driving View	ORCV_V_SPAP_APPT_EST_QTY_RC
Pre-Processor	ORCV_COMMON_MV_REFRESH.COMMON_PRE_PROCESSOR
Check Function	NA
Preserve Data	Y
Restart Indicator	N
Max Concurrent Threads	4
Filter Fields	APPOINTMENT_ID

Customer Order Fulfillment

The Customer Order Fulfillment module is an application available in Supply Chain Hub. This chapter describes the implementation considerations and the operation details.

Implementation Overview

This section provides an implementation overview for Customer Order Fulfillment.

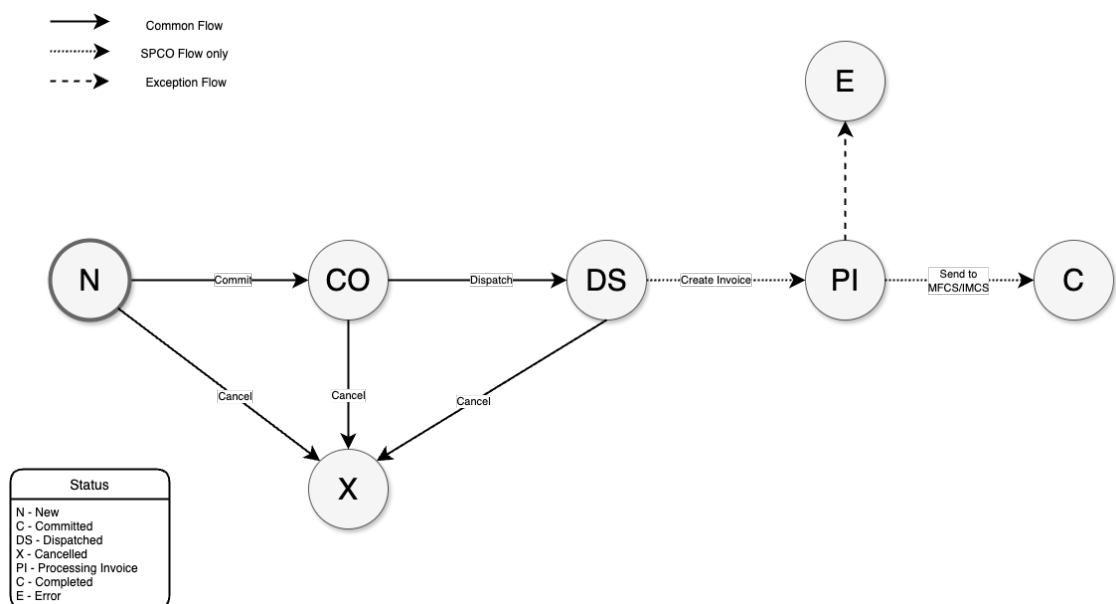
Functionality

This module offers the possibility to review, download and provide updates to Customer Orders. Depending on System Parameters configuration it can follow 2 different flows:

- SPCO Flow: Customer Orders are integrated directly to Customer Order Fulfillment and in Customer Order Fulfillment users can get details, provide status updates and create Invoices for them.
- MFCS Flow: Customer Orders are integrated directly to MFCS and in Customer Order Fulfillment users can get the details and provide status updates.

Status Workflow

Figure 25-1 Customer Order Fulfillment Status Workflow



Customer Order Fulfillment follows a workflow path for Customer Orders and status can change depending on:

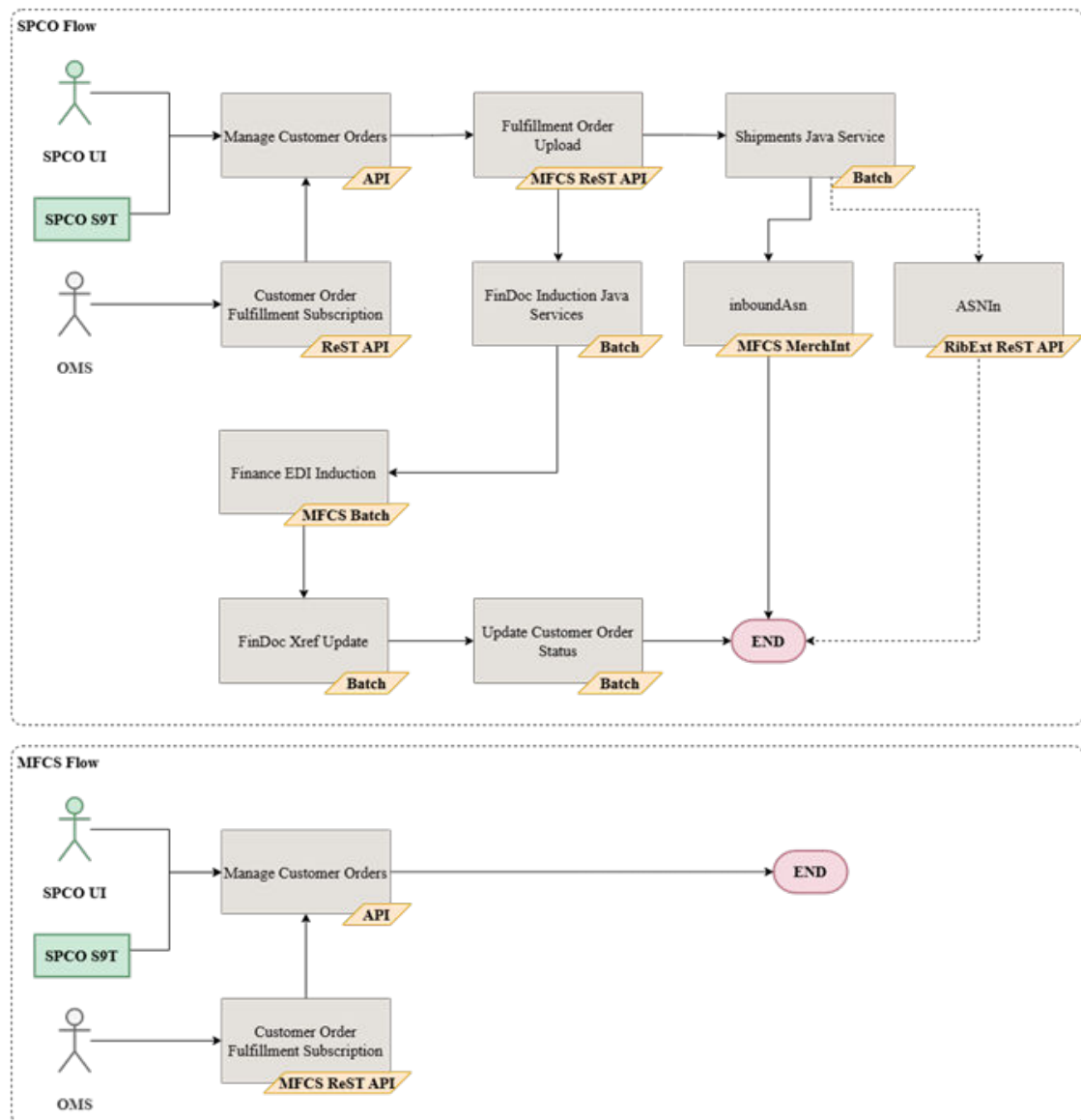
- The user permission
- The current status of the Customer Order

Customer Order Status:

- New (N) - This status is used when the Customer Order are created in MFCS or Customer Order Fulfillment.
- Committed (C) - This Status used when Supplier committed to dispatch the Customer Order, on this status it is possible to provide the Estimated Ship Date and Shipment Status.
- Dispatched (DS) - When the Customer Order is marked as Dispatched by the Supplier the Customer Order moves to this Status. On this status the Customer Order is considered completed whilst in MFCS mode and completed with missing Invoice when in SPCO Mode.
- Cancelled (X) - This Status is used when a Customer Order is cancelled. The cancellation can be initiated by the Supplier or by OMS.
- Processing Invoice (PI) - This Status is only available when Customer Order Fulfillment controls Supplier Sourced Customer Orders instead of MFCS and is used while the consolidated Customer Order, Invoice and ASN are being published to MFCS/IMCS.
- Completed (C) - Once all processing has been completed the Customer Order is moved to Completed.
- Error (E) - This is a technical status used when during the Processing Invoice publishing process an error is found.

Integration High Level

Figure 25-2 Customer Order Fulfillment Integration High Level



Note:

Shipments Java Service integration via RIB-EXT is an optional flow that requires enabling from the configuration. For more details on this process, see the [Supply Chain Hub Frameworks](#) section.

Configuration

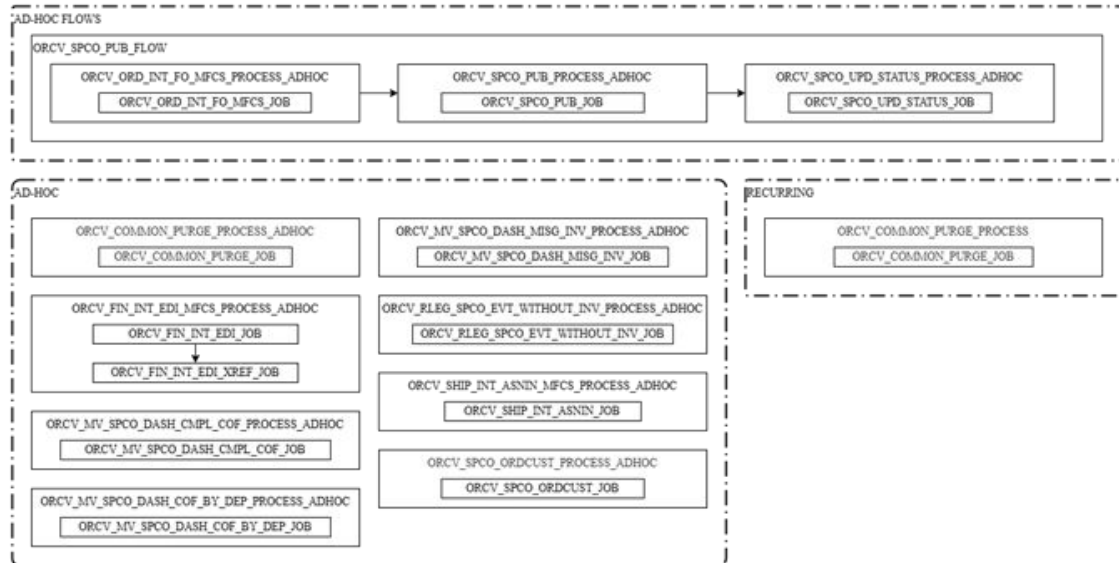
This sections describes Customer Order Fulfillment configurations.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in the [Supply Chain Hub Frameworks](#) section.

Schedule Overview

Figure 25-3 Customer Order Fulfillment Batch Schedule Overview



Other Cloud Services Configuration

This section describes implementation details that must be reviewed from an Oracle Retail Merchandising System perspective.

System Options

MFCS System Option `LOCALIZATION_CONFIG_OPTIONS.DEFAULT_TAX_TYPE` defines the default tax type used for Finance Documents.

Other Implementation Considerations

This section describes Customer Order Fulfillment implementation considerations.

Notifications and Alerts

Customer Orders Fulfillment uses the Events Framework to inform users about some actions with notifications and alerts. Below is the list of notifications available. The notifications are implemented via the Rule Engine Framework documented in the [Supply Chain Hub Frameworks](#) section.

Notifications

SPCO_EVENT_NEW_CUSTOMER_ORDER

New Customer Order(s) received.

Table 25-1 Notifications - SPCO_EVENT_NEW_CUSTOMER_ORDER

PROPERTY	VALUE
Event Name	SPCO_EVENT_NEW_CUSTOMER_ORDER
Event Type	ORCV_SPCO_EVENT_NEW_CUSTOMER_ORDER
Description	Notification triggered when new Customer Order is received.
Trigger	Subscription Rest Service
Target Users	Supplier users that are associated with supplier of the Customer Order

Alerts

SPCO_EVENT_COF_WITHOUT_INV

Customer order fulfillment without invoice.

Table 25-2 Alerts - SPCO_EVENT_COF_WITHOUT_INV

PROPERTY	VALUE
Event Name	SPCO_EVENT_COF_WITHOUT_INV
Event Type	ORCV_SPCO_EVENT_COF_WITHOUT_INV
Alert Type	EMAIL
Description	Email with the Customer Orders without invoice.
Trigger	BATCH
Target Users	Supplier users that are associated with supplier with Customer Orders without invoice.

Custom Validations

The module performs validations by implementing Custom Validation Groups of the Custom Validations Framework described in the [Supply Chain Hub Frameworks](#) section.

The groups listed below can be used to add Custom Validations for Customer Order Fulfillment.

Group Id	Executed When
SPCO_COF_CANCEL	Customer Order is cancelled.
SPCO_COF_COMMIT	Customer Order is committed.
SPCO_COF_CREATE	Customer Order is created.
SPCO_COF_DISPATCH	Customer Order is dispatched.
SPCO_COF_EDIT	Customer Order is Changed.

Specific Batch Configuration

This section describes Customer Order Fulfillment specific batch configurations.

MFCS Control Tables Update Batch

This batch will be responsible for populating the control tables (ORCV_SPCO_ORDCUST_CONTROL, ORCV_SPCO_ORDCUST_DEPT_CONTROL, ORCV_SPCO_ORDCUST_ORD_CONTROL). If the flow is MFCS it will populate the control tables with the Customer Order information existing in the MFCS, enriching this same information with the value of the total_cost, the department of the item(s) belonging to the Customer Order, and the order. If the flow is SPCO it will only calculate the total_cost of the Customer Order, searching for the department of the item(s) belonging to the Customer Order.

- Job: ORCV_SPCO_ORDCUST_JOB
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
Program Name	ORCV_SPCO_ORDCUST_JOB
Maximum Chunk Size	1000
Maximum Concurrent Threads	4
Retry Lock Attempts	3
Retry Wait Time	3
Retry Maximum Attempts	3
Retry Ordered	N
Retry Indicator	N

Shipments and Finance Documents Publish Batch

This batch will be responsible for publishing shipments and financial documents associated with one or more Customer Orders.

- Job: ORCV_SPCO_PUB_JOB
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
Program Name	ORCV_SPCO_PUB_JOB
Maximum Chunk Size	1000
Maximum Concurrent Threads	4
Retry Lock Attempts	3
Retry Wait Time	3

COLUMN NAME	DEFAULT VALUES
Retry Maximum Attempts	3
Retry Ordered	N
Retry Indicator	N

Customer Order Status Update Batch

This batch will be responsible for updating the Customer Order status. If the financial document was not successfully integrated into IMCS, the status will be 'Error', otherwise it will be 'Completed'.

- Job: ORCV_SPCO_UPD_STATUS_JOB
- Frequency: Ad-hoc Flow ORCV_SPCO_PUB_FLOW
- Dependency: ORCV_FIN_INT_EDI_MFCS_PROCESS_ADHOC, ORCV_SHIP_INT_ASNIN_MFCS_PROCESS_ADHOC
- Inter Scheduler Dependency: None

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
Program Name	ORCV_SPCO_UPD_STATUS_JOB
Maximum Chunk Size	10000
Maximum Concurrent Threads	4
Retry Lock Attempts	4
Retry Wait Time	2
Retry Maximum Attempts	3
Retry Ordered	N
Retry Indicator	N

Events Notifications and Alerts Batch

Triggers notifications for Events described in Notifications and Alerts.

- Job: ORCV_RLEG_SPCO_EVT_WITHOUT_INV
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

This batch uses the Rule Engine Framework described in the [Supply Chain Hub Frameworks](#) section with the following configurations.

Rule default configuration (ORCV_RLEG_RULE):

RULE KEY	RULE NAME	PROGRAM NAME	AP P ALI AS	RULE MOD E
SPCO_EVT_WITHOUT_INV	ORCV_SPCO_EVT_WITHOUT_INV	ORCV_RLEG_SPCO_EVT_WITHOUT_INV	SPCO	BATCH

Batch default configuration (ORCV_BATCH_CONFIG):

COLUMN NAME	DEFAULT VALUES
Program Name	ORCV_RLEG_SPCO_EVT_WITHOUT_INV
Maximum Chunk Size	1000
Maximum Concurrent Threads	4
Retry Lock Attempts	3
Retry Wait Time	3
Retry Maximum Attempts	3
Retry Ordered	N
Retry Indicator	Y

Materialized Views

This application requires some results to be materialized in tables using the Materialized View Framework described in the [Supply Chain Hub Frameworks](#) section, below is the list for this application. Materialized Views used in Customer Order Fulfillment home page.

ORCV_MV_SPCO_DASH_COF_BY_DEP

Returns the sum of total cost items by department. Used in Customer Order Fulfillment home page.

Pre-Dependencies: None

Post-Dependencies: None

Frequency: Ad-hoc

ORCV_MV_SPCO_DASH_CMPL_COF

Returns the number of Customer Orders in Completed status (In SPCO flow status = 'C', in MFCS flow status = 'DS') and received in the last 30 days. Used in Customer Order Fulfillment home page.

Pre-Dependencies: None

Post-Dependencies: None

Frequency: Ad-hoc

ORCV_MV_SPCO_DASH_MISG_INV

Returns the number of Customer Orders with dispatched status and without invoice. Used in Customer Order Fulfillment home page.

Pre-Dependencies: None

Post-Dependencies: None

Frequency: Ad-hoc

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
COBL	Badges labels, to be used on the dashboard screen.
CORS	Customer Order Reason.
COST	Customer Order Status.
SHST	Shipment Status.

Templates

Customer Order Fulfillment uses spreadsheets to download and upload process, developed as part of the Retail Supply Chain Hub Frameworks, for creation, maintenance, and extraction of data. To support these processes there are some templates which must be configured, and several configurations should be reviewed as part of the implementation.

Template Customer Order Fulfillment Data

Template key: SPCO_COF

The required permissions per action for this template are the following:

Table 25-3 Template key: SPCO_COF

ACTION TYPE	ROLE NAME
Download Blank	ORCV_SPCO_APP_JOB
Download	ORCV_SPCO_COF_VIEW_JOB
Download	ORCV_SPCO_COF_EDIT_JOB
Upload	ORCV_SPCO_COF_EDIT_JOB

Table 25-4 Worksheet Customer Order Header

ACTION	MANDATORY CELLS / FIELDS	NOTES
Update	Ordcust No	Edits Customer Orders Note: It is only allowed to change the shipment status and expected shipment date fields
Commit	Ordcust No	Commits Customer Orders
Dispatch	Ordcust No	Dispatches Customer Orders
Cancel	Ordcust No	Cancels Customer Orders

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_S9T_CODE_HEAD and ORCV_S9T_CODE_DETAIL tables.

CODE TYPE	CODE TYPE DESCRIPTION
SPCA	Customer Order Fulfillment - Customer Order Action

References

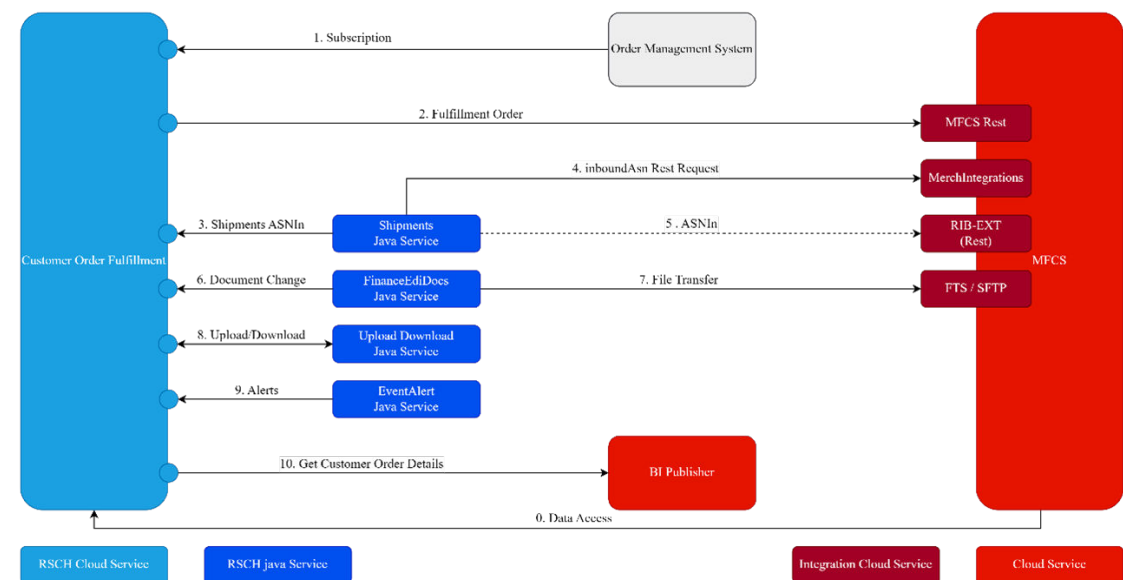
Customer Order Fulfillment also uses the Queries module to enable communication between retailer and supplier users, details about Queries are available in the [Supply Chain Hub Frameworks](#) section.

Operations

This section describes Customer Order Fulfillment operations.

Integration by Module

Figure 25-4 Customer Order Fulfillment Integration by Module



ST EP	DESCRIPTION
0	RSCH Data Access of MFCS data
1	Order Management System calls Customer Order Fulfillment subscription Rest Service.

ST EP	DESCRIPTION
2	Customer Order Fulfillment calls MFCS Fulfillment Upload Rest Service to create/cancel Fulfillment Orders.
3	Shipments Java Services checks if Customer Order Fulfillment has shipments to be published.
4	If there are ASNin to be published, Shipments Java Services sends calls MFCS via RIB-EXT Rest.
5	Finance Document Induction Java Services checks if Customer Order Fulfillment has documents to be published.
6	Finance Document Induction Java Services sends produced files to Integration File Transfer.
7	Integration File Transfer publishes files to MFCS File Transfer Services (FTS) or MFCS shared folder (depending on configuration).
8	Customer Order Fulfillment calls Upload Download Java Service to transform files, depending on the request details, Upload Download Java Service might need to callback to SPCO to get more details.
9	Event Alert Java Service checks if there are alerts to be triggered.
10	Customer Order Fulfillment uses Reports application to call BI Publisher Rest Service to retrieve the Customer Order details from a BI Publisher report.

Services by Module

This section describes the Customer Order Fulfillment services.

Fulfillment Order Service

This service is used to create or cancel Customer Order Fulfillment requests from an external order management system (OMS).

Create

Table 25-5 Create API

Create	
Url Path	<BASE URL>/ords/mfcs/orcv/merch/customerorder/fulfillmentorder/create
Request Method	POST

Table 25-5 (Cont.) Create API

Create	
Sample Request Body	<pre>{ "items": [{ "customerOrderNo": "47244", "fulfillOrderNo": "47244", "details": [{ "item": "100202777", "orderQuantitySuom": 25, "substituteInd": "N", "standardUom": "EA", "transactionUom": "EA", "unitRetail": 99, "retailCurrency": "USD", "comments": "Lalalala", "itemLineNo": 1 }, { "item": "100202742", "orderQuantitySuom": 59, "substituteInd": "N", "standardUom": "EA", "transactionUom": "EA", "unitRetail": 99, "retailCurrency": "USD", "comments": "Test", "itemLineNo": 2 }], "fulfillLocation": 3000, "partialDeliveryInd": "N", "sourceLocationType": "SU", "sourceLocation": 6200, "fulfillLocationType": "V", "deliveryType": "S", "carrierCode": "30", "carrierServiceCode": "D", "consumerDeliveryDate": "2024-06-19", "deliveryCharges": 99, "deliveryChargesCurrency": "USD", "comments": "Test", "orderPlacedStore": null }] }</pre>

Table 25-5 (Cont.) Create API

Create	
	}
Sample Success Message	{ "items": [{ "customerOrderNo": "47244", "fulfillOrderNo": "47244", "confirmationType": "C", "confirmationNo": 1, "fulfillLocation": 3000, "systemCode": "INV", "details": [{ "item": "100202777", "confirmedQuantity": 25, "confirmedQuantityUom": "EA", "itemLineNo": 1 }, { "item": "100202742", "confirmedQuantity": 59, "confirmedQuantityUom": "EA", "itemLineNo": 2 }] }] }] } Response Code: 200 (Success)

Table 25-5 (Cont.) Create API

Create	
Sample Failure Message	<pre>{ "status": "ERROR", "message": "Error found in validation of input payload", "validationErrors": [{ "error": "must be one of Y, N", "field": "createRecord.partialDeliveryInd", "inputValue": "X" }], "businessError": ["Invalid Order Placed Location. Customer order no UNIT_TEST, Fulfill order no UNIT_TEST. 2", "Consumer_delivery_date and Consumer_delivery_time cannot both be populated. Customer order no 03-MAR-24, Fulfill order no 03-MAR-24.", "Invalid substitute indicator for item <string>, Customer order no UNIT_TEST, Fulfill order no UNIT_TEST. Valid values are Y or N.", "Fulfill Location ID must be a valid non-stockholding company store that supports customer orders. Customer order no UNIT_TEST, Fulfill order no UNIT_TEST. V"] }</pre> Response Code: 400 (Bad Request)
Affected Tables	ORCV_ORD_SUB_ORDCUST_HEAD ORCV_ORD_SUB_ORDCUST_DETAIL ORCV_ORD_SUB_ORDCUST_ERR ORCV_SPCO_ORDCUST_HEAD ORCV_SPCO_ORDCUST_DETAIL

Table 25-6 Customer Order Fulfillment Object

Element Name	Required	Data Type	Description
customer_order_no	Yes	String (48)	This column contains the master customer order number from the Order Management System.
fulfill_order_no	Yes	String (48)	This column contains the number from OMS related to the fulfillment details. One or more fulfillment orders could relate back to a single customer order in OMS.
details	Yes	Collection of FulfillOrder Details Object	A list of items associated with Customer Order Fulfillment.
fulfill_location	Yes	Number (10)	This contains the fulfillment location of the customer order. It can only be either a physical store or a virtual store.

Table 25-6 (Cont.) Customer Order Fulfillment Object

Element Name	Required	Data Type	Description
partial_delivery_ind	Yes	String (1)	This column indicates if the order can be picked and shipped partially or if it should be shipped only when complete. Valid values are Y or N.
source_location_type	Yes	String (2)	This column contains the source type for the customer order. Valid values are SU for supplier, ST for store and WH warehouse. This would only contain a value for vendor, warehouse, or multi-site fulfillment orders.
source_location	Yes	Number (10)	This column contains the supplier, store or warehouse the customer order is sourced from. This would only contain a value for vendor, warehouse or multi-site fulfillment orders.
fulfill_location_type	Yes	String (1)	This column contains the fulfillment location type of the customer order. Valid values are S for physical store and V for virtual store.
delivery_type	Yes	String (1)	This column contains the fulfillment method. Valid values are S for ship direct and C for customer pickup.
carrier_code	No	String (4)	This column contains the carrier the order is to be shipped with, if specified on the order.
carrier_service_code	No	String (6)	This column contains the shipping method that was selected by the customer. Valid values are defined as codes for code_type CSVC: F (Free 2 Day Shipping), D (Discounted 4 Day Shipping), O (Overnight Shipping), P (Premium Next Business Day Delivery) and S (Standard Shipping).
consumer_delivery_date	Yes	Date	This column contains the desired date the delivery is required by the customer.
delivery_charges	No	Number (20,4)	This column contains the delivery charges for Drop Ship purchase orders.
delivery_charges_currency	No	String (3)	This column contains the currency of the delivery charge.
comments	No	String (2000)	This column contains any comments about the customer order.
order_placed_store	Yes	Number (10)	Indicates the store number associated with the location that the customer order was placed. For on-line orders this will contain the virtual store number associated with the on-line store. For orders captured in a physical store this will contain the store number for the physical store.

Table 25-7 Fulfillment Order Details Object

Element Name	Required	Data Type	Description
item	Yes	String (25)	This field contains the item in the customer order. This field will contain the original ordered item from the customer order message. This field can also hold the substitute item when a new record is inserted in this table to hold the substitution details when store fulfills using a substitute item.
reference_item	No	String (25)	This field contains the reference item ordered. This will be used only when a specific UPC is ordered. This is supported for vendor drop-ships orders only.
order_quantity_uom	Yes	Number (12,4)	This field holds the quantity ordered for the customer order/fulfillment no/item number. This quantity is in STANDARD_UOM.
substitute_ind	No	String (1)	This field indicates substitute items are allowed for an item on a customer order at the time of store fulfillment. This field is applicable for store fulfillment only in SIM. The valid values are Y or N.
standard_uom	Yes	String (4)	This field contains the standard UOM of the item.
transaction_uom	Yes	String (4)	This field contains the original unit of measure in which the customer had placed the order in OMS. This field is informational only for RMS.
unit_retail	Yes	Number (20,4)	Indicates the unit sales retail of item on the customer order and is maintained in standard UOM.
retail_currency	Yes	String (3)	Indicates currency of unit retail.
comments	Yes	String (2000)	Used to indicate any special instructions for the item, such as services (monograms, engrave, etc).
item_line_no	Yes	Number (10)	Indicates the detail item line number on the customer order.

Cancel

Table 25-8 Cancel API

Cancel	
Url Path	<BASE URL>/ords/mfcs/orcv/merch/customerorder/fulfillmentorder/cancel
Request Method	POST

Table 25-8 (Cont.) Cancel API

Cancel	
Sample Request Body	<pre>{ "items": [{ "customerOrderNo": "5", "fulfillOrderNo": "5", "sourceLocationType": "SU", "sourceLocation": 6200, "fulfillLocationType": "V", "fulfillLocation": 3000, "details": [{ "item": "100202742", "cancelQuantityInSuom": 1, "standardUom": "EA", "transactionUom": "EA", "itemLineNo": 2 }] }] }</pre>
Sample Success Message	<pre>{ "status": "SUCCESS", "message": "Service call is successful" }</pre> Response Code: 200 (Success)
Sample Failure Message	<pre>{ "status": "ERROR", "message": "Error found in validation of input payload", "validationErrors": [{ "error": "must be one of Y, N", "field": "createRecord.partialDeliveryInd", "inputValue": "X" }], "businessError": ["Item %s1 with invalid direct shipment indicator."] }</pre> Response Code: 400 (Bad Request)

Table 25-8 (Cont.) Cancel API

Cancel	
Affected Tables	ORCV_ORD_SUB_ORDCUST_HEAD ORCV_ORD_SUB_ORDCUST_DETAIL ORCV_ORD_SUB_ORDCUST_ERR ORCV_SPCO_ORDCUST_HEAD ORCV_SPCO_ORDCUST_DETAIL

Table 25-9 Customer Order Fulfillment Object

Element Name	Required	Data Type	Description
Element Name	Required	Data Type	Description
customer_order_no	Yes	String (48)	This column contains the master customer order number from the Order Management System.
fulfill_order_no	Yes	String (48)	This column contains the number from OMS related to the fulfillment details. One or more fulfillment orders could relate back to a single customer order in OMS.
details	Yes	Collection of FulfillOrder Details Object	A list of items associated with Customer Order Fulfillment.
fulfill_location	Yes	Number (10)	This contains the fulfillment location of the customer order. It can only be either a physical store or a virtual store.
source_location_type	Yes	String (2)	This column contains the source type for the customer order. Valid values are SU for supplier, ST for store and WH warehouse. This would only contain a value for vendor, warehouse, or multi-site fulfillment orders.
source_location	Yes	Number (10)	This column contains the supplier, store or warehouse the customer order is sourced from. This would only contain a value for vendor, warehouse or multi-site fulfillment orders.
fulfill_location_type	Yes	String (1)	This column contains the fulfillment location type of the customer order. Valid values are S for physical store and V for virtual store.

Table 25-10 Fulfillment Order Details Object

Element Name	Required	Data Type	Description
item	Yes	String (25)	This field contains the item in the customer order. This field will contain the original ordered item from the customer order message. This field can also hold the substitute item when a new record is inserted in this table to hold the substitution details when store fulfills using a substitute item.

Table 25-10 (Cont.) Fulfillment Order Details Object

Element Name	Required	Data Type	Description
cancelQuantityInSuom	Yes	Number (12,4)	This field holds the quantity cancelled for the customer order/fulfillment no/item number. This quantity is in STANDARD_UOM.
standard_uom	Yes	String (4)	This field contains the standard UOM of the item.
transaction_uom	Yes	String (4)	This field contains the original unit of measure in which the customer had placed the order in OMS. This field is informational only for RMS.
item_line_no	Yes	Number (10)	Indicates the detail item line number on the customer order.

Materialized Views Refresh Details

ORCV_MV_SPCO_DASH_COF_BY_DEP

Parameter	Value
Result Synonym	ORCV_MV_SPCO_DASH_COF_BY_DEP
Result Table A	ORCV_SPCO_DASH_COF_BY_DEP_A_R
Result Table B	ORCV_SPCO_DASH_COF_BY_DEP_B_R
Source View	ORCV_V_SPCO_DASH_COF_BY_DEP_R
Driving View	ORCV_V_SPCO_DASH_COF_BY_DEP_RC
Pre-Processor	ORCV_COMMON_MV_REFRESH.COMMON_PRE_PROCESSOR
Check Function	ORCV_SPCO_DASH_MV.CHECK_SPCO_DASH_COF_BY_DEP
Preserve Data	Y
Restart Indicator	N
Max Concurrent Threads	4
Filter Fields	SUPPLIER,DEPT

ORCV_MV_SPCO_DASH_CMPL_COF

Parameter	Value
Result Synonym	ORCV_MV_SPCO_DASH_CMPL_COF
Result Table A	ORCV_SPCO_DASH_CMPL_COF_A_R
Result Table B	ORCV_SPCO_DASH_CMPL_COF_B_R
Source View	ORCV_V_SPCO_DASH_CMPL_COF_R
Driving View	ORCV_V_SPCO_DASH_CMPL_COF_RC
Pre-Processor	ORCV_COMMON_MV_REFRESH.COMMON_PRE_PROCESSOR
Check Function	ORCV_SPCO_DASH_MV.CHECK_SPCO_DASH_CMPL_COF
Preserve Data	Y
Restart Indicator	N
Max Concurrent Threads	4
Filter Fields	SUPPLIER,DEPT

ORCV_MV_SPCO_DASH_MISG_INV

Parameter	Value
Result Synonym	ORCV_MV_SPCO_DASH_MISG_INV
Result Table A	ORCV_SPCO_DASH_MISG_INV_A_R
Result Table B	ORCV_SPCO_DASH_MISG_INV_B_R
Source View	ORCV_V_SPCO_DASH_MISG_INV_R
Driving View	ORCV_V_SPCO_DASH_MISG_INV_RC
Pre-Processor	ORCV_COMMON_MV_REFRESH.COMMON_PRE_PROCESSOR
Check Function	ORCV_SPCO_DASH_MV.CHECK_SPCO_DASH_MISG_INV
Preserve Data	Y
Restart Indicator	N
Max Concurrent Threads	4
Filter Fields	SUPPLIER,DEPT

26

Reports

The Reports module is an application available in Supply Chain Hub. This chapter describes the implementation considerations and the operation details.

Implementation Overview

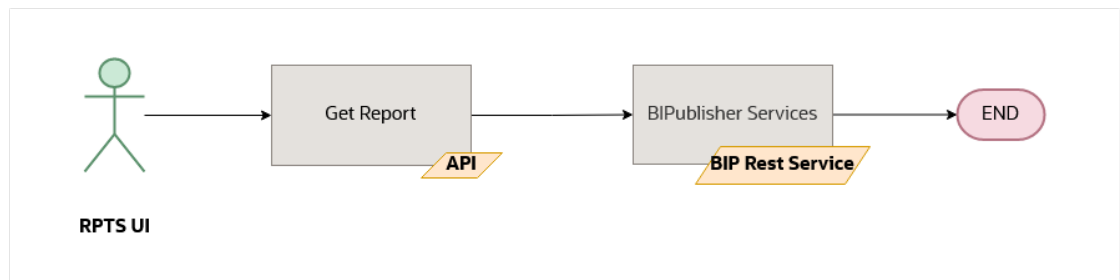
This section provides an implementation overview for Report.

Functionality

Reports gives the possibility for Retailers to share BI Publisher reports with supplier users.

Integration High Level

Figure 26-1 Reports Integration High Level



Configuration

This module has multiple configurations to support its functionalities and, in the subsections, below, those configurations are highlighted. For some of them, it is imperative to validate the functionality of common frameworks in the [Supply Chain Hub Frameworks](#) section.

Report Configuration

In Reports application it is possible to add Reports, it will require the configuration details of the Report Type and Report. The configuration for reports is described in this section.

Database Objects

Configurations details must be defined in Reports tables and Reports filter views created.

Tables

ORCV_RPTS_REPORT_TYPE

This table holds the definition of Report Types and associates with the Application that uses the Report. The access to the Reports is set based on the APP_ALIAS and ENTITY_ID defined at Report Type and only users with access to it will be able to see the Reports of the Report Type. The translations for Report Types are set in ORCV_RPTS_REPORT_TYPE_TL.

Column Name	Data Type	Column Description
REPORT_TYPE_ID	VARCHAR2	This field contains the report type id.
REPORT_TYPE_DESC	VARCHAR2	This field contains the report type description.
APP_ALIAS	VARCHAR2	This field contains the application alias of the report type.
ENTITY_ID	VARCHAR2	This field contains the entity identification of the report type.

ORCV_RPTS_REPORT_HEAD

This table holds the details of Reports, they can be configured with up to 3 filters, those are defined as the name of the view that gets the entries allowed in the filter and they are responsible to apply data level security. The translations for Reports are set in ORCV_RPTS_REPORT_HEAD_TL.

Column Name	Data Type	Column Description
REPORT_ID	VARCHAR2	This field contains the report identification.
REPORT_TYPE_ID	VARCHAR2	This field contains the report type identification according orcv_rpts_report_type.
REPORT_NAME	VARCHAR2	This field contains the report name.
REPORT_DESC	VARCHAR2	This field contains the report description.
ENDPOINT_ID	VARCHAR2	This field contains the endpoint id to call the report, it is required to be configured in ORCV_WS_ENDPOINT.
REPORT_URL_PATH	VARCHAR2	This field contains the url path of the report.
PAYLOAD	VARCHAR2	This field contains the payload for call the report.
FILTER_VIEW_1	VARCHAR2	This field contains the name of the view to filter 1.
FILTER_VIEW_2	VARCHAR2	This field contains the name of the view to filter 2.
FILTER_VIEW_3	VARCHAR2	This field contains the name of the view to filter 3.
FILTER_LABEL_1	VARCHAR2	This field contains the label to filter 1.
FILTER_LABEL_2	VARCHAR2	This field contains the label to filter 2.
FILTER_LABEL_3	VARCHAR2	This field contains the label to filter 3.
STATUS	VARCHAR2	This field contains the status of the report. It can be "Active"/"Inactive".
OUTPUT_FILE_NAME	VARCHAR2	This field contains the name of the report generated.
REPORT_ICON	VARCHAR2	This field contains the report icon image.

Views

It is possible to configure Reports filters based on views, the views must follow the structure below. The reports will be retrieve passing the CODE as the value.

Column Name	Data Type	Column Description
CODE	VARCHAR2	Code value of the filter to be passed to the report.
CODE_DESC	VARCHAR2	Code Description to be shown in Reports pages.
SEQ_NO	NUMBER	Column to order the filter presented in Reports pages.

Packages

When the Reports are requested from Reports pages it will use the following packages to retrieve the Report.

Package Name	Package Description
ORCV_RPTS_MANAGE_P100_UI	Package used in Apex to call Reports API.
ORCV_RPTS_REPORT	Package that uses internal frameworks and retrieves the Report.

Configuration Steps

#	Step	Example
1	Configure the endpoint	Configure the endpoint to be used in Reports in ORCV_WS_ENDPOINT.
2	Create the views defined as Filter Views	<pre>CREATE OR REPLACE VIEW NB_V_ORDER_FILTER_LOV (CODE, CODE_DESC, SEQ_NO) AS SELECT order_no code, 'Order No ' order_no code_desc, ROWNUM seq_no FROM dcv_wv_ordhead oh WHERE EXISTS (SELECT 1 FROM orcv_v_sppp_suppliers sec WHERE oh.supplier = sec.supplier AND ROWNUM = 1) ORDER BY order_no;</pre>
3	Add a Report Type to ORCV_RPTS_REPORT_TYPE	<pre>REPORT_TYPE_ID: SPPP_SAMPLE REPORT_TYPE_DESC: Procure to Pay Sample APP_ALIAS: SPPP ENTITY_ID: ORDER</pre>

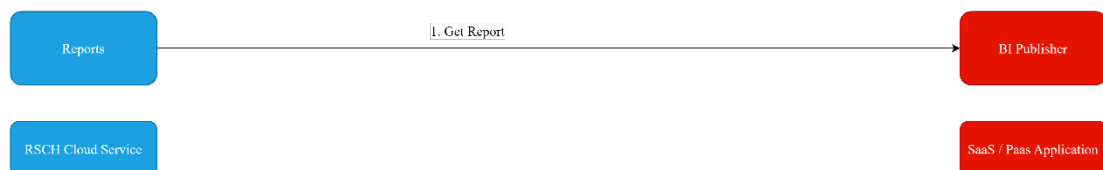
#	Step	Example
4	Add a Report to ORCV_RPTS_REPORT_HEAD	<pre> REPORT_ID: SPPP_SAMPLE_ORDER_PDF REPORT_TYPE_ID: SPPP_SAMPLE REPORT_NAME: Order Detail Sample REPORT_DESC: Order Detail Sample in PDF format ENDPOINT_ID: ORCV_OBIP_REPORTS REPORT_URL_PATH: MFCS%252FOrders%252Ford_det%252Ford_det.xdo/run PAYLOAD: <--Boundary Content-Type: application/json Content-Disposition: form-data; name=""ReportRequest"" { ""byPassCache"": true, ""attributeLocale"": ""en-US"", ""attributeFormat"": ""pdf"", ""parameterNameValues"": { ""listOfParamNameValues"": { ""item"": [{ ""name"": ""PM_order_no"", ""values"": [{""item"": ""@FILTER_1@""}] }] } } } --Boundary--> FILTER_VIEW_1: NB_V_ORDER_FILTER_LOV FILTER_LABEL_1: Order No STATUS: A OUTPUT_FILE_NAME: Order_Detail.pdf </pre>

Operations

This section describes Report operations.

Integration by Module

Figure 26-2 Reports Integration by Module



STEP	DESCRIPTION
1	Get Report from BI Publisher Rest Service.

Supply Chain Hub Frameworks

This section covers the Frameworks and some of the key considerations and common component details when implementing these solutions.

Getting Started

ReST Web Services Endpoints

To call ReST Web Services Endpoints in Retail Supply Chain Hub it is required to use a ReST Client that can connect over HTTP protocol. The ReST Client must be able to reach the server URL and requires authentication with the appropriate permissions granted to perform the HTTP method of GET, POST, PUT and DELETE as per service requirements.

The following details must be taken in consideration:

- REST Server URL
 - Throughout this document you will see several references to a placeholder <BASE_URL>, when calling the service, the placeholder needs to be replaced by the REST Server URL.
- IDCS, Access Token URL and scope to obtain Oauth 2.0 Access Token
 - More details about configuration are available in Security, Rest Services Authentication.

Configure your REST client

With the information gathered so far, you are ready to configure your client to send a REST HTTP request.

1. Construct the request URL. The URL is built using the reference URL from the webservice you are using by substituting the <BASE_URL> placeholder by the REST server URL
2. Choose a ReST Client and perform authentication:

- a. Using cURL to obtain Oauth Access Token:

```
curl -u <client id>:<client secret> -H 'Content-Type: application/x-www-form-urlencoded; charset=UTF-8' -request POST https://<IDCS_BASE_URL>/oauth2/v1/token -d 'grant_type=client_credentials&scope=rgbu:orcv:merch'
```

This will return a JSON payload containing access token like below:

```
{"access_token":"<access token>","token_type":"Bearer","expires_in":3600}
```

Include this access token in the 'Authorization' header of the request. If you are using cURL, use `--header` or `-H` as follows:

```
{--header 'Authorization: Bearer <access token>'}
```

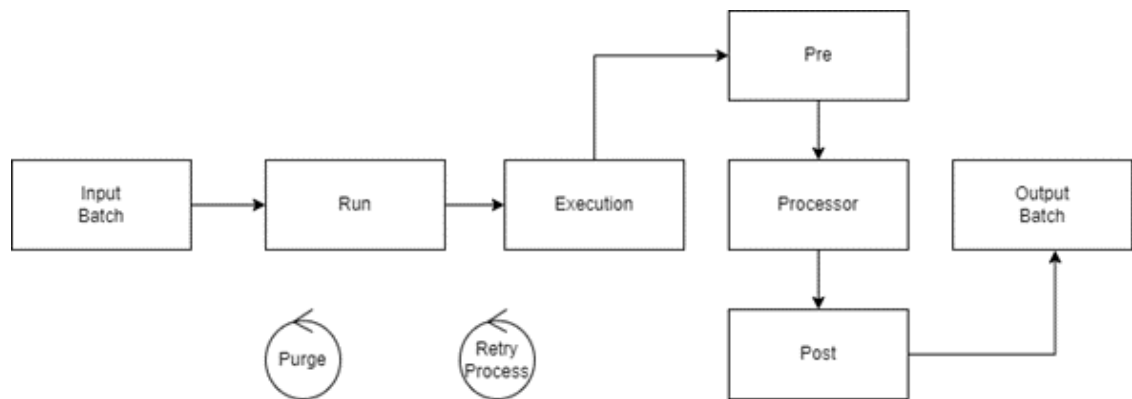
- b. Using Postman with Oauth 2.0 authorization token. To receive an Oauth token, go to the Authorization tab and select Oauth 2.0 from the drop down. Enter the URL, Client ID, Client Secret and Scope and click Get New Access Token. Then click Proceed and Use Token in the next dialog. Then click Send to call the endpoint.

Batch Framework

This section describes the Batch Framework.

Batch Engine Components

Figure 27-1 Batch Engine Components



Configuration

Configuration table: ORCV_BATCH_CONFIG

- Defines:
 - What is the program batch, and each function name for the Pre, Processor, and Post Processor execution steps
 - Maximum number of threads, the maximum number of retry attempts, the maximum number of retry lock attempts, and the maximum number of retry counts
 - Defined whether a process is ordered or not

Pre Processor

- Checks if the process ID is a retry
- Starts audit log process for the batch program
- If it is not a retry process, calls the PRE function associated with the program batch and sets thread control.



Note:

All process batches use the ORCV_THREAD_CONTROL table to set the weight of threads and control the retry mechanism.

Processor

- Gets the program batch configuration
 - Max concurrent threads
 - Retry lock attempts
 - Retry wait time
- If the process is Asynchronous:
 - Creates scheduled jobs until reaching the maximum number of concurrent threads defined in the configuration
- Calls the Processor function associated with the program batch in the configuration table

Post Processor

- In the case of an Asynchronous processor, it will wait for the Processor to finish
- Checks for errors associated with the process ID
- Invokes the Post Processor function configured for the program batch
- Sets the process batch as finished in the batch audit log table

Retry Mechanism

- Retrieves all process IDs with an error that have the number of retry counts not equal to the maximum number of retries in the configuration for that batch program
- Invokes the Run function

Configuration

Table	Description
ORCV_BATCH_ENGINE	This table contains the configuration settings for batch programs: Batch program name Pre, Processor, and Post functions If batch status will be controllable in ORCV_BATCH_STATUS_CONTROL
ORCV_BATCH_CONFIG	This table contains batch program configuration settings: Max threads, max retry attempts, max retry count Retry rule implemented as FIFO

Technical Batch Recovery Actions

If there is a technical issue with a particular batch execution and the batch shows as In Progress on a New Start and does not allow a new execution, records related to the execution/ process ID with issues need to be deleted from the following table to allow a new start:

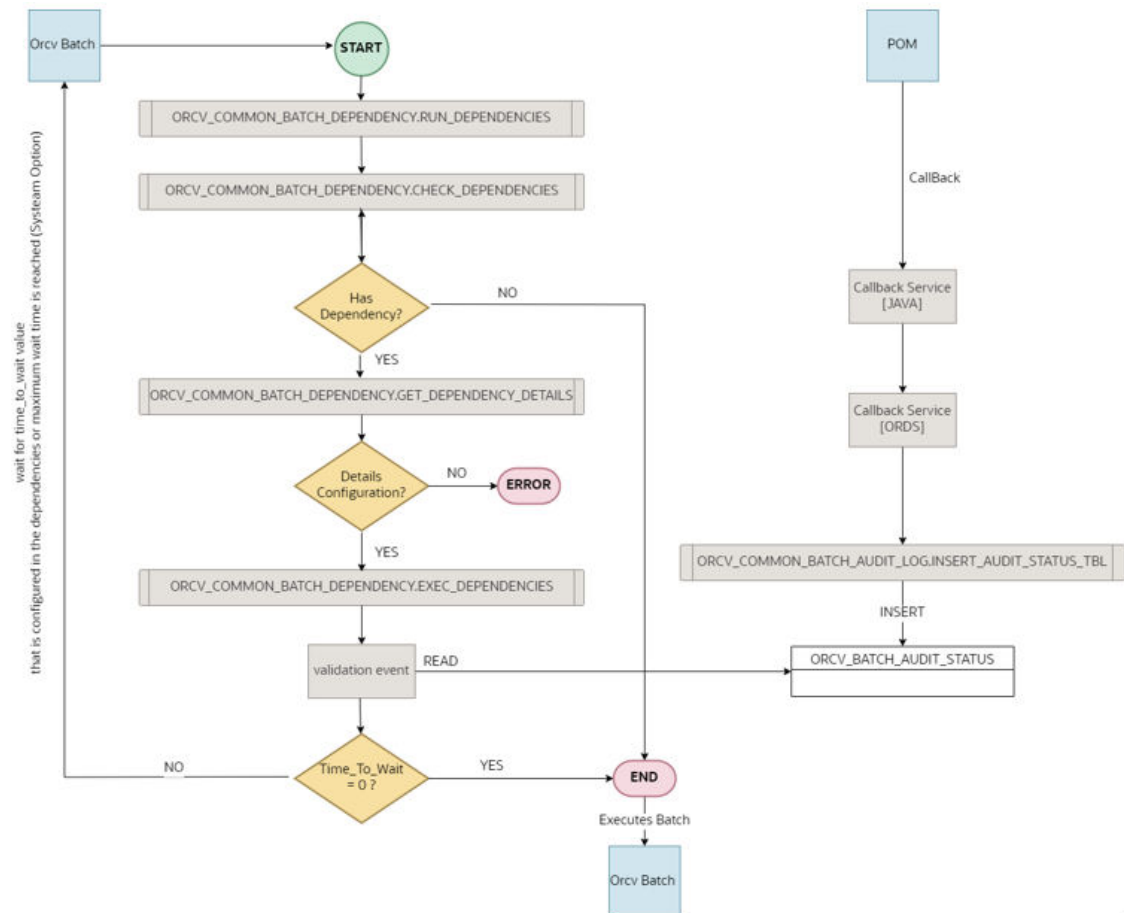
- ORCV_BATCH_STATUS_CONTROL
- ORCV_BATCH_THREAD_CONTROL
- ORCV_BATCH_ENGINE_THREADS

Batch Dependency Framework

Batch Framework will allow handling batches whose execution depends on others, which may be internal jobs or external.

Batch Flow Diagram

Figure 27-2 Batch Dependency Framework Design Overview



Main Database Functions

Package: ORCV_COMMON_BATCH_DEPENDENCY

- RUN_DEPENDENCIES:
 - Function that orchestrates the steps necessary to handle the dependencies between batches
- CHECK_DEPENDENCY
 - Checks if, for the given program name, there is an active dependency configuration to handle
 - If not, returns TRUE

- **GET_DEPENDENCY_DETAILS**
 - Gets the dependencies configuration for a program name
 - If no dependency details exist, it returns FALSE
- **EXEC_DEPENDENCIES**
 - Executes actions/rules defined at dependencies configuration
 - Executes/calls the validation event configured as a dependency for the program name
 - If it ends successfully, will return the time (in minutes) the batch must wait before it can start
 - It will return a time_to_wait of 0 if the batch must run immediately
- **VALIDATE_RPM**
 - Used for batch jobs that need to run after RPCS/RPM jobs.
 - It checks the synonym DCV_WV_JOB_AUDIT_LOG for an entry of the job name from the configuration details to validate the dependency
- **VALIDATE_RMS_WRAPPER**
 - Used for batch jobs that need to run after MFCS batches running an RMS Wrapper
 - It checks the synonym DCV_WV_JOB_AUDIT_LOG for an entry of the job name in the configuration details to validate the dependency
- **VALIDATE_ORCV_AUDIT**
 - Validation function for internal batches that only need ORCV_BATCH_AUDIT_STATUS table validation
- **VALIDATE_TRUE**
 - Validation function for batches that only return true. Always returns a time_to_wait of 0
- **GET_LAST_AUDIT**
 - This function will return the last date time and activity status that the “child” batch was written into the table ORCV_BATCH_AUDIT_STATUS
 - The last status and date time will be returned with values only if the “child” batch has a date time higher than the logs date time of the “parent” batch:
 - * Or if there are no logs for the “parent” batch
 - * Or in case the “parent” batch has logs (it has been executed) at a moment right before the child is being executed.

System Parameters

Default batch configuration is based on System Parameters defined below.

Area	Parameter	Description	Value 1	Value 2
BATCH	BATCH_USER	Default integration user.	MFCS_RDS_CUSTOM (Schema Owner)	NA
BATCH	MAX_THREADS_ENGIN E	The default number of concurrent threads on the batch engine	240	5

Area	Parameter	Description	Value 1	Value 2
BATCH	TIME_INTERVAL	Time interval in minutes for "AuditStatus" and "AuditLogs" validation for batch dependencies	30	NA
BATCH	TOTAL_WAIT_TIME	Total wait time in minutes for "AuditStatus" and "AuditLogs" validation for batch dependencies	720	NA
BATCH	WAIT_PROCESS	Default time in second to wait for next loop	1	NA

Batch Dependency Configuration Example

Configuration table: ORCV_BATCH_DEPENDENCY_HEAD

Column Name	Description	Example
PROGRAM_NAME	This field contains the program name	ORCV_MV_PCEG_ORDRCV_TRAN_DATA
ACTIVE	This field is a flag to indicate if rule is Y - Active or N - Inactive	Y

Configuration table: ORCV_BATCH_DEPENDENCY_DTL

Column Name	Description	Example
PROGRAM_NAME	This field contains the program name	ORCV_MV_PCEG_ORDRCV_TRAN_DATA
CHILD_PROGRAM_NAME	This field contains the program name which the parent program is dependent on	MERCH_DUMMY_START_NIGHT_BATCH_PROCESS
CHILD_ACTIVITY_NAME	This field contains the activity name which the parent program is dependent on	SALSTAGE_JOB
VALIDATION_TYPE	This field contains the type of validation for the dependency. TIME or VALIDATION	VALIDATION
VALIDATION_TIME	This field contains the time to wait for the program name before checking again	10
VALIDATION_EVENT	This field contains the time to wait for the program name before checking again	ORCV_COMMON_BATCH_DEPENDENCY. VALIDATE_RMS_WRAPPER
POST_DATE_CHECK	Value to subtract to VDATE in the validation process	NULL
JOB_NAME	Job name on job_audit_log table	Salstage
HAS_WRAPPER	Controls if the dependency job uses a wrapper. Y - Yes, N - No	Y
MESSAGE_CHECK	Message to check on job_audit_log table for the end of single execution batches	salstage

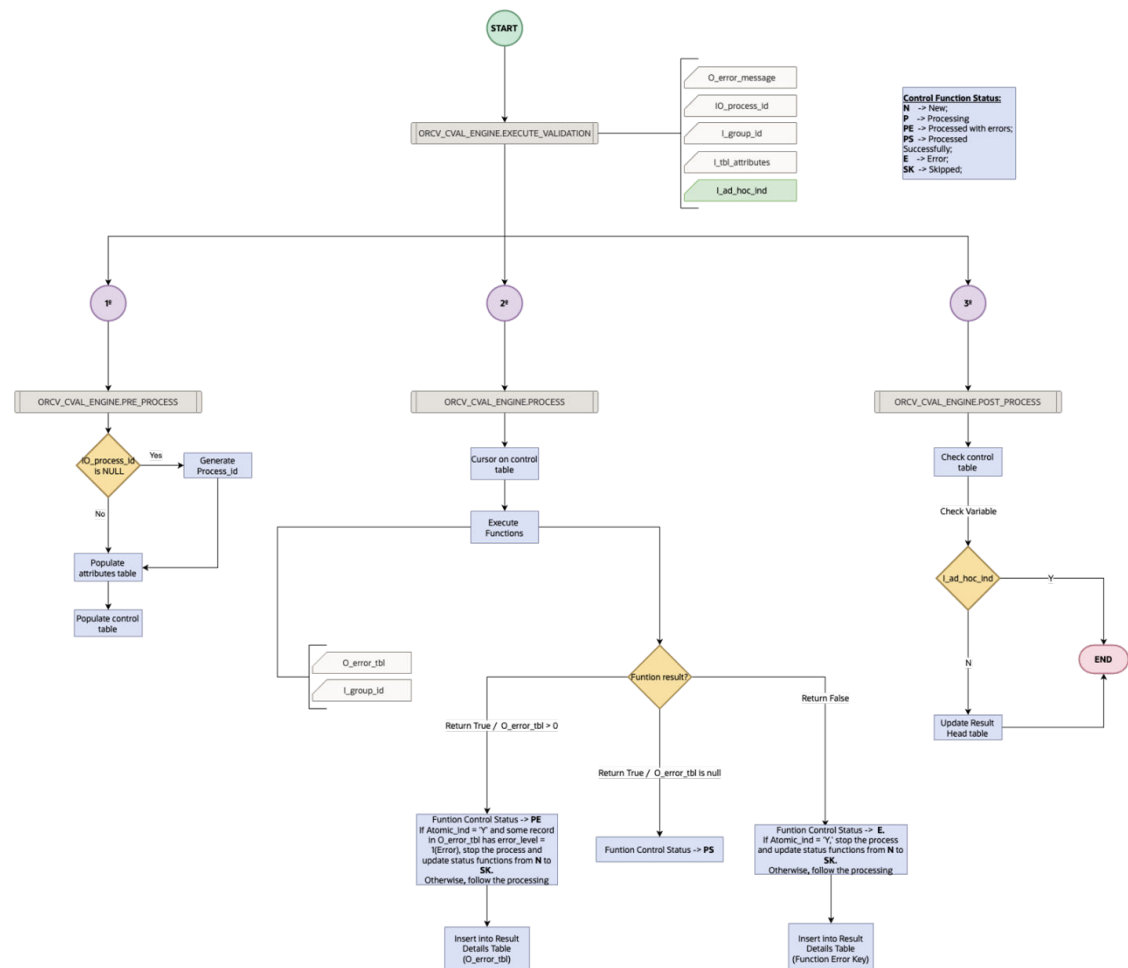
Custom Validations Framework

The Custom Validations Framework works as a validation engine, allowing you to create or extend validation groups with multiple validation functions and then execute the validations group in bulk in an atomic or non-atomic way.

Custom validations are defined in multiple tables with prefix ORCV_CVAL and their executions are controlled by the table ORCV_CVAL_PROCESS_CONTROL. The validations result in case of errors are consolidated in the tables ORCV_CVAL_RESULTS_HEAD and ORCV_CVAL_RESULTS_DETAIL.

Process Overview

Figure 27-3 Custom Validations Process Overview



Database Objects

Object Type	Name	Sub Object Type	Sub Object Name	Description
Table	ORCV_CVAL_GROUP_HEAD	NA	NA	This table contains the custom validation group header information
Table	ORCV_CVAL_GROUP_DETAIL	NA	NA	This table contains the functions of the validation group
Table	ORCV_CVAL_PROCESS_CONTROL	NA	NA	This table contains the control of process validation
Table	ORCV_CVAL_PROCESS_ATTRIBUTES	NA	NA	This table contains the attributes to the functions of the process
Table	ORCV_CVAL_RESULT_HEAD	NA	NA	This table contains the attributes to the functions of the process
Table	ORCV_CVAL_RESULT_DETAIL	NA	NA	This table contains the attributes to the functions of the process
Package	ORCV_CVAL_ENGINE	Function	PRE_PROCESS	Generate process id and populate process tables
		Function	PROCESS	Process Function: Populate process tables
		Function	POST_PROCESS	Post Process Function: Update process status and create record in Result Head Table according to parameters
		Function	EXECUTE_VALIDATION	Create process entry in ORCV_CVAL_PROCESS_CONTROL

Configuration

#	Step	Example
1	Create the validation functions into its module, respecting the function parameter types of the Custom Validation module	ORCV_SPIM_MGMT_ITEM_VAL.VALIDATE_HEADER
2	Create a group of validations into ORCV_CVAL_GROUP_HEAD	INSERT INTO ORCV_CVAL_GROUP_HEAD (GROUP_ID, GROUP_DESC, APP_ALIAS, ENTITY_ID, ATOMIC_IND) VALUES ('SPIM_ITEM_SUBMIT', 'ITEM SUBMIT VALIDATIONS', 'SPIM', 'ITEM', 'N');
3	Insert into ORCV_CVAL_GROUP_DETAIL all validations functions for the group	INSERT INTO ORCV_CVAL_GROUP_DETAIL (GROUP_ID, FUNCTION, ERROR_KEY, SEQ_NO, STATUS) VALUES ('SPIM_ITEM_SUBMIT', 'ORCV_SPIM_MGMT_ITEM_VAL.VALIDATE_HEADER', 'ORCV_SPIM_ERR.ERR\$SPIM_VALIDATE_HEADER', 1, 'A');

Custom Flex Attributes

Custom Flex Attribute Solution (CFAS) is a metadata-driven framework that enables setting up additional attributes on pre-enabled entities without modifying base code.

The entities enable with this framework are configured in tables with prefix ORCV_CFA.

Database Objects

This section describes the Database Objects.

Tables

Name	Description
ORCV_CFA_ENTITY	This table holds the Entities available on CFA.
ORCV_CFA_ENTITY_KEY	This table holds the columns by CFA Entity.
ORCV_CFA_ATTRIB_GROUP_SET	This table holds the attributes group set.
ORCV_CFA_ATTRIB_GROUP_SET_TL	This table holds the translated description for group set.
ORCV_CFA_ATTRIB_GROUP	This table holds the attributes group.
ORCV_CFA_ATTRIB_GROUP_TL	This table holds the labels for groups.
ORCV_CFA_ATTRIB	This table holds the attributes.
ORCV_CFA_ATTRIB_TL	This table holds the attributes labels.
ORCV_CFA_GEN_VIEW_GROUP	This table holds the group IDs to generate CFA views by batch ORCV_COMMON_CFA_GEN_JOB.

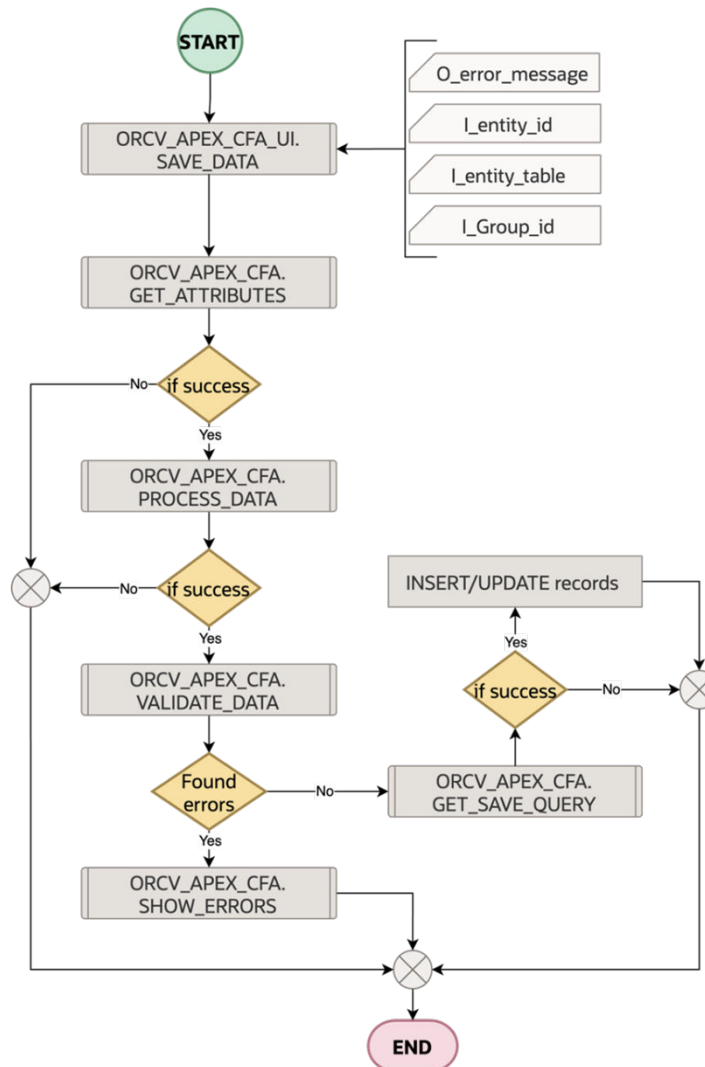
Packages

Name	Function	Description
ORCV_COMMON_CFA	GEN_VIEW_GRP	Generates views group according input parameters (I_entity_id, I_entity_table, I_group_set_id, I_group_id).
	GEN_VIEW_GRP_BY_GRP	Generates views group according input parameter (I_group_id). Calls GEN_VIEW_GRP with all parameters with NULL value except I_group_id.
ORCV_APEX_CFA_UI	SAVE_DATA	Saving data to table (staging or temporary table as defined on ORCV_CFA_ENTITY table). This function calls the functions in ORCV_APEX_CFA package.
ORCV_APEX_CFA	GET_ATTRIBUTES	Sets attributes table object based on GROUP_ID.
	PROCESS_DATA	Prepares data to be validated and saved to table (staging or temporary table as defined on ORCV_CFA_ENTITY table).
	VALIDATE_DATA	Validates if the data is consistent with the rules defined on ORCV_CFA_ATTRIBUTES table.
	GET_SAVE_QUERY	Gets one query to save data to table (staging or temporary table as defined on ORCV_CFA_ENTITY table).
ORCV_CVAL_CFA	VALIDATE	Function to validate CFAs data.

Name	Function	Description
ORCV_COMMON_CFA - GEN_BATCH	PRE_PROCESSOR	Insert into ORCV_THREAD_CONTROL table all groups to generate view, groups must be in table ORCV_CFA_GEN_VIEW_GROUP.
	PROCESSOR	Function to generate the Group View calling ORCV_COMMON_CFA.GEN_VIEW_GRP_BY_GRP

Process Overview

Figure 27-4 Custom Flex Attributes Process Overview



Configuration

The following is an example of a potential CFA configuration for Event Planning:

#	Step	Example
1	Add Entity	INSERT INTO orcv_cfa_entity (entity_id, entity_table, cfa_table, cfa_tmp_table, ext_entity_id) VALUES ('EVENT', 'PCEP_EVENT', 'ORCV_PCEP_EVENT_CFA', NULL, NULL);
2	Add Entity Key	INSERT INTO orcv_cfa_entity_key (entity_id, entity_table, key_col, key_number, data_type) VALUES ('EVENT', 'PCEP_EVENT', 'EVENT_ID', 1, 'NUMBER');
3	Add Attribute Group Set	INSERT INTO orcv_cfa_attrib_group_set (group_set_id, group_set_desc, entity_id, entity_table, display_seq) VALUES (ORCV_CFA_ATTRIB_GROUP_SET_SEQ.NEXTVAL, 'Group set – Event', 'EVENT', 'PCEP_EVENT', 1);
4	Add Attribute Group Set translation	INSERT INTO orcv_cfa_attrib_group_set_tl (group_id, lang, group_set_desc) VALUES (ORCV_CFA_ATTRIB_GROUP_SET_SEQ.CURRVAL, 1, 'Group set desc translated');
5	Add Attribute Group	INSERT INTO orcv_cfa_attrib_group (group_id, group_desc, group_set_id, group_default_id, group_view_name, display_seq) VALUES (ORCV_CFA_ATTRIB_GROUP_SEQ.NEXTVAL, 'Attrib Grupo #1 – Event', NULL, NULL, 'NB_CFA_PCEP_GROUP1', 1);
6	Add Attribute Group translation	INSERT INTO orcv_cfa_attrib_group_tl (group_id, lang, group_desc) VALUES (ORCV_CFA_ATTRIB_GROUP_SEQ.CURRVAL, 1, 'Group desc translated');
7	Add Attributes	INSERT INTO orcv_cfa_attrib (attrib_id, attrib_desc, group_id, view_col_name, storage_col_name, display_seq, data_type, ui_widget, code_type, default_value, enable_ind, value_req, maximum_length, lowest_allowed_value, highest_allowed_value) VALUES (ORCV_CFA_ATTRIB_SEQ.NEXTVAL, 'Attrib event 1', ORCV_CFA_ATTRIB_GROUP_SEQ.CURRVAL, 'attr event-1', 'VARCHAR2_1', 1, 'VARCHAR2', 'TI', NULL, 'Default-1', 'Y', 'Y', NULL, NULL, NULL);
8	Add Attributes translation	INSERT INTO orcv_cfa_attrib_tl (attrib_id, lang, attrib_desc) VALUES (ORCV_CFA_ATTRIB_SEQ.CURRVAL, 1, 'Attrib event 1 translated');

The tables defined for the CFA must respect the expected structure, so the framework is able to insert records to it.

Column	Type	Notes
PRIMARY_KEY_1 to 5	Same as ORCV_CFA_ENTITY_KEY table definition for the key.	Column names and types must match with ORCV_CFA_ENTITY_KEY configuration.
GROUP_ID	NUMBER(10)	This field will be populated with the GROUP_ID defined for the CFA.
VARCHAR2_1 to 10	VARCHAR2(250)	10 VARCHAR2 columns named VARCHAR2_1, VARCHAR2_2, VARCHAR2_N ... VARCHAR2_10.
NUMBER_11 to 20	NUMBER	10 NUMBER columns named NUMBER_11, NUMBER_12, NUMBER_N ... NUMBER_20.
DATE_21 to 25	DATE	5 DATE columns named DATE_21, DATE_22, DATE_N ... DATE_25.

After configuration and tables are created, it is required to execute the batch to generate the views required for the CFA. To do it is required to insert the new group ID to ORCV_CFA_GEN_VIEW_GROUP and execute batch

ORCV_COMMON_CFA_GEN_PROCESS_ADHOC. After the CFA views are generated for the groups in ORCV_CFA_GEN_VIEW_GROUP the entries are removed.

Other implementation Considerations

This section describes implementation considerations.

Custom Validations

It is possible to execute validations by using Custom Validations framework. Those are controlled in the application to which the CFA is being added to the entity. For the Custom Validations groups where those validations are required new entries should be added with the new functions to validate the CFAs values. More details are available in the Custom Validations chapter.

Example:

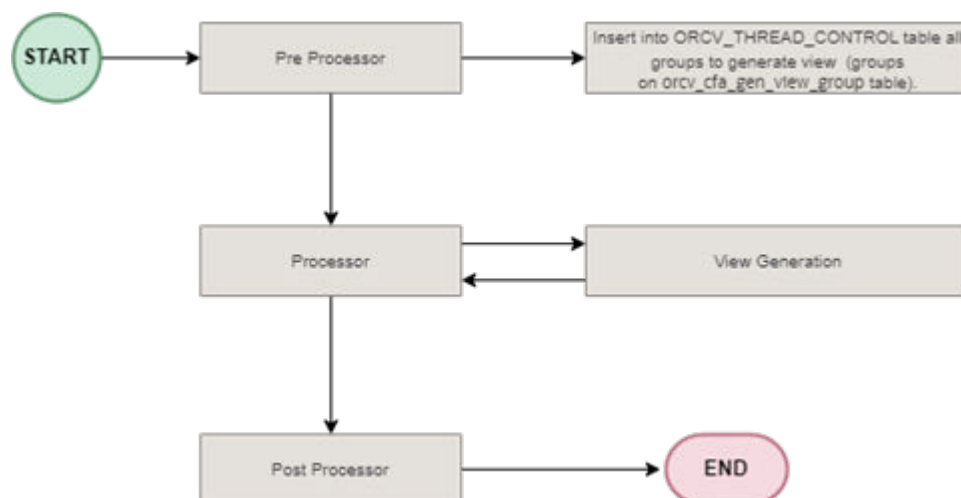
#	Step	Example
1	Define a CFA validation for Event Planning.	INSERT INTO orcv_cval_group_detail (group_id, function, error_key, seq_no, status) values ('PCEP_OFFER_WORKSHEET', 'ORCV_PCEP_CVAL_OFFER.VALIDATE_OFFER_CFA', 'ORCV_PCEP_CVAL_ERR.ERR\$PCEP_OV_VLD_OCFA', 26, 'A');

CFA Group View Generator Batch

The CFA Group View Generator Batch service creates the views for CFA Groups. Those views are required to maintain CFAs data. The batch process uses the table ORCV_CFA_GEN_VIEW_GROUP to find the views that are to be generated. When new attributes are created, changed, or delete from the attributes group it is necessary recreate the view.

- Job: ORCV_COMMON_CFA_GEN_JOB
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

Figure 27-5 CFA Group View Generator Batch Overview



Configuration

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_COMMON_CFA_GEN_JOB
MAX_CHUNK_SIZE	1
MAX_CONCURRENT_THREADS	1
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Operations

Frequency: Ad-hoc

Base URL: <BASE_URL>/ords/mfcs/orcv/common/<URL_PATH>

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
batch/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_COMMON_CFA_GEN_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG
batch/restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_COMMON_CFA_GEN_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG

URL PATH	REQUEST METHOD	REQUEST BODY	AFFECTED TABLES
batch/log	GET	URL query parameters: /log?jobName= ORCV_COMMON_CFA_GEN_JOB& executionId=<EXEC_ID>& logDate=YYYY-MM-DD	N/A
batch/kill	POST	{ "executionId": <EXEC_ID>, "jobName": "ORCV_COMMON_ CFA_GEN_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
batch/status	GET	URL query parameters: /status?jobName=ORCV_COMMON_ CFA_GEN_JOB &executionId=<EXEC_ID>	N/A
/ping	GET	Empty	N/A

Rules Engine Framework

This section describes the Rules Engine Framework.

Process Overview

Rule Engine framework acts as an engine for rule execution, allows rules to be configured in either batch or function mode and execute them via a single point of entry web service.

Rules defined as batch mode are executed using the Batch Engine Framework.

The web service requests require the rule key identifier in parameter RULE_KEY in the request payload.

Database Objects

Object Type	Name	Sub Object Type	Sub Object Name	Description
Table	ORCV_RLEG_RULE	NA	NA	This table holds the rule definition. The column program name is a concatenation of ORCV_RLEG_ and rule key in upper case, it will represent the reference to the table ORCV_BATCH_CONFIG in case the rule mode is batch.
Table	ORCV_RLEG_FUNCTION	NA	NA	This table contains the functions list associated to the rule.

Object Type	Name	Sub Object Type	Sub Object Name	Description
Table	ORCV_RLEG_PARAMETERS	NA	NA	This table contains the parameters associated to the rule.
Table	ORCV_RLEG_ATTR	NA	NA	This table contains the attributes associated to the rule.
Table	ORCV_RLEG_MAP_ATTR_HEAD	NA	NA	This table contains the mapping of the columns that will be displayed in the Rule Attributes dynamic grid.
Table	ORCV_RLEG_MAP_ATTR_DETAIL	NA	NA	This table contains details of the mapping of the columns that will be displayed in the Rules Attributes dynamic grid.
Package	ORCV_RLEG_ORDS	Function	WRP_JSON_TO_STG_OBJECT	Deconstructs the body CLOB from the web service request and retrieves the rule key parameter.
		Function	BATCH_START_ORDS	Calls WRP_JSON_TO_STG_OBJ and, if the rule mode is batch, calls the batch engine framework with the rule program name as the batch to be executed.
		Function	BATCH_RESTART_ORDS	Calls WRP_JSON_TO_STG_OBJ and in case the rule mode is batch calls the batch engine framework with the rule program name as the batch to be executed and the execution id to be restarted.

Configuration

#	Step	Example
1	Insert rule key into ORCV_RLEG_RULE	INSERT INTO ORCV_RLEG_RULE (ROW_ID, RULE_KEY, RULE_NAME, APP_ALIAS, RULE_MODE, STATUS) VALUES (ORCV_RLEG_RULE_SEQ.NEXTVAL, 'SPIM_ITEM_EVT_PND', 'ORCV_RLEG_SPIM_EVENT_ITEM_JOB', 'SPIM', 'B', 'A');
2	Insert the mapping of the columns into ORCV_RLEG_MAP_ATTR_HEAD	INSERT INTO ORCV_RLEG_MAP_ATTR_HEAD (RULE_KEY, VIEW_NAME, VALIDATION_FUNC) VALUES ('SPPY_QA_DEFECT', 'ORCV_V_SPPY_RLEG_QA_DEFECT', 'ORCV_SPPY_RLEG_RULE_ATTR_VAL.SUPP_OR_PO_TYPE IS NOT NULL');
3	Insert details of the mapping of the columns into ORCV_RLEG_MAP_ATTR_DETAIL	INSERT INTO ORCV_RLEG_MAP_ATTR_DETAIL (RULE_KEY, SEQ, COLUMN_ATTR, LABEL_VIEW, LABEL_APEX, TYPE, LIST_VALS, REQUIRED, DEFAULT_VALUE, MAX_SIZE) VALUES ('SPPY_QA_DEFECT', 1, 'NUMBER_1', 'SUPPLIER', 'Supplier', 'SELECTLIST', 'ORCV_V_SPPY_RLEG_SUPP_TL', 0, NULL, NULL);

#	Step	Example
4	Insert batch engine configuration for the rule key program name into ORCV_BATCH_ENGINE	INSERT INTO ORCV_BATCH_ENGINE (PROGRAM_NAME, SEQ_NO, PRE_PROCESSOR, PROCESSOR, POST_PROCESSOR, ENABLED, PURGE_PROGRAM) VALUES ('ORCV_RLEG_SPIM_ITEM_EVT_PND', 1, 'ORCV_SPIM_EVENT_ITEM_BATCH.PRE_PROCESSOR', 'ORCV_SPIM_EVENT_ITEM_BATCH.PROCESSOR', NULL, 'Y', 'N');
5	Insert batch configuration for the rule key program name into ORCV_BATCH_CONFIG	INSERT INTO ORCV_BATCH_CONFIG (PROGRAM_NAME, MODULE_HIER_ID, MAX_CHUNK_SIZE, MAX_CONCURRENT_THREADS, RETRY_LOCK_ATTEMPTS, RETRY_WAIT_TIME, RETRY_MAX_ATTEMPTS_COUNT, RETRY_ORDERED, RETRY_IND) VALUES ('ORCV_RLEG_SPIM_ITEM_EVT_PND', ORCV_COMMON_UTILS.GET_MODULE_HIER_ID(I_module_id => 'ITEMS_MANAGEMENT', I_sub_module_id_1 => 'EVENTS'), 1000, 4, 3, 3, 3, 'N', 'N');

Web Service

Authorization Method: OAuth 2.0

Base URL: <BASE_URL>/ords/mfcs/orcv/ruleengine/rules/<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
batch/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_RLEG_JOB", "agentExecutionId": "", "parameters": "RULE_KEY=<RULE_KEY>" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG
batch/restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_RLEG_JOB", "agentExecutionId": "", "parameters": "RULE_KEY=<RULE_KEY>" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG

URL Path	Request Method	Request Body	Affected Tables
batch/log	GET	URL query parameters: /log?jobName=ORCV_RLEG_JOB &executionId=<EXEC_ID>&logDate=YYYY-MM-DD	N/A
batch/kill	POST	{ "executionId": <EXEC_ID>, "jobName": "ORCV_RLEG_JOB", }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG
batch /status	GET	URL query parameters: /status?jobName=ORCV_RLEG_JOB &executionId=<EXEC_ID>	N/A
ping	GET	Empty	N/A

Profile

This section describes the Profile application.

Implementation Overview

This section provides an Implementation Overview.

Functionality

The Profile application will be used to manage existing and new users. Using this application an administrator can associate a user with a department and assign user permissions to the applications.

Retail Supply Chain Hub functional security supports a role-based, declarative model where resources are protected by roles that are assigned to users. In this manner, an application role becomes the container that grants permissions to its members to access the application tasks, screens, and functionalities within.

Roles, also referred to as Job Roles, align with the application's functionalities. Roles are used to classifying users based on job responsibilities and actions to be performed in the application. When a user logs into the application, based on the roles assigned to the user, the system determines which roles have been granted to the user, and the system features are enabled accordingly.

The roles definition follows a hierarchy permission type definition from lower to high priority inheriting the previous definition:

1. Access to the Application
2. No Access to Entity
3. View Only
4. Edit Access
5. Approval

For all the applications a denomination will be followed considering the application alias, the entity, and the permission type definition. For example, for Procure to Pay, the application alias is SPPP, one of the entities supported in Purchase Orders with ID ORDER, and applying the hierarchy permission type definition logic is:

1. ORCV_SPPP_APP_JOB – user with access to the application
2. ORCV_SPPP_ORDER_NA_JOB – the user doesn't have access to entity Purchase Orders in Procure to Pay
3. ORCV_SPPP_ORDER_VIEW_JOB – user can view purchase order details in Procure to Pay.
4. ORCV_SPPP_ORDER_EDIT_JOB – user can view and edit purchase order details in Procure to Pay.
5. ORCV_SPPP_ORDER_APPR_JOB – user can view, edit, and approve purchase orders in Procure to Pay.

Configuration

This section describes the Profile configurations.

Security

To have access to the applications it is required to assign roles to users, Profile application requires the user to be created in Supplier Evaluation with the required permissions.

The application level security is defined in table ORCV_SEC_APP_ROLES.

Column Name	Description	Example
USER_ID	Unique identifier for the user	JOHN.DOE@CUSTOMER.COM
APP_ALIAS	Unique identifier for application alias.	SPCH
ENTITY_ID	This field contains entity ID	NA
PERMISSION_TYPE	This field contains the user permissions to applications	APP
NOTIFICATION_IND	This field contains if notifications are active: (Y)es,(N)o	N
ROLE_NAME	This field contains user role	ORCV_SPCH_APP_JOB

Once the user is created in Supplier Evaluation with the required permissions the user will be assigned to one of the 4 Application Roles:

- RETAILER_ADMINISTRATOR: Role name that determines user type is retailer and admin.
- RETAILER_USER: Role name that determines user type is retailer.
- SUPPLIER_ADMINISTRATOR: Role name that determines user type is supplier and admin.
- SUPPLIER_USER: Role name that determines user type is supplier.

Any record that is being mastered in Supplier Evaluation is marked as read-only in the Profile application and even an administrator user type will not be able to change this information, only in Supplier Evaluation or IDCS. The access to the multiple modules in Retail Supply Chain Hub is managed in Profile.

The admin permission allows a retailer the access to Profile application in admin mode which allows the configuration of new suppliers and define the access level for each user and entity. If the user is a supplier user, then in Profile the user can define the permissions of each supplier user to their own suppliers.

Application Configuration

This section describes Application Configurations.

Application Options

Some applications require options to be configured at different levels, Global applied to the complete application and Supplier Defaults to define the default configuration of suppliers.

Those configurations can be defined in the ORCV_PRFL_CFG_APP_OPTIONS (more details can be found in the Data Model document).

Column Name	Description
APP_ALIAS	The App Alias of the application to be configured
ATTRIBUTE_ID	Attribute identifier
ATTRIBUTE_DESC	Attribute description
ATTRIBUTE_VALUE	Value of the attribute
CFG_LEVEL	Configuration level: (G)lobal, (S)upplier Defaults, (C)ustomers Defaults

Supplier Configurations

Configurations for Suppliers at the application level are also required and those can be defined in two tables ORCV_PRFL_CFG_APP_SUPP and ORCV_PRFL_CFG_APP_SUPP_OPTIONS, the first defines the applications configured for suppliers, and the second the details of the options for the suppliers.

Association between suppliers and applications is defined in ORCV_PRFL_CFG_APP_SUPP (more details can be found in Retail Supply Chain Hub Data Model document).

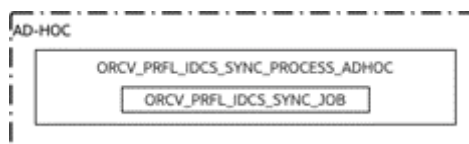
Column Name	Description
SUPPLIER	Supplier ID
APP_ALIAS	Application Alias

Application Supplier Options are defined in ORCV_PRFL_CFG_APP_SUPP_OPTIONS (more details can be found in the Data Model document).

Column Name	Description
SUPPLIER	Supplier ID
APP_ALIAS	Application Alias
ATTRIBUTE_ID	Attribute identifier, aligned with ORCV_PRFL_CFG_APP_OPTIONS
ATTRIBUTE_VALUE	Value of the attribute

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in MCEFIO.

Figure 27-6 Profile Batch Schedule Overview

Other Implementation Considerations

This section provides additional Implementation Considerations.

Specific Batch Configuration

This section describes specific batch configurations.

IDCS Synchronization Batch

Synchronizes users from IDCS to Profile configuration tables ORCV_SEC_USER and ORCV_SEC_USER_APP. It will create non existing users and inactivate the removed users from IDCS. Only users with the roles identified in the SECURITY System Parameters for parameters RETAILER_ADMIN_ROLE, RETAILER_ROLE, SUPPLIER_ADMIN_ROLE and SUPPLIER_ROLE will be considered and only one can be assigned to each individual user otherwise the batch will return an error. Existing entries identified with ORIGIN different from IDCS will not be changed by the batch.

- Job: ORCV_PRFL_IDCS_SYNC_JOB
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_PRFL_IDCS_SYNC_JOB
MAX_CHUNK_SIZE	1
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types available.

The codes are defined in the ORCV_CODE_HEAD and ORCV_CODE_DETAIL tables.

Code Type	Description
PRSS	Profile Supplier Status
PRSU	Profile Supplier Undefined
USPM	User Permission
USST	User Status

Operations

This section describes available operations.

Batch by Module

This section describes the batch by module.

IDCS Synchronization Batch

Authentication Method: OAuth2

BaseURL: <BASE_URL>/ords/mfcs/orcv/profile/batch/idcssync/batch/<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_PRFL_IDCS_SYNC_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SEC_USER ORCV_SEC_USER_APP ORCV_SEC_USER_APP_TMP ORCV_SEC_USER_TMP
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_PRFL_IDCS_SYNC_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SEC_USER ORCV_SEC_USER_APP ORCV_SEC_USER_APP_TMP ORCV_SEC_USER_TMP
log	GET	URL query parameters: /log? jobName=ORCV_PRFL_IDCS_SYNC_JOB &executionId=<EXEC_ID>&logDate=YYYY- DD-MM	N/A

URL Path	Request Method	Request Body	Affected Tables
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_PRFL_IDCS_SYNC_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SEC_USER ORCV_SEC_USER_APP ORCV_SEC_USER_APP_TMP ORCV_SEC_USER_TMP
status	GET	URL query parameters: /status? jobName=ORCV_PRFL_IDCS_SYNC_JOB &executionId=<EXEC_ID>&logDate=YYYY- DD-MM	N/A
status/all	GET	URL query parameters: /status?<EXEC_IDs>	N/A
config	GET	Empty	N/A

Configuration Management

The configuration management application is an aggregator of configuration pages links for different applications. Depending on the roles assigned to a user different application configuration links can be visible to the user.

Configuration management setup is configurable in the table ORCV_CFGM_CONFIG_LINK, and by default has the following configuration:

Column Name	Example	Description
LINK_ID	RLEG_SPCH	This field contains the key uniquely identifying each row on the table.
LINK_TITLE	Rule Engine Configuration	This field contains the link configuration label.
LINK_TEXT	NA	This field contains the link configuration description.
APP_ID	100801010	This field contains the application id of the configuration link.
APP_ALIAS	RLEG	This field contains the application alias.
PAGE_NO	100	This field contains the default page number of the configuration link.
ICON_IMG	fa fa-server-play	This field contains the icon image
PARENT_APP_ALIAS	SPCH	This field contains the hub alias in which this configuration link entry is displayed.

Inventory

Inventory is a module available in Retail Supply Chain Hub. This chapter describes the implementation considerations and the operation details for this application.

Functionalities

Supply Chain Hub works as the entry point to a group of applications that combines retailers and suppliers in the same portal. Depending on which applications are available in the Hub the users will be presented with a summary in the homepage and according to the user type (supplier or retailer) will have visibility on Operational News and Queries.

Operational News is a functionality that allows the retailers to share news with targeted suppliers in a central place, the retailers have permissions to create and edit them while the suppliers can only view them.

Another functionality is Queries, this functionality aims to allow the dialog between retailers and suppliers regarding a specific entity, like a Purchase Order or Shipment. This way the dialog is focused on that specific entity and makes easier the context.

Working as true Hub, Supply Chain Hub adapts to the applications that are installed in the system not only with different charts being presented in the homepage but also with a menu with access to all those applications so switching between applications keeps the context of the user in a seamless experience.

Configuration

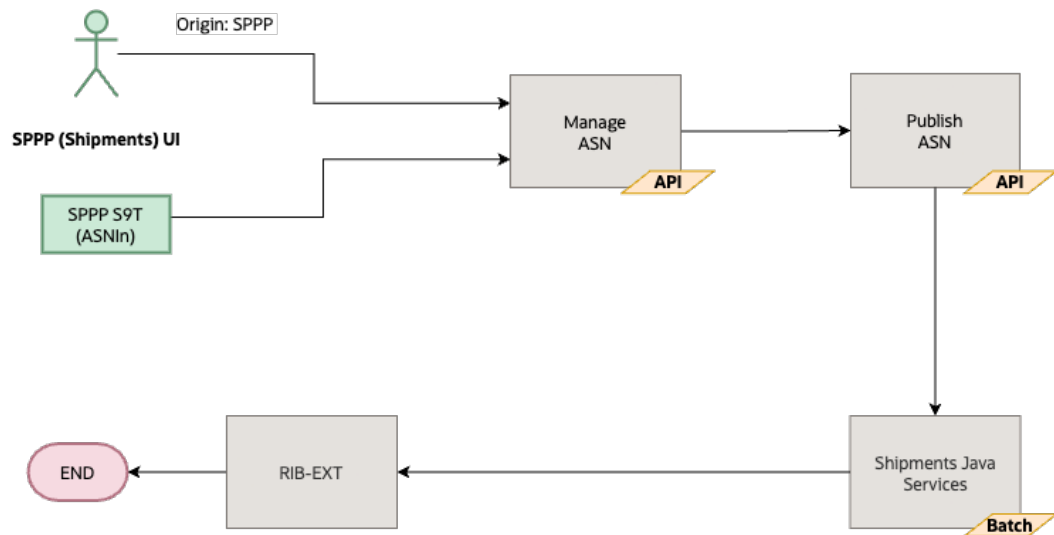
This application has multiple configurations to support its functionalities and in the subsections below those configurations are highlighted.

POM Configuration

Batch scheduler details with all batches, dependencies and frequencies can be downloaded following the steps in MCEFIO.

Figure 27-7 Inventory POM Schedule

SPPP - Shipments - Integration Overview



Other Implementation Considerations

This section describes additional Implementation Considerations.

Materialized Views

This application requires some results to be materialized in tables using the Materialized View Framework described in MCEFIO and below is the list for this application.

ORCV_MV_INV_IF_TRAN_DATA

Description: Creates a snapshot of MFCS table IF_TRAN_DATA into ORCV_MV_INV_IF_TRAN_DATA.

Pre-Dependencies: MFCS.SALSTAGE_JOB

Post-Dependencies: ORCV_INV_DAILY_HIST_JOB

Frequency: Nightly

ORCV_MV_INV_ITEM_LOC_SOH

Description: Creates a snapshot of MFCS table ITEM_LOC_SOH into ORCV_MV_INV_ITEM_LOC_SOH at transactional level and levels above.

Pre-Dependencies: None

Post-Dependencies: None

Frequency: Nightly

Specific Batch Configuration

This section describes specific batch configurations.

Daily Item Loc History

Loads history table ORCV_INV_DAILY_IL_HIST with data from ORCV_MV_INV_IF_TRAN_DATA snapshot with currencies converted to the local currency aggregated by day, item, location, transaction type and disc type.

- Job: ORCV_INV_DAILY_IL_HIST_JOB
- Frequency: Nightly
- Dependency: None
- Inter Scheduler Dependency: None

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_INV_DAILY_IL_HIST_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Weekly Item Loc History

Loads history table ORCV_INV_WEEKLY_IL_HIST with data from ORCV_INV_DAILY_IL_HIST aggregated by end of weekday, item, location, transaction type and disc type.

- Job: ORCV_INV_WEEKLY_IL_HIST_JOB
- Frequency: Nightly
- Dependency: ORCV_INV_DAILY_IL_HIST_JOB
- Inter Scheduler Dependency: None

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_INV_WEEKLY_IL_HIST
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N

Column Name	Default Values
RETRY_IND	N

Operations

This section describes operations.

Materialized Views Refresh Details

This section describes Materialized Views Refresh Details.

ORCV_MV_INV_IF_TRAN_DATA

Column	Description	Value
Result Synonym	Name of the synonym pointing to the result table.	ORCV_INV_IF_TRAN_DATA_A_R
Result Table A	Name of _A result table.	ORCV_INV_IF_TRAN_DATA_B_R
Result Table B	Name of _B result table.	ORCV_V_INV_IF_TRAN_DATA_R
Source View	The View that contains business logic to be used to populate the result table.	ORCV_V_INV_IF_TRAN_DATA_R C
Driving View	This field contains the view to distribute the refresh process.	ORCV_COMMON_MV_REFRESH. COMMON_PRE_PROCESSOR
Pre-Processor	Name of the pre-process procedure if applicable.	ORCV_INV_MV. CHECK_TRAN_DATA
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	LOCATION

ORCV_MV_INV_ITEM_LOC_SOH

Column	Description	Value
Result Synonym	Name of the synonym pointing to the result table.	ORCV_INV_ITEM_LOC_SOH_A_R
Result Table A	Name of _A result table.	ORCV_INV_ITEM_LOC_SOH_B_R
Result Table B	Name of _B result table.	ORCV_V_INV_ITEM_LOC_SOH_R
Source View	The View that contains business logic to be used to populate the result table.	ORCV_V_INV_ITEM_LOC_SOH_RC

Column	Description	Value
Driving View	This field contains the view to distribute the refresh process.	ORCV_COMMON_MV_REFRESH. COMMON_PRE_PROCESSOR
Pre-Processor	Name of the pre-process procedure if applicable.	NA
Check Function	Name of the check function to validate if there is a need to refresh or if there are any batch dependencies.	NA
Preserve Data	Flag to decide if the data needs to be preserved until the next run cycle.	Y
Restart Indicator	Flag to indicate if to allow restart batch processor or not.	N
Max Concurrent Threads	This field contains the max_concurrent_threads that created the record.	4
Filter Fields	Column name of the view to be used for threading.	LOC

Batch by Module

This section describes batches by module.

Daily Item Loc History

Authentication Method: OAuth2

BaseURL: <BASE_URL>/ords/mfcs/orcv/merch/inventory/dailyilhist/URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": " ORCV_INV_DAILY_IL_HIST_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_INV_IF_TRAN_DATA_CNVCURR ORCV_INV_DAILY_IL_HIST

URL Path	Request Method	Request Body	Affected Tables
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_INV_DAILY_IL_HIST_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_INV_IF_TRAN_DATA_CNVCURR ORCV_INV_DAILY_IL_HIST
log	GET	URL query parameters: /log?jobName=ORCV_INV_DAILY_IL_HIST_JOB &executionId=<EXEC_ID>&logDate= YYYY-MM-DD	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_INV_DAILY_IL_HIST_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_INV_IF_TRAN_DATA_CNVCURR ORCV_INV_DAILY_IL_HIST
status	GET	URL query parameters: /status? jobName=ORCV_INV_DAILY_IL_HIST_JOB&executionId=<EXEC_ID> & logDate=YYYY-MM-DD	N/A
status/all	GET	URL query parameters: /status?<EXEC_IDS>	N/A
config	GET	Empty	N/A

Weekly Item Loc History

Authentication Method: OAuth2

BaseURL: <BASE_URL>/ords/mfcs/orcv/merch/inventory/wklyilhist/URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_INV_WKLY_IL_HIST_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_INV_WEEKLY_IL_HIST
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_INV_WKLY_IL_HIST_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_INV_WEEKLY_IL_HIST
log	GET	URL query parameters: /log?jobName=ORCV_INV_WKLY_IL_HIST_JOB &executionId=<EXEC_ID>&logDate=YYYY-MM-DD	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_INV_WKLY_IL_HIST_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_INV_WEEKLY_IL_HIST
status	GET	URL query parameters: /status?jobName=ORCV_INV_WKLY_IL_HIST_JOB &executionId=<EXEC_ID>&logDate=YYYY-MM-DD	N/A

URL Path	Request Method	Request Body	Affected Tables
status/all	GET	URL query parameters: /status?<EXEC_IDs>	N/A
config	GET	Empty	N/A

Queries

This section describes the common functionality and configurations available for Queries.

Functionalities

Queries aims to allow the dialog between retailers and suppliers regarding a specific entity, like a Purchase Order or Shipments. Therefore, it is possible for multiple users to discuss topics about a particular entity in its own context.

Configuration

This application has multiple configurations to support its functionalities and, in the subsections, below, those configurations are highlighted.

Application Options

Application Options core functionality is detailed in Profile, Queries has the following Application Options defined.

The application options are defined in table ORCV_PRFL_CFG_APP_OPTIONS.

App Alias	Attribute	Description	Default Value	Configuration Level
QUERY	QUERY_CUT_OFF	Number of days post closure, a query is cut off from the Queries Screen	14	Global

Security

The application security is also described in Profile, the permissions defined for this application are defined below.

The security is defined in table ORCV_SEC_APP_ROLES.

App Alias	Entity	User Type	Permission Type	Role Name	Role Description
QUERY	NA	NA	APP	ORCV_QUERY_APP_JOB	Query Access

System Parameters

Listed below is the list of configurations at the system level that are used by this application. Those configurations include configurations specific to the module.

The parameters are defined in table ORCV_SYSTEM_PARAMETERS.

Area	Parameter	Description	Value 1	Value 2
CONFIG	QUERY_EXIST_CLASS	Class used in the APEX button where there are new updates to the queries for the user.	t-Button--danger	-
CONFIG	QUERY_N_EXIST_CLASS	Class used in the APEX button where there are no new updates to the queries for the user.	t-Button--hot	-
PURGE	PURGE_DAYS	Default purge Period.	ORCV_QUERY_HEAD	120
QUERY	AUTO_CLS_QUERY_DELAY	Number of days to automatically close Queries.	7	-

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types and the codes available, the sequence and their description.

The codes are defined in tables ORCV_CODE_HEAD and RCV_CODE_DETAIL.

Code Type	Description
QMST	Query Message Status
QRST	Query Status Description

Batch WebServices

This section describes the list of batch jobs used by this application.

ORCV_QUERY_MAINTENANCE_JOB

Description: Query Maintenance Job closes queries without updates for a certain number of days defined in system parameter AUTO_CLS_QUERY_DELAY. This batch job is based on the Batch Framework detailed in this document.

Pre-Dependencies: None

Post-Dependencies: None

Web Services:

- Authorization Method: OAuth 2.0
- Frequency: Ad-hoc
- URL: <BASE_URL>/ords/mfcs/orcv/query/maintenance/<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
batch/start	POST	Empty	ORCV_QUERY_HEAD ORCV_QUERY_DETAIL

URL Path	Request Method	Request Body	Affected Tables
batch/restart	POST	{ "executionId": <EXEC_ID> }	ORCV_QUERY_HEAD ORCV_QUERY_DETAIL
batch/log	GET	URL query parameters: /log? jobName=ORCV_QUERY_MAINTENANCE_JOB &executionId=<EXEC_ID>&logDate=YYYY-MM-DD	N/A
batch/kill	POST	{ "executionId": <EXEC_ID> }	N/A
batch/status	GET	URL query parameters: /status? jobName=ORCV_QUERY_MAINTENANCE_JOB &executionId=<EXEC_ID>&logDate=YYYY-MM-DD	N/A
/ping	GET	Empty	N/A

Upload and Download Framework

This section describes the common functionality, configurations, and process overview available for Upload Download Framework.

Functionalities

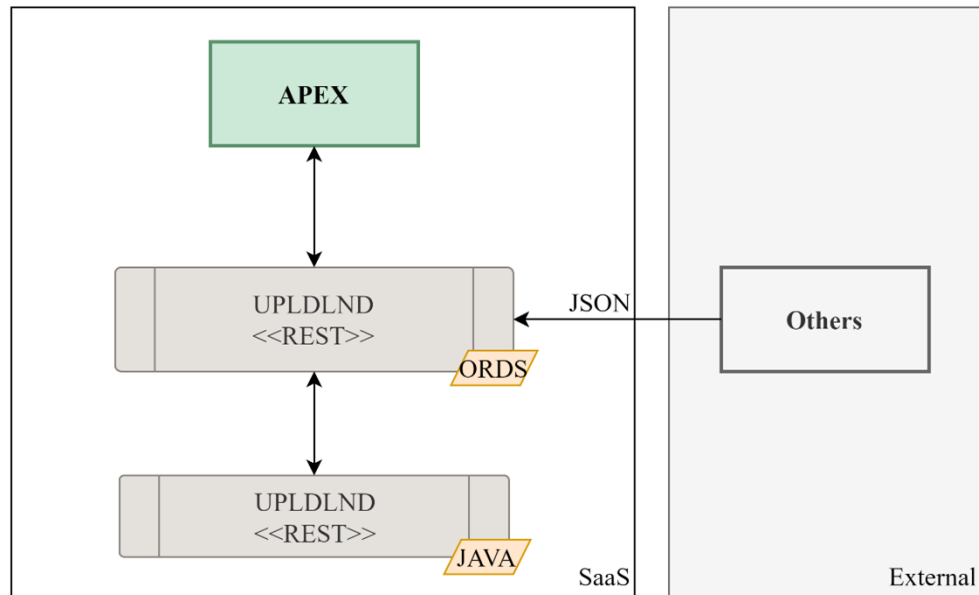
Upload Download Framework adds to Retail Supply Chain Hub a common core functionality to process upload and download of files that can be used by multiple modules in applications and integrations.

The framework includes the Upload Download application to have direct access to all the templates available for the user so they can download a blank template and upload a new file based on those blank templates. It is also possible to check the results of the upload where the user can validate if it was processed successfully or if any error occurred with the details of the failure. Other applications import the Upload Download application which allows the upload of files using the framework in the context of the application itself.

Note: The column SEQ_NO_MANDATORY in ORCV_S9T_WKSHT table defines which lines in the uploaded file must be processed. In case the value is defined the Upload Download Framework checks the value against SEQ_NO in ORCV_S9T_WKSHT_COLS table and if the line does not have value for the column the line is discarded and not processed. In case SEQ_NO_MANDATORY is NULL all lines are processed.

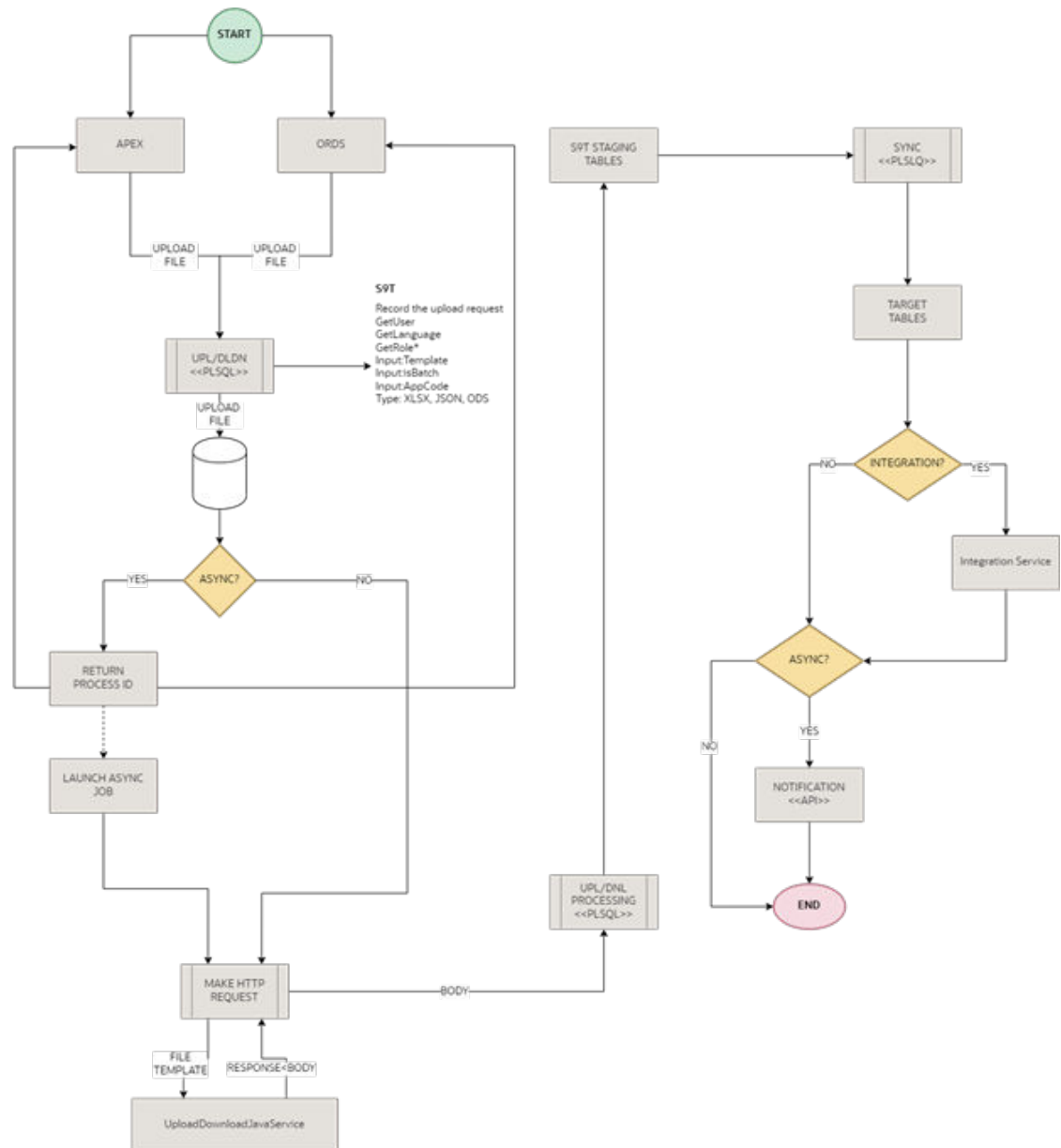
Architecture

Figure 27-8 Upload and Download Framework Architecture Overview



Process Overview

Figure 27-9 Upload/Download Process Overview



Upload

From an upload perspective, the information being uploaded follows the defined template structure accordingly and will be processed synchronously or asynchronously based on the parameter being passed in the REST service call payload.

- The upload can be initiated by APEX, or any external entity.

- It can be configured for each template and file type in column AVAILABLE_FOR of table ORCV_S9T_TEMPLATE_FILE_TYPE that indicates whether it is available for APEX, ORDS, or ALL.
- To start the upload, it will be required at least to pass:
 - The template type
 - The corresponding action type (in this case Upload)
 - The processing type: ASYNC or SYNC
 - What kind of file type: XLSX / JSON
 - And correspondent file object
- The Java service will parse the request and check the required template format for the operation and accordingly will stage that information on the database using ORDS.
- Depending on the operation type the return to the user can be just the process identifier, for asynchronous operations or the information if it was successful or not the upload in synchronous operation.
- Additional to the normal upload process, this can trigger an integration that can result in a corresponding notification to the user once all the records or successfully integrated or not.

 Note:

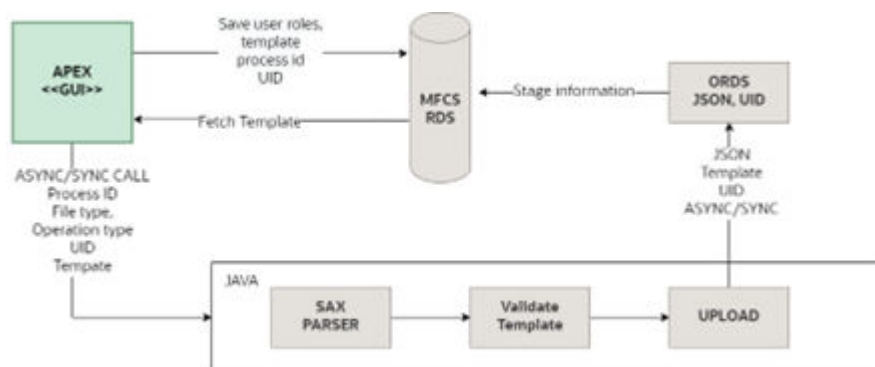
Upload Download Framework has a size limit of 20mb per compressed file and files larger than that might not be possible to process.

APEX

Considering that we do not have a continuous flow of authorization between APEX/Java/ORDS it will be required to register the request from APEX associated with a specific process on Java so that once the ORDS is invoked we can understand who the owner of this upload process is and enable data filter policy at database side.

During the request to ORDS, the process identifier will be handover to ensure that the correct database context will be injected during the upload process.

Figure 27-10 Upload and Download Framework Oracle APEX Upload Implementation Overview



1. User Login on Apex application.
2. The user session and roles will be injected using the security framework.
3. APEX prehook sets the database context for the user.
4. The user specifies the template and uploads the file.
5. The S9T process table will keep the information about the user and session.
6. During the process, the S9T process will check the information about the process identifier that was received and will inject the context into the database.
7. With this information the process will check what user permissions are; for example, if the user has access to supplier information or if can upload the required information.

JSON Upload

The upload in JSON follows the same process flow as the upload in XLSX from APEX UI but is only available as a ReST web service, the request payload structure is based on the template key defined on ORCV_S9T_TEMPLATE, ORCV_S9T_WKSHT, and ORCV_S9T_WKSHT_COLS tables.

For a template to be enabled for JSON upload, the template key needs to have a record in table ORCV_S9T_TEMPLATE_FILE_TYPE where file_type is JSON.

The upload with JSON implements Client Credentials security flows described on chapter 'Oracle REST Data Services (ORDS) and Application Express (APEX)'.

Configuration requirements for request with Client Credential:

- IDCS Application scope mapped to the appropriate S9T role from ORCV_S9T_TEMPLATE_EVENT_ROLE for the template upload action on ORCV_SEC_ROLES_SCOPES.
- User created on IDCS with the name matching the IDCS application client ID used to request the token.
- User configured on Profile application with the appropriate roles and security level permission for the actions that want to be performed via JSON upload.

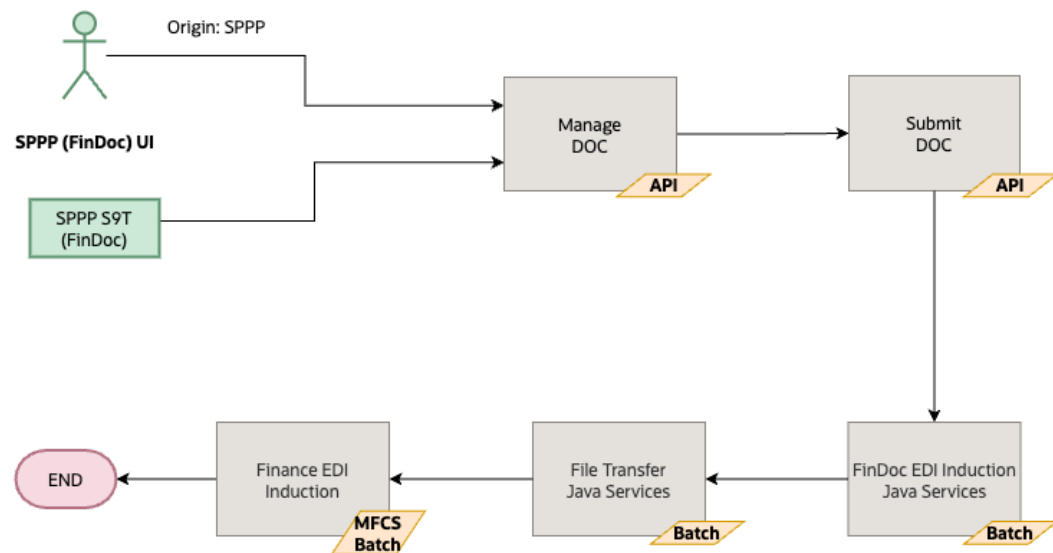
On upload as JSON, the columns keys defined for the template key on ORCV_S9T_WKSHT_COLS are mapped on the upload payload as attributes with any underscore character removed and the next character after it in uppercase, transformed to camel case.

Download

From a download perspective, it will be possible to download a blank template in the format required or with data accordingly to data restrictions and filter criteria.

Figure 27-11 Download Flow

SPPP - Finance Documents - Integration Overview



1. User Login on Apex application.
2. The user session and roles will be injected using the security framework.
3. APEX prehook sets the database context for the user.
4. The user specifies the template and uploads the file.
5. APEX request the download through a database call and not using the UPLDLND Java API.
6. The database will call the same PL\SQL method has exposed by ORDS.
7. Applying data filter policy and returning the corresponding JSON.
8. Get the credentials configured on APEX to do a call to the Java service.
9. Invokes the Java service with the JSON output from the database call.
10. The Java service based on JSON will do the transformation to the file type requested.

ORDS Download Processing Flow

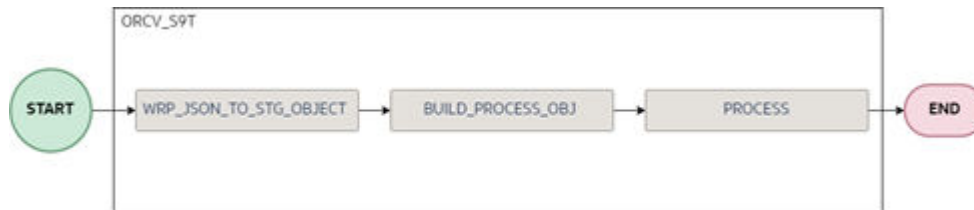
After the Java Service sends the user information and file template, the ORDS service will try to complement this information to process the download.

The first step will be to use a function called `WRP_JSON_TO_STG_OBJECT`, which receives a JSON payload as the response from the Java service. This function will parse the JSON and retrieve information about the user, filter criteria, and template file.

Then, ORDS will use another function, `BUILD_PROCESS_OBJ`, to get more information about the template file structure such as how many worksheets and columns are part of this template.

Finally, to proceed to the action of Download, ORDS will call a generic function, `PROCESS`, responsible for validating the template file structure and will call another function responsible to perform the action.

Figure 27-12 ORDS Service Download Flow



Web Service

This section describes the web service.

ORDS

Authorization Method: OAuth 2.0

Base URL: <BASE_URL>/ords/mfcs/orcv/upldnld/<URL_PATH>

Excel Format

URL Path	Request Method	Request Body
/downloadui	POST	{ "fileType": "XLSX", "downloadType": "D", "filters": [{ "filterValues": ["FILTER1, FILTER2"] }], "templateKey": "TEMPLATE_KEY", "appCode": "APP_CODE" }

JSON Format

URL Path	Request Method	Request Body
/upload	POST	<pre>{ "templateKey": "SPIM_ITEM_DATA", "appCode": "SPIM", "isAsync": "Y", "fileType": "JSON", "fileName": "example.json", "file": { "data": { "userLang": "1", "worksheets": [{ "wkshtKey": "ITEM_HEADER", "lines": [{ "attributes": { "action": "NEW", "item": "1", "itemType": "SKU", "vpn": null, "supplier": 6200, "status": "A", "vendorDesc": "TestSKULC_MultiSupps", "itemDesc": "Example", "inventoryInd": "Y", "orderableInd": "Y", "sellableInd": "Y", "diff1": null, "diff2": null, "diff3": null, "diff4": null, "dept": 3081, "class": 2, "subclass": 1, "brandName": null, "seasonPhase": null, "onSaleDate": null, "offSaleDate": null, "stockAvailable": 2, "rrp": null, "ref1": null, "ref2": null, "ref3": null, "ref4": null, "ref5": null } }] }] } } }</pre>

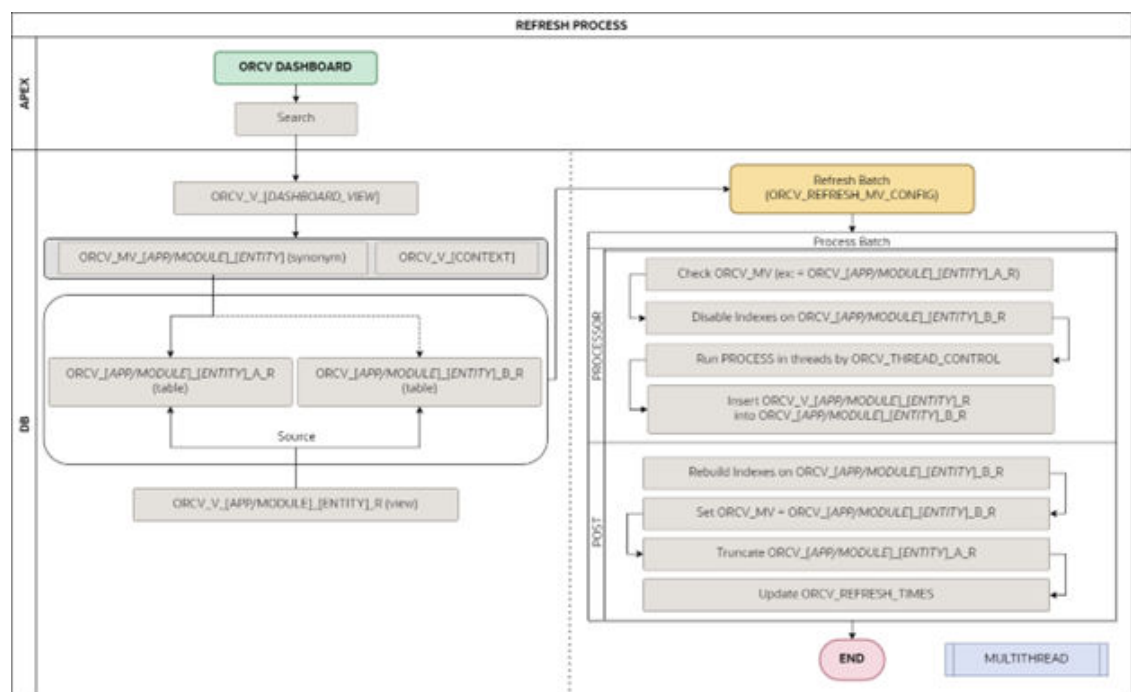
URL Path	Request Method	Request Body
		} }] }] } } }

Materialized View Framework

The Materialized View Framework will allow a configurable approach to create, maintain, and refresh data on tables based on the principle of materialized views.

Process Overview

Figure 27-13 Materialized View Framework Implementation Process Overview

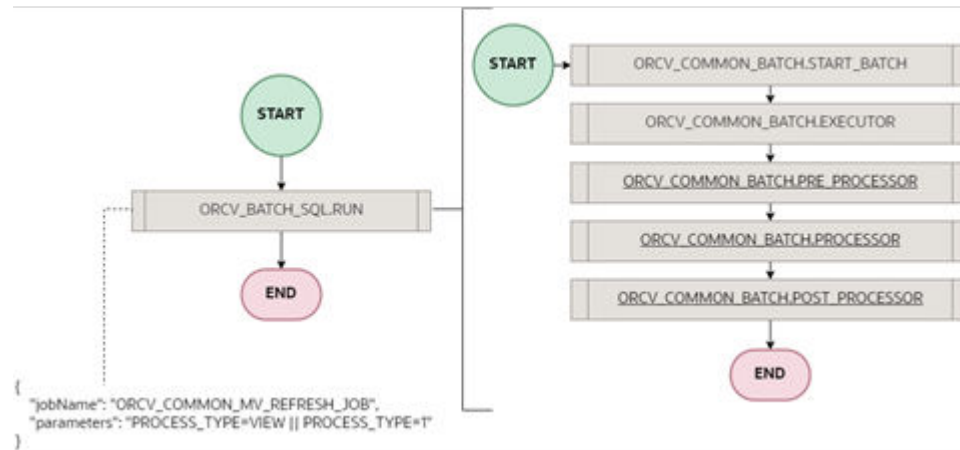


Batch Flow

This section describes the batch flow.

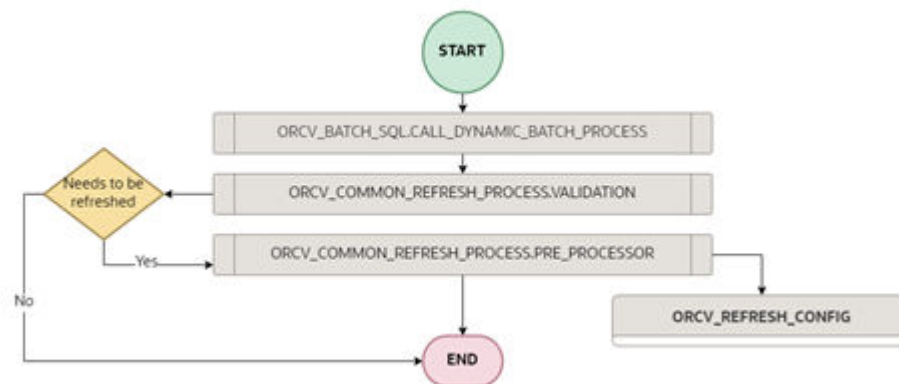
Run

Figure 27-14 Materialized View Run Function



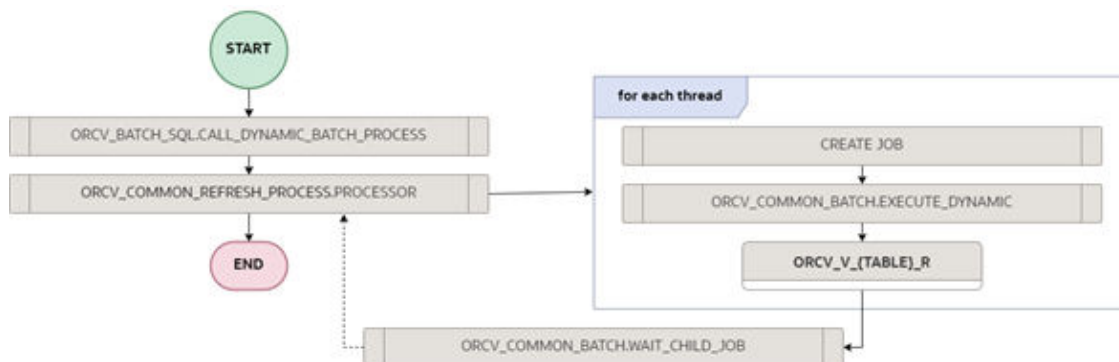
Pre Processor

Figure 27-15 Materialized View Pre Processor Function



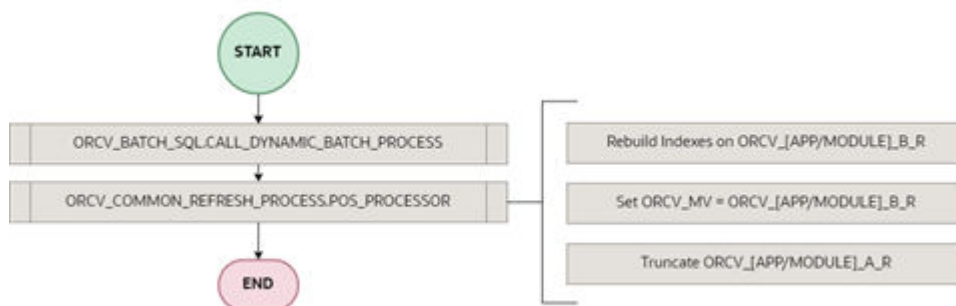
Processor

Figure 27-16 Materialized View Processor Function



Post Processor

Figure 27-17 Materialized View Post Processor Function



Refresh Materialize View Web Service

Service that starts the refresh materialized view batch.

Frequency: Ad-hoc

Request Method: POST

Refresh Materialize View	
Url Path	<BASE URL>/ords/mfcs/orcv/common/refreshmv/batch/start
Request Method	POST

Refresh Materialize View	
Sample Request Body	{ "jobName": "ORCV_COMMON_MV_REFRESH_JOB", "parameters": "PROCESS_TYPE=<PROCESS_TYPE> PROCESS_TYPE_VALUE=<RESULT_SYNONYM>" }
Sample Success Message	{ "executionId": 181, "status": "Submitted", "executionInfo": "null" } Response Code: 200 (Success)

Parameters:

Parameter	Description	Mandatory
PROCESS_TYPE	Valid values are: VIEW MOD (Module) CHAIN This value indicates the definition of process type value.	Y
PROCESS_TYPE_VALUE	Result Synonym from ORCV_REFRESH_MV_CONFIG by process_type	Y
REFRESH	Valid values are: F (force refresh without checking if the materialized view needs refreshing or checks for its dependency batch)	N

Database Objects

Object Type	Name	Sub Object Type	Sub Object Name	Description
Table	ORCV_REFRESH_MV_CONFIG	NA	NA	Configuration of refresh process.
Table	ORCV_REFRESH_MV_DEPENDENCY	NA	NA	Configuration of refresh process dependency.
Table Type	ORCV_REFRESH_MV_CONFIG_OBJ	NA	NA	NA
Package	ORCV_COMMON_MV_REFRESH	Function	COMMON_PRE_PROCESSOR	Function that segregates in threads all records the view source.
		Function	VALIDATION	Validate if there is a need to Refresh the Materialized View.
		Function	PRE_PROCESSOR	Prepare all records for Processor step, organize head when MOD or CHAIN, created detail views and set level dependency.

Object Type	Name	Sub Object Type	Sub Object Name	Description
		Function	PROCESSOR	Apply the Rule like an example, truncated the current table A and fill table, by threads or not.
		Function	POST_PROCESSOR	Rebuild indexes, switch synonym if process step has executed with success.

Configurations

Objects

Object Type	Name	Description
Table	ORCV_[APP/MODULE]_A_R	Table that will contain info. Used for synonym
Table	ORCV_[APP/MODULE]_B_R	Table that will contain info. Used for synonym.
View	ORCV_V_[APP/MODULE]_R	View with information to insert into A_R or B_R tables.
View	ORCV_V_[APP/MODULE]_RC	View used to control threads.
View	ORCV_V_[APP/MODULE]_VL	View used to control if there is new data.



Note:

A Synonym ORCV_MV_[APP/MODULE] for ORCV_[APP/MODULE]_A_R table will need to be created.

Purge Framework

This section describes the Purge Framework.

Overview

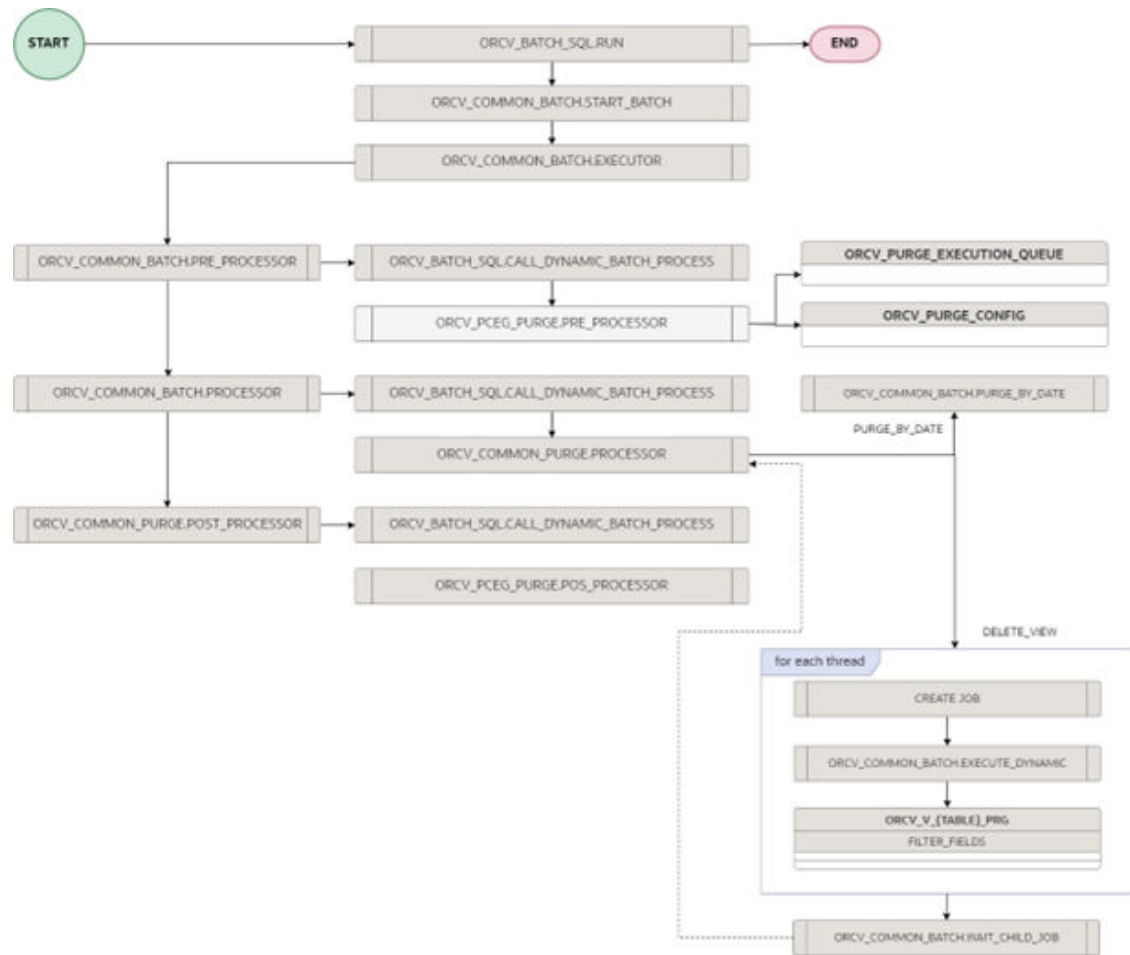
As part of Retail Supply Chain Hub, the mechanism for purging data from all applications is provided by a framework that can be configured with the entities, frequencies, and dependencies between them.

Out of the box, the purging framework can be configured with two different purging mechanisms implemented depending on the target table to be purged. If the table that needs to be purged is partitioned by a date type column, then a purge by date method can be used. In this scenario partitions older than the PURGE_DAYS system parameter will be dropped.

As an alternative, the approach is based on a delete view driving logic that requires a driving view configuration. This view will return the rows to be purged for each table and each thread, returning the selected column to filter the row, as FILTER_FIELDS, and the thread number that will purge the record. Both the filter column and the number of threads that can be configured in the table ORCV_PURGE_CONFIG.

Batch Flow

Figure 27-18 Purge Framework Batch Flow



Configuration

Affected Table	Purge Days
ORCV_THREAD_CONTROL	30
ORCV_LOGS	30
ORCV_BATCH_AUDIT_LOG	30
ORCV_WORKSPACE_LOGIN	2

Purge Web Service

Request Method: POST

Authorization Method: OAuth 2.0

Frequency: Nightly

URL: <BASE_URL>/ords/mfcs/orcv/merch/common/purge/<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
batch/start	POST	Empty	Configured purge tables
batch/restart	POST	{ "executionId": <EXEC_ID> }	Configured purge tables
batch/log	GET	URL query parameters: /log?jobName=ORCV_COMMON_PURGE_JOB& executionId=<EXEC_ID>&logDate=YYYY-MM-DD	N/A
batch/kill	POST	Empty	N/A
batch/status	GET	URL query parameters: /status? jobName=ORCV_COMMON_PURGE_JOB& executionId=<EXEC_ID>	N/A
/ping	GET	Empty	N/A

Events Framework

The Events Framework supports the overall managing of notifications on screen and alerts between all applications.

This includes notifying the user in-app whenever an event occurs of type Notification and alerting the configured users outside of the applications, through e-mail or others for events of type Alert. An Event can originate one or multiple Notifications and/or Alerts.

The notifications and alerts are driven by a template that can be defined at the database or calling external services, for example for:

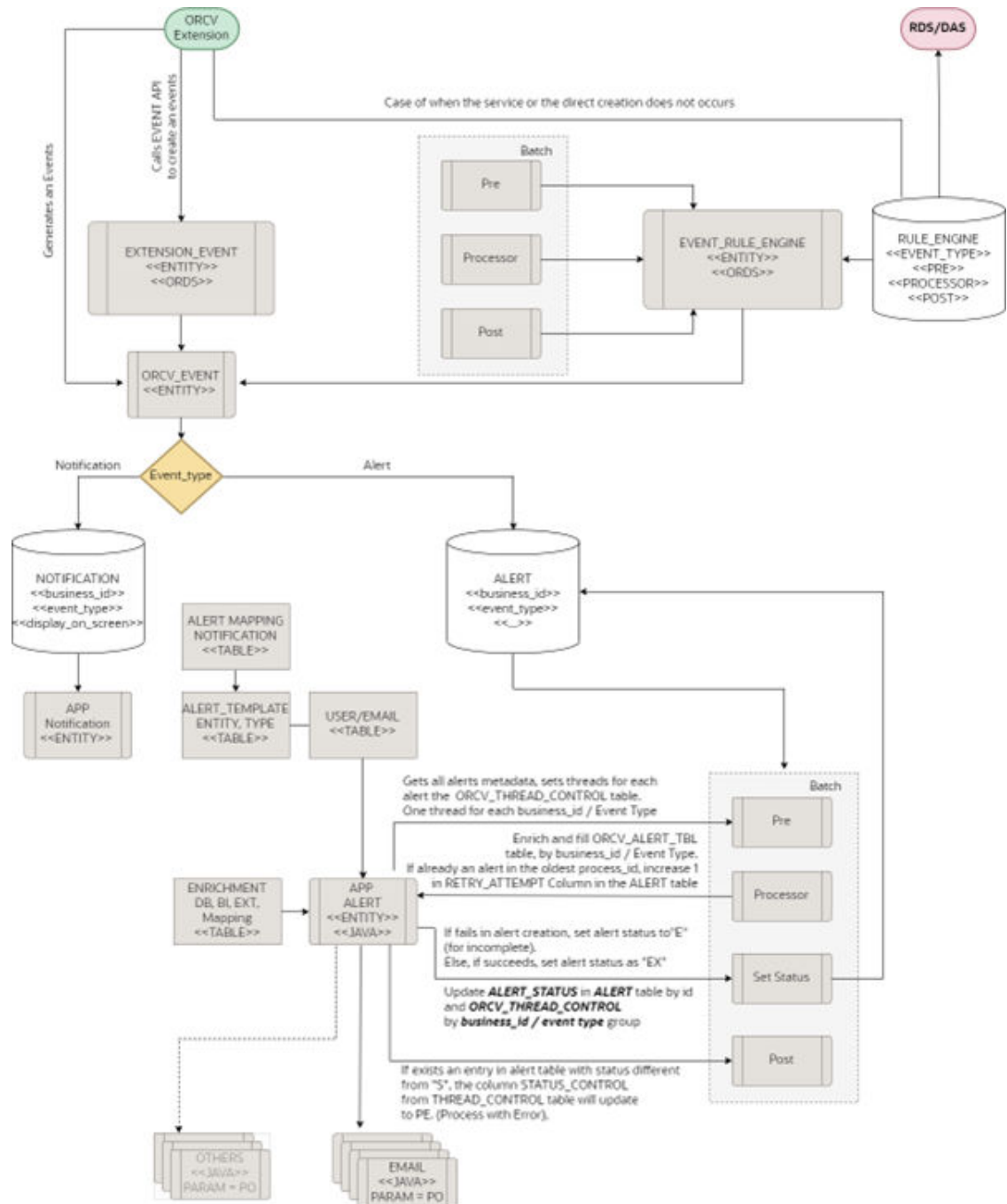
- Alerts – For a specific alert it will be possible to configure an Oracle Business Intelligence Publisher that on request, through a ReST endpoint, return a Portable Document Format (PDF) that will be include has an attachment on the email.
- Notifications – For a specific notification will be possible to configure a URL that once the user receives the notification on app can to a contextual launch for example to a specific entity on the screen.

Considering that the framework is configurable it will be possible to extend existing events with new notifications or even alerts.

The Events Framework has a retry mechanism that is responsible for handling the cases when there is an error creating the alert, or it was created but cannot be sent.

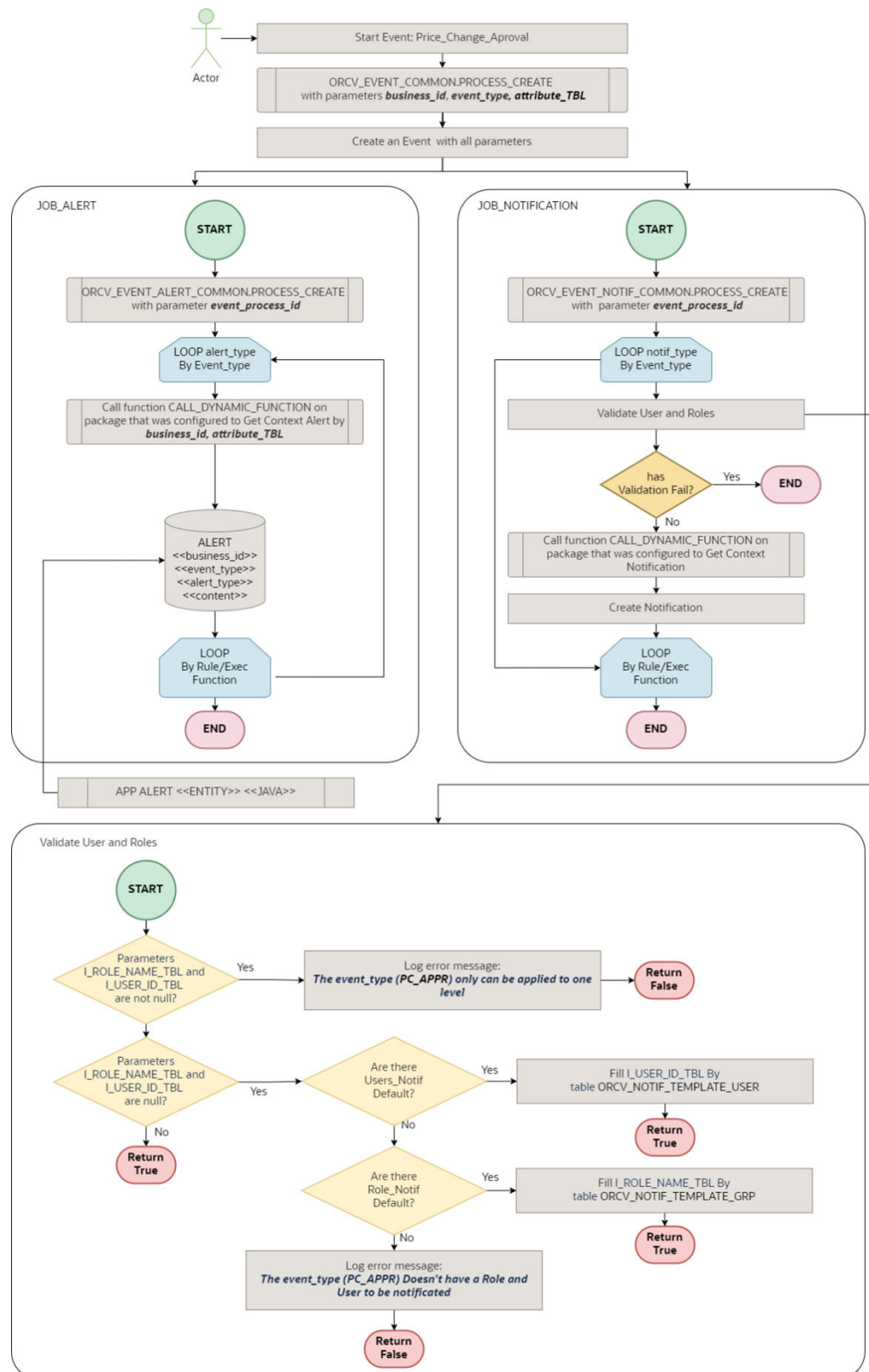
Process Overview

Figure 27-19 Events Process Overview



Events Process Flow

Figure 27-20 Events Process Flow



Configuration

This section describes configurations.

Configuration Tables

Configuration table: ORCV_EVENT

This table will contain events.

Column Name	Description	Example
EVENT_TYPE	This field contains the event type uniquely identifying each row on the table	ORCV_PRICE_INT_PC_EVENT
EVENT_DESC	This field contains the event description	Message error in RPCS integration
ALERT_ENQUEUE_EVENT	This column holds the function for alert enqueue event	ORCV_EVENT_ALERT_AQP.ENQUEUE
NOTIF_ENQUEUE_EVENT	This column holds the function for notif enqueue event	NULL
STATUS	This field contains the event status	A

Configuration table: ORCV_EVENT_NOTIF_CONFIG

The table contains the notifications configurations.

Column Name	Description	Example
EVENT_TYPE	Indicates the event type. The convention to create the event type ID is to use the first number as the identifier for a given entity	ORCV_PCEG_EVENT_NORMAL
NOTIF_TEMPLATE_NAME	The Notification Event Template name.	PCEG_EVENT_NORMAL
CONTENT_FUNCTION	This field contains the name of the function that returns content notification	ORCV_PCEG_EVENT.CREATE_CONTENT_NOTIF
QUEUE_NAME	This column holds the queue name of the template	ORCV_EVENT_NOTIF_AQ
NUMBER_QUEUES	This column holds the number of queues	4
MAX_RETRY	This field contains the max retry attempts count that was used in the process of retry	NULL
WAIT_TIME_RETRY	This field contains the wait time to retry attempts	NULL
STATUS	This field contains the status code of the event (A - Active; I - Inactive)	A
DISCARD_BUSINESS_ID	Indicates if the notification will discard business ID, valid values (Y, N), Defaults "N"	Y

Configuration table: ORCV_EVENT_NOTIF_TEMPLATE

The table containing the notifications templates.

Column Name	Description	Example
NOTIF_TEMPLATE_NAME	The Notification Event Template unique name.	SPIM_EVENT_SUBM_NEW_ITEM
SEVERITY	The severity of the template Notification (1-Critical, 2_Important, 3-Normal)	3
LAUNCHABLE	Indicates if the notification URL is launchable	N
APPLICATION_CODE	Application where notification came from	SPIM
NAME	This column contains the name of the template	SPIM_EVENT_SUBM_NEW_ITEM
DESC_TEMPLATE	Free message text, to be associated to template	New Items submitted
URL	This column contains the URL associated with the notif template	NULL
NOTIF_TYPE	Notification type of the Notification Template	NULL
NOTIF_TARGET	Notification target	NULL
CONTENT_ENDPOINT_ID	Endpoint ID for the content	NULL

Configuration table: ORCV_EVENT_NOTIF_TEMPLATE_TL

The table containing the notifications templates translations.

Column Name	Description
LANG	Contains a number that uniquely identifies a language
SEVERITY	The severity of the template Notification (1-Critical, 2_Important, 3-Normal)
LAUNCHABLE	Indicates if the notification URL is launchable
APPLICATION_CODE	Application where notification came from
NAME	This column contains the name of the template
DESC_TEMPLATE	Free message text, to be associated to template
URL	This column contains the URL associated with the notif template
NOTIF_TYPE	Notification type of the Notification Template
NOTIF_TARGET	Notification target
CONTENT_ENDPOINT_ID	Endpoint ID for the content

Configuration table: ORCV_EVENT_ALERT_CONFIG

Table that defines the alert templates for each Event Type.

Column Name	Description	Example
IS_ASYNC	This column holds the flag to show if the alert is asynchronous.	Y
ATTRIBUTE_FUNCTION	This field contains the name of the function that returns the attributes for the event.	ORCV_SPIM_EVENT_NOTIF_ALERT.GET_EVENT_ALERT_ATTRIBUTES

Column Name	Description	Example
ALERT_TEMPLATE_NAME	This column contains the unique name of the template	SPIM_EVENT_ITEMS_PENDING_ACTION
EVENT_TYPE	This field contains the event type.	ORCV_SPIM_EVENT_ITEMS_PENDING
STATUS	This field contains the status. Valid values (A-Active, I - Inactive).	A
CONTENT_FUNCTION	This field contains the name of the function that return content alert.	ORCV_SPIM_EVENT_NOTIF_ALERT.CREATE_CONTENT_ALERT_PENDING_ACTION
QUEUE_NAME	This column holds the queue name of the template.	ORCV_EVENT_ALERT_AQ
NUMBER_QUEUES	This column holds the number of queues.	4
MAX_RETRY	This field contains the max retry attempts count that was used in the process of retry.	3
WAIT_TIME_RETRY	This field contains the wait time to retry attempts.	NULL

Configuration table: ORCV_EVENT_ALERT_TEMPLATE

The table containing the alerts configuration.

Column Name	Description	Example
ALERT_TEMPLATE_NAME	This column contains the name of the template.	ORCV_SPIM_EVENT_ITEMS_PENDING
ALERT_TYPE	Indicates what kind of alert the context corresponds to. Valid values (EMAIL, SMS, and so on).	EMAIL
FORMAT_FILE	Indicates what kind of title (HTML, CSV, or TEXT).	HTML
CONTENT_VALUE	Content value.	NULL
CONTENT_ENDPOINT_ID	Content WS endpoint ID.	NULL
CONTENT_WALLET_KEY	Content WS endpoint wallet key.	NULL
CONTENT_WALLET_MAP	Content WS endpoint wallet map.	NULL
TITLE_TYPE	Indicates what kind of title (for example: BIP, EXTERNAL-WS).	DB_TEXT
TITLE_ENDPOINT_ID	Title WS endpoint ID.	NULL
TITLE_VALUE	The description of the title the alert.	Items Pending
TITLE_WALLET_KEY	Title WS endpoint wallet key.	NULL
TITLE_WALLET_MAP	Title WS endpoint wallet map.	NULL
SUBJECT_TYPE	Indicates what kind of subject (U-rl or T-ext).	DB_TEXT
SUBJECT_ENDPOINT_ID	Subject WS endpoint ID.	NULL
SUBJECT_VALUE	The description of the subject the alert.	Items Pending
SUBJECT_WALLET_KEY	Subject WS endpoint wallet key.	NULL

Column Name	Description	Example
SUBJECT_WALLET_MAP	Subject WS endpoint wallet map.	NULL
BODY_TYPE	Indicates what kind of body (U-rl or T-ext).	BIP
BODY_ENDPOINT_ID	Body WS endpoint ID.	ORCV_BIP_REPORTS
HEADER_VALUE	The description of the header the alert.	NULL
BODY_VALUE	The description of the body the alert.	MFCS%252FOrders%252Ford_det%252Ford_det.xdo/run
FOOTER_VALUE	The description of the footer the alert.	NULL
BODY_WALLET_KEY	Body WS endpoint wallet key.	bipBasicClient
BODY_WALLET_MAP	Body WS endpoint wallet map.	oracle.retail.apps.platform.services.bip.security.basic
CONTENT_AUTH_TYPE	Content authorization type: BASIC, IDCS.	BASIC
CONTENT_AUTH_URL	Content URL for non-basic authorization type.	NULL
CONTENT_SCOPE	Content scope for non-basic authorization type.	NULL
TITLE_AUTH_TYPE	Title authorization type: BASIC, IDCS.	NULL
TITLE_AUTH_URL	Title URL for non-basic authorization type.	NULL
TITLE_SCOPE	Title scope for non-basic authorization type.	NULL
SUBJECT_AUTH_TYPE	Subject authorization type: BASIC, IDCS.	NULL
SUBJECT_AUTH_URL	Subject URL for non-basic authorization type.	NULL
SUBJECT_SCOPE	Subject scope for non-basic authorization type.	NULL
BODY_AUTH_TYPE	Body authorization type: BASIC, IDCS.	NULL
BODY_AUTH_URL	Body URL for non-basic authorization type.	NULL
BODY_SCOPE	Body scope for non-basic authorization type.	NULL
BODY_LIST_LIMIT	Maximum body list size for sending the alert by email. If not defined the System Parameters BODY_LIST_LIMIT should be used as a default.	NULL

Configuration table: ORCV_EVENT_ALERT_TEMPL_ATTACH

The table containing the alerts configuration.

Column Name	Description	Example
ALERT_TEMPL_ATTACH_ID	Unique identifier for rows in this table.	1
ALERT_TEMPLATE_NAME	This column contains the unique name of the template	ORCV_SPIM_EVENT_ITEMS_PENDING
FORMAT_FILE	Indicates what kind of file format; for example: CSV, PDF	PDF
ATTACH_NAME	This column contains the name of the attach.	application/octet-stream
ATTACH_TYPE	Indicates what kind of attachment type (URL or TEXT).	BIP
ATTACH_MIME_TYPE	Indicates what kind of MIME type.	application/octet-stream
ATTACH_VALUE	This column contains the attached, which could be a base64 string.	{BIP_SERVER}/xmlpserver/services/rest/v1/reports/MFCS%252FOrders%252Ford_det%252Ford_det.xdo/run
ATTACH_WALLET_KEY	This field contains the wallet key.	bipBasicClient
ATTACH_WALLET_MAP	This field contains the wallet map.	oracle.retail.apps.platform.services.bip.security.basic
ATTACH_AUTH_TYPE	Content authorization type: BASIC, IDCS.	BASIC
ATTACH_AUTH_URL	Content URL for non-basic authorization type	NULL
ATTACH_SCOPE	Content scope for non-basic authorization type	NULL

Configuration table: ORCV_EVENT_ALERT_BATCH_CONFIG

This table contains the batch configurations for the event_type/alert_type combination.

Column Name	Description	Example
EVENT_TYPE	The event type identifier (Order creation, Order update, and so on).	ORCV_SPIM_EVENT_ITEMS_PENDING
ALERT_TYPE	Indicates what kind of Alert it is. Valid values (EMAIL, SMS, and so on).	EMAIL
STATUS	This field contains the status of the configuration. (A-Active, I-Inactive).	A
ADHOC_IND	This column indicates whether the flow is classified as an ad hoc flow. Valid values are: 'Y', 'N'.	N

System Parameters

Listed below is the list of configurations at system level that are used by Events Framework.

The parameters are defined in table ORCV_SYSTEM_PARAMETERS.

Area	Parameter	Description	Value 1	Value 2
BATCH	THRESHOLD_INT	Default threshold limit	10	NA
CONFIG	MAX_JOBS_RUNNIN G	Maximum jobs to run in parallel to process events_notifications	16	NA
CONFIG	MAX_JOBS_RUNNIN G	Maximum jobs to run in parallel to process event_alerts	16	NA
NOTIF	APEX_NOTIF_MAX_ ROWS	Default value for the max number of rows to show in the apex list of notifications	10	NA
PURGE	NOTIF_RETENTION_ PERIOD	Default retention period in days before purging deleted notifications	60	NA
PURGE	PURGE_DAYS	Default purge Period	ORCV_EVENT_ALERT	30
PURGE	PURGE_DAYS	Default purge Period	ORCV_EVENT_ALERT_ ATTACH	30
PURGE	PURGE_DAYS	Default purge Period	ORCV_EVENT_ALERT_ ERROR	30
PURGE	PURGE_DAYS	Default purge Period	ORCV_EVENT_NOTIF	30
PURGE	PURGE_DAYS	Default purge Period	ORCV_EVENT_NOTIF_ GRP_STATUS	30
PURGE	PURGE_DAYS	Default purge Period	ORCV_EVENT_NOTIF_ USER_STATUS	30
PURGE	PURGE_DAYS	Default purge Period	ORCV_EVENT_PROCESS	30
PURGE	PURGE_DAYS	Default purge Period	ORCV_EVENT_PROCESS_ ATTRIBUTES	30
PURGE	PURGE_DAYS	Default purge Period	ORCV_EVENT_PROCESS_ ROLE_NAME	30
PURGE	PURGE_DAYS	Default purge Period	ORCV_EVENT_PROCESS_ TARGET_APP	30
PURGE	PURGE_DAYS	Default purge Period	ORCV_EVENT_PROCESS_ TARGET_ROLE	30

Codes

Certain operations require codes to define the options available at system level. Below you can find the list of code types and the codes available, the sequence and their description.

The codes are defined in tables ORCV_CODE_HEAD and RCV_CODE_DETAIL.

Code Type	Description
ALET	Alert Error Types
ALST	Alert Status
NTST	Notifications Status
NTST	Notifications Status

Batch

This section describes the list of batch jobs used by this framework.

Module	Program Name	Max Chunk Size	Max Concurrent Threads	Retry Lock Attempts	Retry Indicator
EVENTS	ORCV_EVENT_ALERT_JOB	1000	4	3	N

Web Service

Frequency: Ad-hoc

Authorization Method: OAuth 2.0

Base URL: <BASE_URL>/EventAlertsJavaServicesWeb/services/orcv/merch/event/integration/alerts/<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
batch/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_EVENT_ALERT_JOB", "agentExecutionId": "", "parameters": "EVENT_TYPE=<EVENT_TYPE>" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_EVENT_PROCESS ORCV_EVENT_ALERT ORCV_EVENT_ALERT_ATTACH ORCV_EVENT_NOTIF ORCV_EVENT_ALERT_ERROR
batch/restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_EVENT_ALERT_JOB", "agentExecutionId": "", "parameters": "EVENT_TYPE=<EVENT_TYPE>" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_EVENT_PROCESS ORCV_EVENT_ALERT ORCV_EVENT_ALERT_ATTACH ORCV_EVENT_NOTIF ORCV_EVENT_ALERT_ERROR

URL Path	Request Method	Request Body	Affected Tables
batch/log	GET	URL query parameters: /log? jobName=ORCV_EVENT_ALERT_JOB& executionId=<EXEC_ID>&date=YYYY-MM-DD	N/A
batch/kill	POST	{ "executionId": <EXEC_ID>, "jobName": "ORCV_EVENT_ALERT_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
batch/status	GET	URL query parameters: /status?jobName=ORCV_EVENT_ALERT_JOB&executionId=<EXEC_ID>	N/A
/ping	GET	Empty	N/A

POM (Process Orchestration and Monitoring)

Batch Schedule Download

Download Through Upload Download Application UI

Retail Supply Chain Hub has multiple batch jobs that must be scheduled in POM, to allow for the initial setup and update of the ORCV POM schedule, the Base Schedule spreadsheet can be downloaded through the Upload Download application.

Download requirements:

- Retailer type user with access provided to POM Module application through the Profile.

For a user that fulfills the download requirements the download can be done from the Upload Download application on the Download Blank tab with the following selections:

- Template Source: POM Module
- Template Name: Pom Batch Schedule Template
- File Type: XLSX

Download through Web Service

As an alternative way of downloading the ORCV schedule in POM, the following service can be invoked through ReST and provides the default spreadsheet for ORCV schedule.

Request Method: POST

Authorization Method: OAuth 2.0 with Password Credentials Token

Note: Password Credentials Token needs to be requested for a user with group ORCV_POM_APP_JOB assigned to him in IDCS or in ORCV_SEC_USER_APP.

Base URL: <BASE_URL>/ords/mfcs/orcv/upldnld/<URL_PATH>/

URL Path	Request Method	Request Body	Response
/downloadui	POST	{ "fileType": "XLSX", "downloadType": "B", "filters": [{ "filterValues": [""] }], "templateKey": "BATCH_SCHEDULE_DATA", "appCode": "POM" }	Base64 encoded XLSX file, named: BATCH_SCHEDULE_DATA.XLSX

Batch Schedule Configuration Download

Download Through Upload Download Application UI

Retail Supply Chain Hub has multiple batch jobs that must be scheduled in POM with different frequencies, to allow for the initial setup, the Schedule Configuration spreadsheet can be downloaded through the Upload Download application.

Download requirements:

- Retailer type user with access provided to POM Module application through the Profile.

For a user that fulfills the download requirements the download can be done from the Upload Download application on the Download Blank tab with the following selections:

- Template Source: POM Module
- Template Name: Pom Batch Schedule Configuration Template
- File Type: XLSX

Download through Web Service

As an alternative way of downloading the ORCV schedule in POM, the following service can be invoked through ReST and provides the default spreadsheet for ORCV schedule.

Request Method: POST

Authorization Method: OAuth 2.0 with Password Credentials Token

Note: Password Credentials Token needs to be requested for a user with group ORCV_POM_APP_JOB assigned to him in IDCS or in ORCV_SEC_USER_APP.

Base URL: <BASE_URL>/ords/mfcs/orcv/upldnld/<URL_PATH>/

URL Path	Request Method	Request Body	Response
/downloadui	POST	<pre>{ "fileType": "XLSX", "downloadType": "B", "filters": [{ "filterValues": [""] }], "templateKey": "BATCH_SCHEDULE_CONFIG_DATA", "appCode": "POM" }</pre>	Base64 encoded XLSX file, named: BATCH_SCHEDULE_CONFIG_DATA.XLSX

Callback Service

The POM callback service allows Retail Supply Chain Hub to receive notifications of external batch status, which can then be set as data dependencies to processes like the materialized view refresh or other internal batches. The main objective of this callback is to log the batch process status that can then be used to trigger dependencies, from a data replication perspective from other batches.

Web Service

This section describes the web service.

ORDS Web Service

Request Method: POST

Authorization Method: OAuth 2.0

Base URL: <BASE_URL>/ords/mfcs/orcv/common/batch/pom/<URL_PATH>

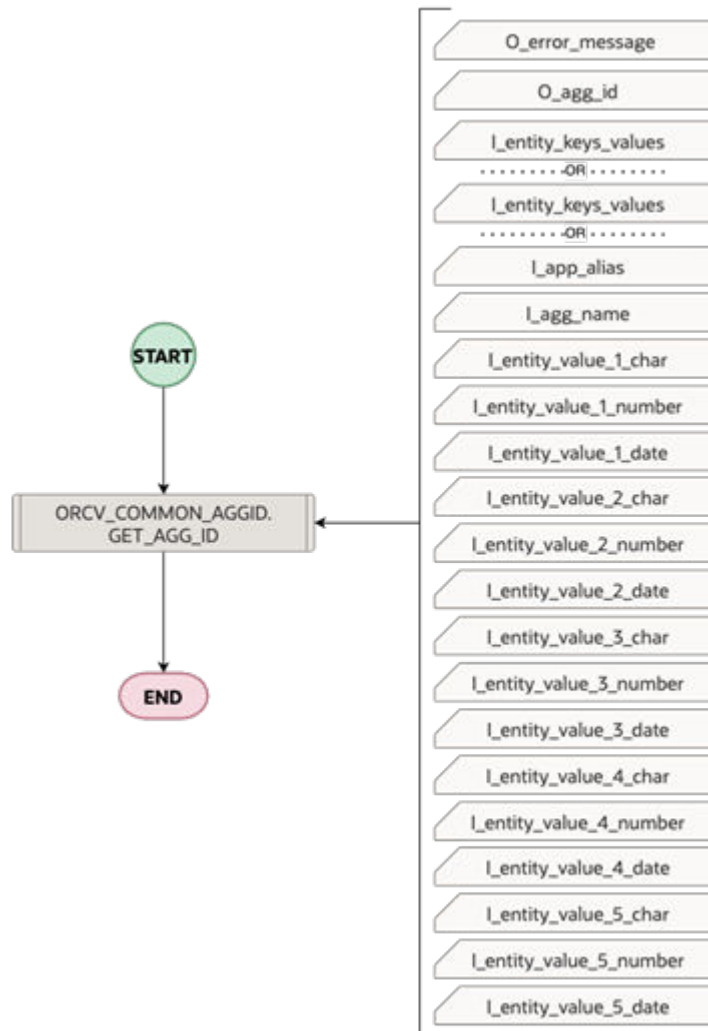
URL Path	Request Method	Request Body	Response
/auditlogcallback	POST	{ "callerId": "XXX", "correlationId": "37", "processName": "MERCH_BATCH_PROCESS_01", "processExecutionId": "MERCH_BATCH_PROCESS_01", "activityName": "MERCH_BATCH_JOB", "activityExecutionId": "123456", "status": "COMPLETED", "activityStatus": "ACTIVITY_COMPLETED", "failedActivity": null }	ORCV_BATCH_AUDIT_LOG
/ping	GET	Empty	NA

Aggregator Framework

The Aggregator Framework (AGG) creates a common identifier based on given keys and rules. It receives the details to determine the configuration to be applied and the key values, then concatenates multiple columns and values and resulting in a return object with the identifier.

Process Overview

Figure 27-21 Aggregator Framework Process Overview



Database Objects

This section describes database objects.

Tables

This section describes tables.

ORCV_AGGID_CONFIG

This table contains the configuration for generation of aggregation identifiers based on table data.

Table Columns

Column Name	Data Type	Description
APP_ALIAS	VARCHAR2	Unique number that identifies a price event. Fed by a sequence.
AGG_NAME	VARCHAR2	Unique identifier for this config inside this app alias.
ENTITY_TABLE	VARCHAR2	Name of the table to be used as source for the identifier creation.
ENTITY_KEY_1	VARCHAR2	Name of the column to use to filter from entity table.
ENTITY_KEY_2	VARCHAR2	Name of the column to use to filter from entity table.
ENTITY_KEY_3	VARCHAR2	Name of the column to use to filter from entity table.
ENTITY_KEY_4	VARCHAR2	Name of the column to use to filter from entity table.
ENTITY_KEY_5	VARCHAR2	Name of the column to use to filter from entity table.
AGG_DELIMITER	VARCHAR2	Delimiter to use between AGG fields in the identifier created.
AGG_FIELD_1	VARCHAR2	Name of the column to use in identifier creation.
AGG_FIELD_2	VARCHAR2	Name of the column to use in identifier creation.
AGG_FIELD_3	VARCHAR2	Name of the column to use in identifier creation.
AGG_FIELD_4	VARCHAR2	Name of the column to use in identifier creation.
AGG_FIELD_5	VARCHAR2	Name of the column to use in identifier creation.
CREATE_ID	VARCHAR2	This field contains the user that created the record.
CREATE_DATETIME	DATE	This field contains the datetime when the record is created.
LAST_UPDATE_ID	VARCHAR2	This field contains the user that last updated the record.
LAST_UPDATE_DATETIME	DATE	This field contains the datetime when the record is last updated.
ACTIVE_IND	VARCHAR2	This field indicates if the config is active.

Table Constraints

Table	Columns	Constraint
ORCV_AGGID_CONFIG	APP_ALIAS, AGG_NAME	APP_ALIAS, AGG_NAME must be unique.

Packages

This section describes packages.

Package ORCV_COMMON_AGGID

Function	Description
VALIDATE_DATA_TYPES	Validates whether the input parameters data types against the table columns data types and return all the WHERE clause conditions to use in SELECT
GET_AGG_ID	Return an aggregation identifier for a given configuration for the following parameters: O_error_message O_agg_id I_entity_keys_values

Function	Description
GET_AGG_ID	Return an aggregation identifier for a given configuration for the following parameter: I_entity_keys_values
GET_AGG_ID	Return an aggregation identifier for a given configuration for the following parameters: I_app_alias I_agg_name I_entity_value_1_char I_entity_value_1_number I_entity_value_1_date I_entity_value_2_char I_entity_value_2_number I_entity_value_2_date I_entity_value_3_char I_entity_value_3_number I_entity_value_3_date I_entity_value_4_char I_entity_value_4_number I_entity_value_4_date I_entity_value_5_char I_entity_value_5_number I_entity_value_5_date

Configuration

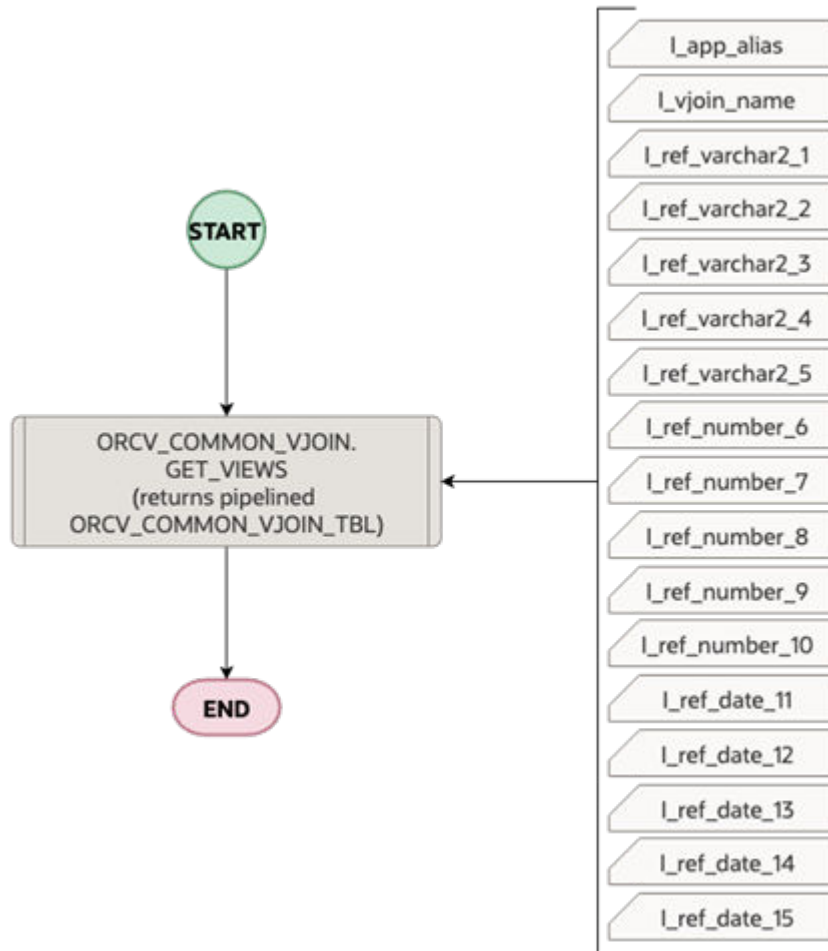
#	Step	Example
1	Add the configuration to ORCV_AGGID_CONFIG	<pre> MERGE INTO orcv_aggid_config c USING (SELECT 'SPPM' app_alias, 'SPPM_PRM' agg_name, 'ORCV_V_SPPM_PRM_AGG' entity_table, 'PRICE_EVENT_ID' entity_key_1, NULL entity_key_2, NULL entity_key_3, NULL entity_key_4, NULL entity_key_5, '_' agg_delimiter, '''SPPM_SUPPLIER''' agg_field_1, 'SUPPLIER' agg_field_2, 'START_DATE' agg_field_3, 'EXCLUSIVE_DISCOUNT_IND' agg_field_4, NULL agg_field_5, 'Y' active_ind FROM DUAL) tbl ON (c.app_alias = tbl.app_alias AND c.agg_name = tbl.agg_name) WHEN NOT MATCHED THEN INSERT (app_alias, agg_name, entity_table, entity_key_1, entity_key_2, entity_key_3, entity_key_4, entity_key_5, agg_delimiter, agg_field_1, agg_field_2, agg_field_3, agg_field_4, agg_field_5, active_ind) VALUES (tbl.app_alias, tbl.agg_name, tbl.entity_table, tbl.entity_key_1, tbl.entity_key_2, tbl.entity_key_3, tbl.entity_key_4, tbl.entity_key_5, tbl.agg_delimiter, tbl.agg_field_1, tbl.agg_field_2, tbl.agg_field_3, tbl.agg_field_4, tbl.agg_field_5, tbl.active_ind); </pre>
2	Add a call to the API with the required parameters.	<pre> ORCV_COMMON_AGGID.GET_AGG_ID(I_app_alias => 'SPPM', I_agg_name => 'SPPM_PRM', I_entity_value_1_number => price_event_id); </pre>

View Join Framework

View Join is a Framework that joins multiple views or tables together in a specific order defined in its configuration as a pipelined function.

Process Overview

Figure 27-22 View Join Framework Process Overview



Database Objects

This section describes database objects.

Tables

This section describes tables.

ORCV_COMMON_VJOIN_CONFIG

This table contains the configuration for Views Join.

Table Columns

Column Name	Data Type	Description
APP_ALIAS	VARCHAR2	This field indicates the APP Alias.
VJOIN_NAME	VARCHAR2	This field indicates the View Join name.
SEQ_NO	NUMBER	This field indicates the view sequence number execution.
WEIGHT	NUMBER	This field indicates Weight for the view, higher weight means higher priority on view execution.
VIEW_NAME	VARCHAR2	This field indicates the Source View Name.
CREATE_ID	VARCHAR2	This field contains the user that created the record.
CREATE_DATETIME	DATE	This field contains the datetime when the record is created.
LAST_UPDATE_ID	VARCHAR2	This field contains the datetime when the record is last updated.
LAST_UPDATE_DATETIME	DATE	This field contains the user that last updated the record.

Table Constraints

Table	Columns	Constraint
ORCV_COMMON_VJOIN_CONFIG	APP_ALIAS, VJOIN_NAME, SEQ_NO	APP_ALIAS, VJOIN_NAME, SEQ_NO must be unique.

Packages

This section describes packages.

Package ORCV_COMMON_VJOIN

Function	Description
GET_COLUMNS	Prepares a string with all columns names to build the final query.
GET_WHERE_CLAUSE	Prepares the where clause of one column (reference/filter column) to build the final query.
GET_WHERE_CLAUSE_ALL	Prepares the WHERE clause of all columns (reference/filter columns) to build the final query. Evokes the GET_WHERE_CLAUSE function for each reference/filter column.
GET_VIEWS	Prepares a string to obtain data from all parameterized views in the ORCV_COMMON_VJOIN_CONFIG table. The views to be aggregated are defined in the ORCV_COMMON_VJOIN_CONFIG.VIEW_NAME column and are grouped by the ORCV_COMMON_VJOIN_CONFIG.VJOIN_NAME column. The function ORCV_COMMON_VJOIN.GET_VIEWS will return data as a pipelined table object.

Views

All views defined in table/column ORCV_COMMON_VJOIN_CONFIG/VIEW_NAME must comply with the format below:

Column	Type	Description
REF_VARCHAR2_1	VARCHAR2	Varchar Column-1 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_VARCHAR2_2	VARCHAR2	Varchar Column-2 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_VARCHAR2_3	VARCHAR2	Varchar Column-3 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_VARCHAR2_4	VARCHAR2	Varchar Column-4 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_VARCHAR2_5	VARCHAR2	Varchar Column-5 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_NUMBER_6	NUMBER	Number Column-6 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_NUMBER_7	NUMBER	Number Column-7 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_NUMBER_8	NUMBER	Number Column-8 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_NUMBER_9	NUMBER	Number Column-9 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_NUMBER_10	NUMBER	Number Column-10 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_DATE_11	DATE	Data Column-11 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_DATE_12	DATE	Data Column-12 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_DATE_13	DATE	Data Column-13 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_DATE_14	DATE	Data Column-14 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
REF_DATE_15	DATE	Data Column-15 used to filter data when we call the ORCV_COMMON_VJOIN.GET_VIEWS function.
VARCHAR2_1	VARCHAR2	Column-1 used to store alphanumeric type column.
VARCHAR2_1	VARCHAR2	Column-2 used to store alphanumeric type column.
VARCHAR2_1	VARCHAR2	Column-3 used to store alphanumeric type column.
VARCHAR2_1	VARCHAR2	Column-4 used to store alphanumeric type column.
VARCHAR2_1	VARCHAR2	Column-5 used to store alphanumeric type column.
NUMBER_6	NUMBER	Column-6 used to store numeric type column.
NUMBER_7	NUMBER	Column-7 used to store numeric type column.
NUMBER_8	NUMBER	Column-8 used to store numeric type column.
NUMBER_9	NUMBER	Column-9 used to store numeric type column.
NUMBER_10	NUMBER	Column-10 used to store numeric type column.
DATA_11	DATE	Column-11 used to store date type column.
DATA_12	DATE	Column-12 used to store date type column.
DATA_13	DATE	Column-13 used to store date type column.
DATA_14	DATE	Column-14 used to store date type column.

Column	Type	Description
DATA_15	DATE	Column-15 used to store date type column.

Configuration

#	Step	Example
1	Create a View to be joined	<pre> CREATE OR REPLACE FORCE EDITIONABLE VIEW ORCV_V_SPPF_DASH_ALERTS1 AS WITH count_tb AS (SELECT qmf.no_days no_days, COUNT(1) ccount FROM orcv_v_sppf_suppliers sups, orcv_v_sppf_quest_min_freq qmf WHERE qmf.no_days > 0 AND NOT EXISTS (SELECT 0 FROM orcv_sppf_questn_reply_head qrh WHERE qrh.supplier = sups.supplier AND qrh.reply_Date >= sysdate - qmf.no_days) SELECT NULL ref_varchar2_1, NULL ref_varchar2_2, NULL ref_varchar2_3, NULL ref_varchar2_4, NULL ref_varchar2_5, TO_NUMBER(NULL) ref_number_6, TO_NUMBER(NULL) ref_number_7, TO_NUMBER(NULL) ref_number_8, TO_NUMBER(NULL) ref_number_9, TO_NUMBER(NULL) ref_number_10, TO_DATE(NULL) ref_date_11, TO_DATE(NULL) ref_date_12, TO_DATE(NULL) ref_date_13, TO_DATE(NULL) ref_date_14, TO_DATE(NULL) ref_date_15, REPLACE(REPLACE(err.error_text, '%s1', TO_CHAR(ctb.ccount)), '%s2', TO_CHAR(ctb.no_days)) varchar2_1, (SELECT cd.code_desc FROM orcv_v_das_code_detail_tl cd WHERE cd.code_type = 'ISST' AND cd.code = ac.alert_title) varchar2_2, ac.alert_type varchar2_3, ac.alert_icon varchar2_4, NULL varchar2_5, NULL number_6, NULL number_7, NULL number_8, NULL number_9, NULL number_10, NULL date_11, </pre>

#	Step	Example
		NULL date_12, NULL date_13, NULL date_14, NULL date_15 FROM orcv_sppf_alert_cfg ac, orcv_errors err, count_tb ctb WHERE ac.alert_id = 'DASH_SUP_NO_QST' AND err.error_key = ac.alert_message;

#	Step	Example
2	Create another View to be joined	<pre> CREATE OR REPLACE FORCE EDITIONABLE VIEW ORCV_V_SPPF_DASH_ALERTS2 AS WITH count_tb AS (SELECT qmf.no_days no_days, COUNT(1) ccount FROM orcv_v_sppf_suppliers sups, orcv_v_sppf_quest_min_freq qmf WHERE qmf.no_days > 0 AND NOT EXISTS (SELECT 0 FROM orcv_sppf_questn_reply_head qrh WHERE qrh.supplier = sups.supplier AND qrh.reply_Date >= sysdate - qmf.no_days) SELECT NULL ref_varchar2_1, NULL ref_varchar2_2, NULL ref_varchar2_3, NULL ref_varchar2_4, NULL ref_varchar2_5, TO_NUMBER(NULL) ref_number_6, TO_NUMBER(NULL) ref_number_7, TO_NUMBER(NULL) ref_number_8, TO_NUMBER(NULL) ref_number_9, TO_NUMBER(NULL) ref_number_10, TO_DATE(NULL) ref_date_11, TO_DATE(NULL) ref_date_12, TO_DATE(NULL) ref_date_13, TO_DATE(NULL) ref_date_14, TO_DATE(NULL) ref_date_15, REPLACE(REPLACE(err.error_text, '%s1', TO_CHAR(ctb.ccount)), '%s2', TO_CHAR(ctb.no_days)) varchar2_1, (SELECT cd.code_desc FROM orcv_v_das_code_detail_tl cd WHERE cd.code_type = 'ISST' AND cd.code = ac.alert_title) varchar2_2, ac.alert_type varchar2_3, ac.alert_icon varchar2_4, NULL varchar2_5, NULL number_6, NULL number_7, NULL number_8, NULL number_9, NULL number_10, NULL date_11, NULL date_12, NULL date_13, </pre>

#	Step	Example
		<pre> NULL date_14, NULL date_15 FROM orcv_sppf_alert_cfg ac, orcv_errors err, count_tb ctb WHERE ac.alert_id = 'DASH_SUP_NO_QST' AND err.error_key = ac.alert_message; </pre>
3	Insert the VJOIN configuration to the configuration table	<pre> INSERT INTO ORCV_COMMON_VJOIN_CONFIG (app_alias, vjoin_name, seq_no, weight, view_name) VALUES ('SPPF', 'DASH_ALERTS', 1, 0, 'ORCV_V_SPPF_DASH_ALERTS1'); -- INSERT INTO ORCV_COMMON_VJOIN_CONFIG (app_alias, vjoin_name, seq_no, weight, view_name) VALUES ('SPPF', 'DASH_ALERTS', 2, 0, 'ORCV_V_SPPF_DASH_ALERTS2'); </pre>
4	Call the API	<pre> SELECT varchar2_1 alert_desc, varchar2_2 alert_title, varchar2_3 alert_type, varchar2_4 alert_icon, varchar2_5 alert_action FROM TABLE(ORCV_COMMON_VJOIN.GET_VIEWS(I_app_alias => 'SPPF', I_vjoin_name => 'HOME_ALERTS')) </pre>

Database Integration Framework

The Database Integration Framework enables common functionalities in Retail Supply Chain Hub for integration with external endpoints.

Publish

This framework has the capability to publish to external systems by enabling APIs that implement the publish mechanism.

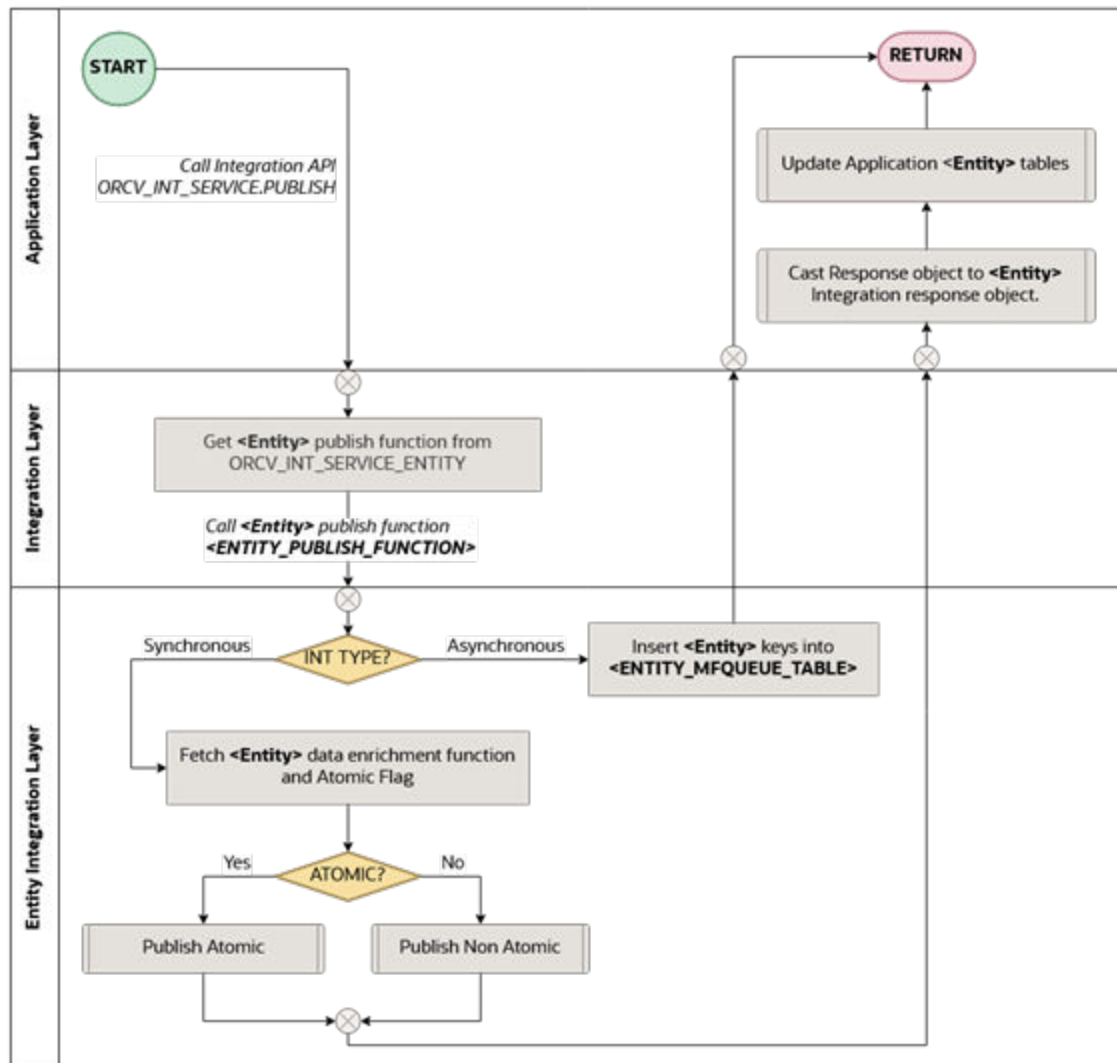
Process Overview

This section provides a process overview.

Publish Flow

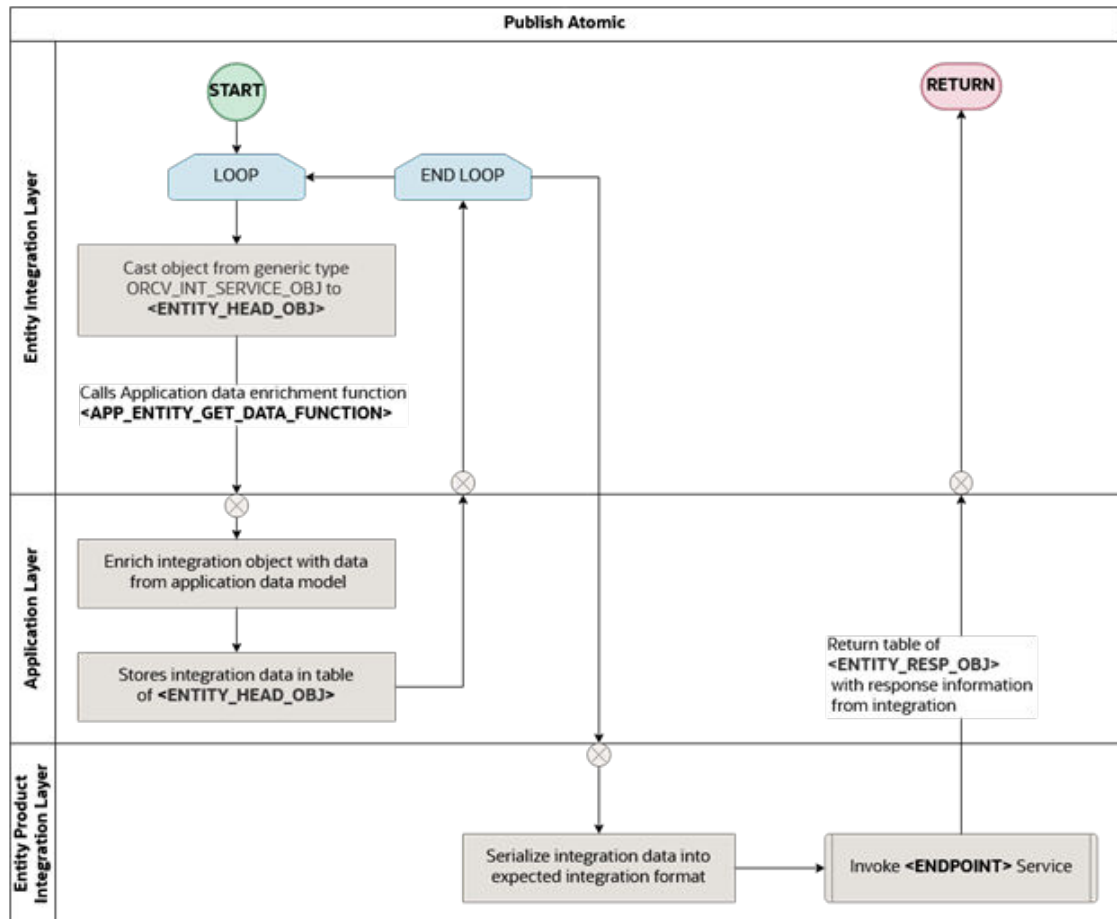
Publish allows the call of integration endpoints in synchronous mode. Those calls can be either atomic or non-atomic. Atomic is in case all the entries need to be integrated together in bulk while the non-atomic deals with each entry individually.

Figure 27-23 Database Integration Framework Publish Flow



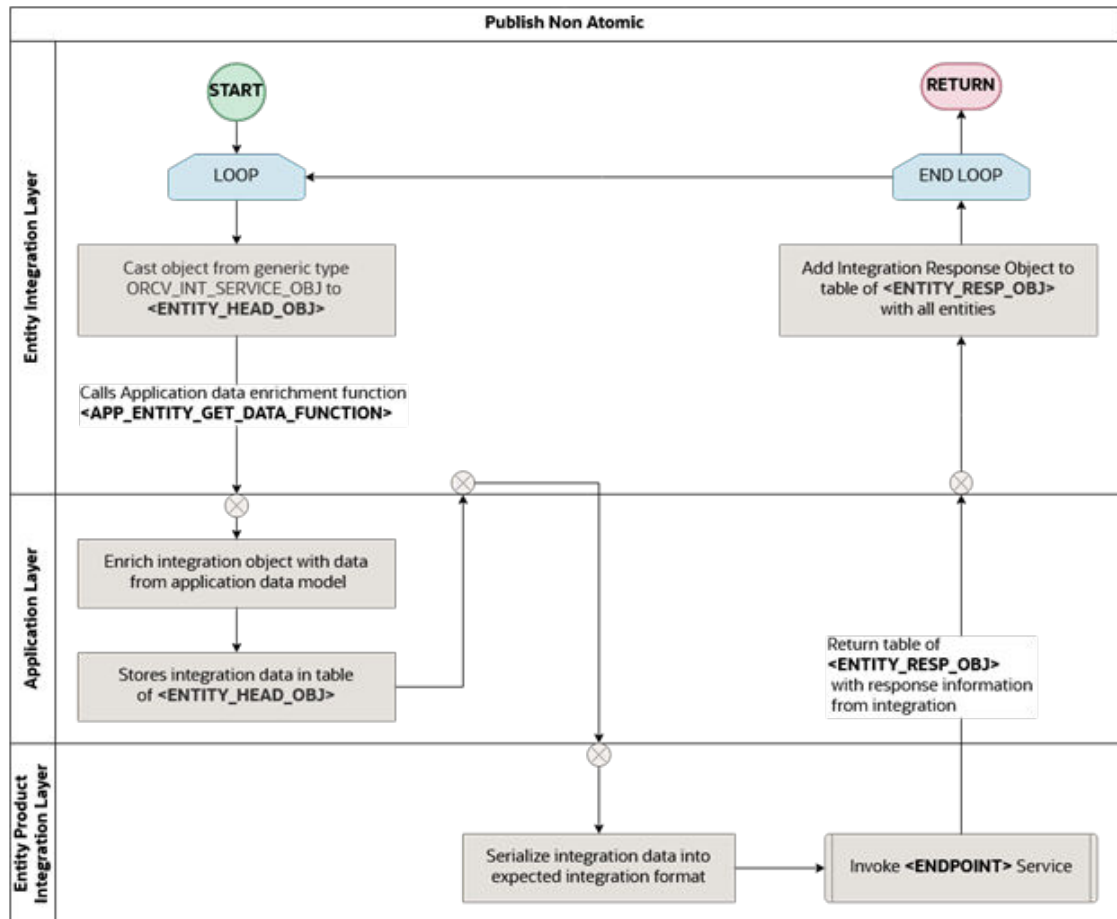
Publish Atomic

Figure 27-24 Database Integration Framework Publish Flow Atomic



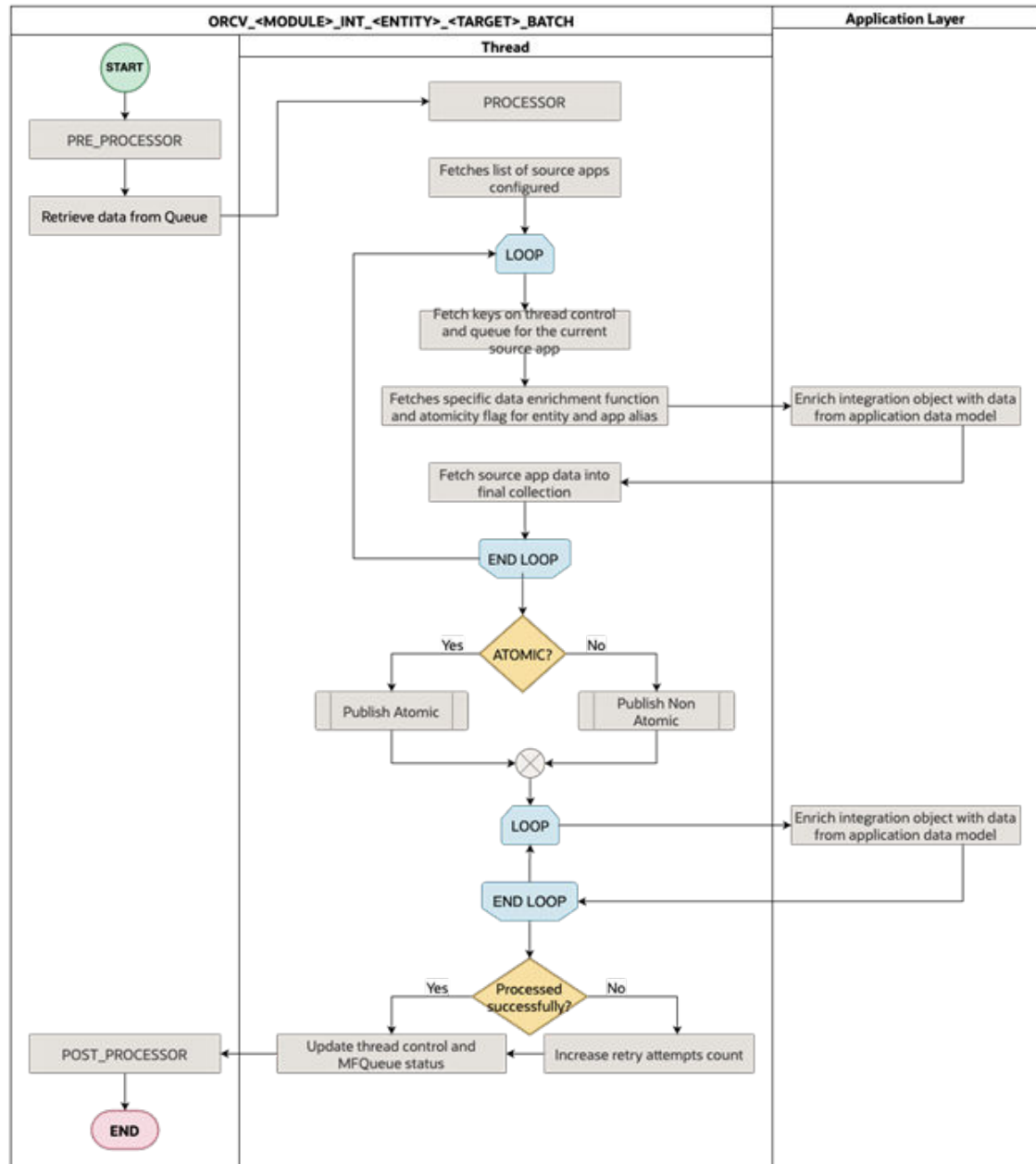
Publish Non Atomic

Figure 27-25 Database Integration Framework Publish Flow Non-Atomic



Publish Asynchronous Batch

Figure 27-26 Database Integration Framework Publish Asynchronous



Configuration

Publish API is available as the `ORCV_INT_SERVICE.PUBLISH` function that handles the generic publication process.

Publish Function

ORCV_INT_SERVICE.PUBLISH function is a generic call that handles the generic publication process and returns Boolean, true in case of success and false otherwise.

Parameter	In/Out	Parameter Type	Description
O_error_message	OUT	ORCV_LOGS.BACKTRACE_DESC%TYPE	Error message in case of error during function call
O_service_result	OUT	ORCV_INT_SERVICE_RESULT	TBL with the output of the integration process
I_service_keys_tbl	IN	ORCV_INT_SERVICE_TBL	TBL of the keys of the entities to be integrated.
I_entity_id	IN	ORCV_INT_SERVICE_ENTITY.ENTITY_ID%TYPE	Integration Entity ID from ORV_INT_SERVICE_ENTITY
I_app_alias	IN	VARCHAR2	Application Alias
I_integration_type	IN	VARCHAR2	Type of Integration. Accepted values are SYNC, ASYNC, and BATCH.
I_is_retry	IN	BOOLEAN	Flag indicating whether the entity should be attempt to retry if it fails

Table ORCV_INT_SERVICE_ENTITY

This table holds the publication and subscription mappings for synchronous integrations.

Column	Data Type	Description
Entity ID	VARCHAR2(10)	This field contains the integration entity.
Publication Function	VARCHAR2(61)	This field contains the function responsible for publishing information to the base product.

Table ORCV_INT_SERVICE_MAPPING

This table holds mapping information for synchronous integrations to fetch data from.

Column	Data Type	Description
Application Alias	VARCHAR2(10)	This field contains the app alias mapping to the entity.
Application Entity Get Data Function	VARCHAR2(61)	This field contains the function responsible for fetching data.
Application Status Function	VARCHAR2(61)	This field contains the function responsible setting application data status.
Atomic Indicator	VARCHAR2(1)	This field contains the indication about whether the integration process should be processed in an atomic way: "N" for entries to be processed individually, "Y" for entries to be processed in bulk.

Integration Object Type

ORCV_INT_SERVICE_OBJ is a base object type for handling the reference keys sent for integration and each Entity must be under this object following the entity specific requirements.

Field	Data Type	Description
SEQ_NO	NUMBER(10)	Sequence that can define the order of entries.
ACTION	VARCHAR2(6)	Action to be taken for the entry.
EXT_SOURCE	VARCHAR2(50)	Source of the Integration Entity.
EXT_REF_NO_1	VARCHAR2(15)	External Reference Number 1
EXT_REF_NO_2	VARCHAR2(15)	External Reference Number 2
EXT_REF_NO_3	VARCHAR2(15)	External Reference Number 3
EXT_REF_NO_4	VARCHAR2(15)	External Reference Number 4
KEY_1_VARCHAR	VARCHAR2(15)	Entity Varchar2 identifier 1.
KEY_1_NUMBER	NUMBER(15)	Entity Number identifier 1.
KEY_1_DATE	DATE	Entity Date identifier 1.
KEY_2_VARCHAR	VARCHAR2(15)	Entity Varchar2 identifier 2.
KEY_2_NUMBER	NUMBER(15)	Entity Number identifier 2.
KEY_2_DATE	DATE	Entity Date identifier 2.
KEY_3_VARCHAR	VARCHAR2(15)	Entity Varchar2 identifier 3.
KEY_3_NUMBER	NUMBER(15)	Entity Number identifier 3.
KEY_3_DATE	DATE	Entity Date identifier 3.
KEY_4_VARCHAR	VARCHAR2(15)	Entity Varchar2 identifier 4.
KEY_4_NUMBER	NUMBER(15)	Entity Number identifier 4.
KEY_4_DATE	DATE	Entity Date identifier 4.
EXT_GROUP_NO	VARCHAR2(15)	External Group Number

Integration Response Object Type

ORCV_INT_SERVICE_RES_OBJ is a base object for the handling the response of the integration process.

Field	Data Type	Description
STATUS	VARCHAR2(2)	Response Status Code.
MESSAGE	VARCHAR2(50)	Response Message Body.
EXT_SOURCE	VARCHAR2(50)	Source of the Integration Entity.
SEQ_NO	NUMBER(10)	Sequence that can define the order of entries.
RETRY_ATTEMPT_COUNT	NUMBER(3)	Number of retry attempts.
EXT_REF_NO_1	VARCHAR2(15)	External Reference Number 1.
EXT_REF_NO_2	VARCHAR2(15)	External Reference Number 2.
EXT_REF_NO_3	VARCHAR2(15)	External Reference Number 3.
EXT_REF_NO_4	VARCHAR2(15)	External Reference Number 4.
KEY_1_VARCHAR	VARCHAR2(15)	Entity Varchar2 identifier 1.
KEY_1_NUMBER	NUMBER(15)	Entity Number identifier 1.
KEY_1_DATE	DATE	Entity Date identifier 1.
KEY_2_VARCHAR	VARCHAR2(15)	Entity Varchar2 identifier 2.
KEY_2_NUMBER	NUMBER(15)	Entity Number identifier 2.
KEY_2_DATE	DATE	Entity Date identifier 2.
KEY_3_VARCHAR	VARCHAR2(15)	Entity Varchar2 identifier 3.
KEY_3_NUMBER	NUMBER(15)	Entity Number identifier 3.

Field	Data Type	Description
KEY_3_DATE	DATE	Entity Date identifier 3.
KEY_4_VARCHAR	VARCHAR2(15)	Entity Varchar2 identifier 4.
KEY_4_NUMBER	NUMBER(15)	Entity Number identifier 4.
KEY_4_DATE	DATE	Entity Date identifier 4.
EXT_GROUP_NO	VARCHAR2(15)	External Group Number.

Entity Publish Function

This is a generic definition of the function being called by ORCV_INT_SERVICE.PUBLISH that needs to be implemented for each integration entity. It must follow the specification below and returns Boolean true in case of success and false otherwise.

Parameter	In/Out	Parameter Type	Description
O_error_message	OUT	ORCV_LOGS.BACKTRACE_DESC%TYPE	Error message in case of error during function call
O_service_result	OUT	ORCV_INT_SERVICE_RESULT	Table with the output of the integration process
I_service_keys_table	IN	ORCV_INT_SERVICE_TBL	Table of the keys of the entities to be integrated.
I_app_alias	IN	VARCHAR2	Application Alias
I_integration_type	IN	VARCHAR2	Type of Integration. Accepted values are SYNC, ASYNC, and BATCH.
I_is_retry	IN	BOOLEAN	Flag indicating if the entity should be attempted to retry in case it fails

Application Entity Get Data Function

This is a generic definition of the function being called by ORCV_<ENTITY>_SERVICE.PUBLISH that retrieves the entity data based on application data. It must follow the specification below and returns Boolean true in case of success and false otherwise.

Parameter	In/Out	Parameter Type	Description
O_error_message	OUT	ORCV_LOGS.BACKTRACE_DESC%TYPE	Error message in case of error during function call
IO_int_service_obj	IN/OUT	ORCV_INT_<ENTITY>OBJ	TBL of Entity to be populated with data by the function. The object type needs to be under ORCV_INT_SERVICE_OBJ

Entity Configuration Table

ORCV_INT_SERVICE_ENTITY table holds the publication and subscription mappings for synchronous integrations.

Column	Data Type	Nullable	Comment
ENTITY_ID	VARCHAR2(10)	N	This field contains the integration entity.

Column	Data Type	Nullable	Comment
PUBLICATION_FUNCTION	VARCHAR2(61)	Y	This field contains the function responsible for publishing information to the base product.
SUBSCRIPTION_FUNCTION	VARCHAR2(61)	Y	This field contains the function responsible for subscribing information from the base product.

Integration Service Mapping Table

ORCV_INT_SERVICE_MAPPING table holds mapping information for synchronous integrations to fetch data from the Application.

Column	Data Type	Nullable	Comment
ENTITY_ID	VARCHAR2(10)	N	This field contains the integration entity.
APP_ALIAS	VARCHAR2(10)	N	This field contains the app alias mapping to the entity.
DATA_FUNCTION	VARCHAR2(61)	N	This field contains the function responsible for fetching data.
ATOMIC_IND	VARCHAR2(1)	N	This field contains the indication about whether the integration process should be processed in an atomic way: "N" for entries to be processed individually, "Y" for entries to be processed in bulk.

Example Configuration

In the example below is the configuration for RTV Entity used by Quality Control.

Configuration	Value	Configuration Table
Entity ID	RTV	ORCV_INT_SERVICE_ENTITY
Publication Function	ORCV_INV_INT_RTV_SERVICE.PUBLISH	ORCV_INT_SERVICE_ENTITY
Subscription Function		ORCV_INT_SERVICE_ENTITY
Application Alias	QLCT	ORCV_INT_SERVICE_MAPPING
Application Entity Get Data Function	ORCV_QLCT_PUB_RTV.GET_RTV_INTEGRATION_DATA	ORCV_INT_SERVICE_MAPPING
Atomic Indicator	N	ORCV_INT_SERVICE_MAPPING
Entity Integration Object	ORCV_INV_INT_RTV_HEAD_OBJ	N/A

Interactive Grid Additional Columns

Oracle APEX Interactive Grids can be extended to have additional reference data available, it is possible by replacing default views that are available as placeholders out of the box.

To use this functionality, it is necessary to change the provided "NB" views, keeping the Key columns and the reference columns names, and set the HIDDEN_ID to "N" on ORCV_IG_HIDDEN_COLUMN table.

The configuration of each of the relevant “NB” views and grid details can be found on each application’s implementation operations guide.

Database Objects

This section describes database objects.

Tables

Name	Description
ORCV_IG_HIDDEN_COLUMN	This table holds the columns that should be hidden for each application, grid and user role. The column ORCV_IG_HIDDEN_COLUMN must be set to true so it is visible and column ORCV_IG_HIDDEN_COLUMN can be changed to the required column header.
ORCV_IG_HIDDEN_COLUMN_TL	This table holds the columns translations that should be hidden for each application, grid and user role.

Views

All views defined are defined with the naming convention ORCV_V_<APP_ALIAS>_<ENTITY>_NB and must comply with the format below:

Column	Data Type	Notes
<KEY_COLUMN_N >	NUMBER / VARCHAR2 / DATE	The number of key columns vary depending on the number of columns for the entity, it must align with the base table being replaced.
REF_1	VARCHAR2(255)	The reference column 1.
REF_2	VARCHAR2(255)	The reference column 2.
REF_3	VARCHAR2(255)	The reference column 3.
REF_4	VARCHAR2(255)	The reference column 4.
REF_5	VARCHAR2(255)	The reference column 5.
REF_6	VARCHAR2(255)	The reference column 6.
REF_LINK	VARCHAR2(4000)	The reference link to be added.
REF_LINK_TEXT	VARCHAR2(255)	The text to be added to the REF_LINK column.

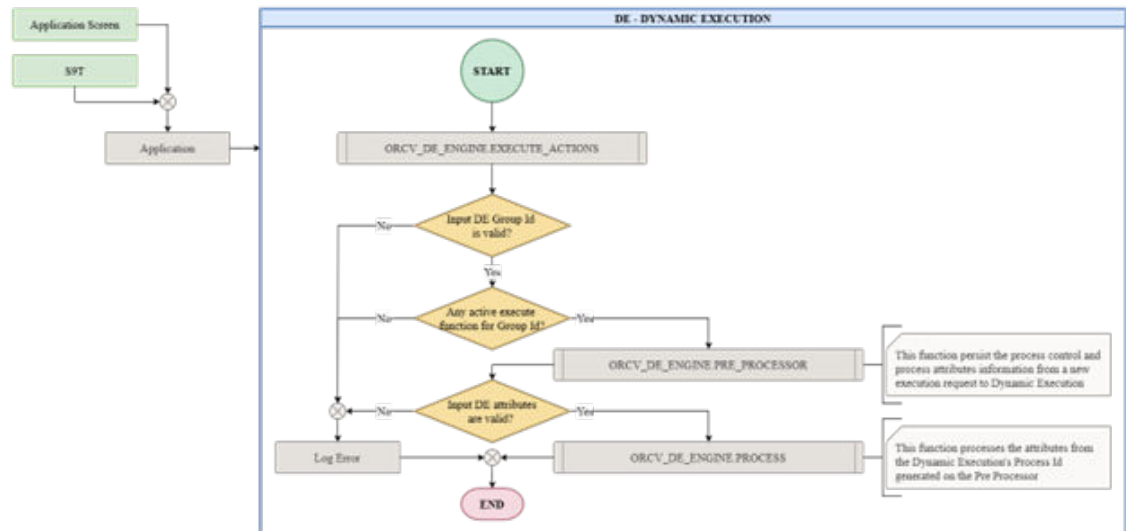
Dynamic Execution Framework

Retail Supply Chain Hub applications flows are designed and developed in code. The purpose of Dynamic Execution Framework is to provide an easy way to customize the process flow of an application:

- Add Additional Steps
- Disable Steps
- Modify Validations
- Modify Permissions to execute Steps
- Use User Groups to manage permissions to user actions
- Enable/Disable action buttons based on Status/User permissions

Process Overview

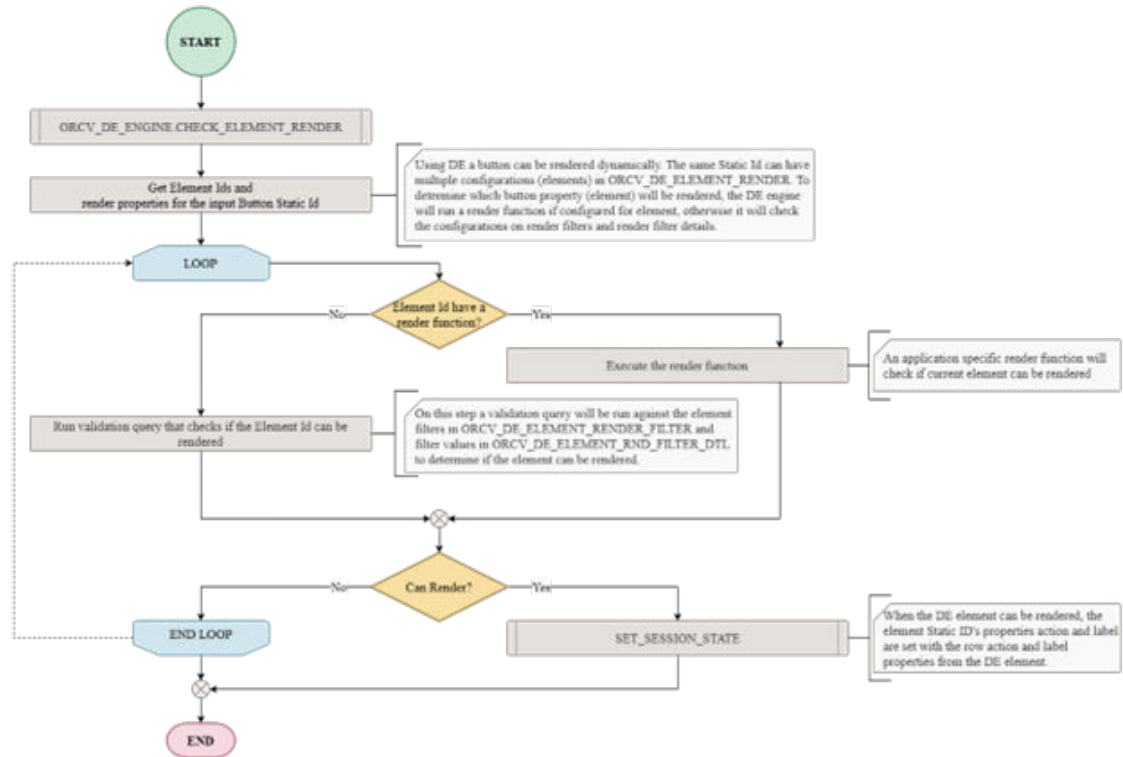
Figure 27-27 Dynamic Execution Framework Process Overview



Dynamic Button Rendering Overview

Buttons can be divided in two types, static and dynamic. Static buttons always have the same label and row action and they are defined on the application itself. Dynamic buttons may vary their label and row action according to their render condition. Dynamic buttons need to have two support page items <button_static_id>_LBL and <button_static_id>_ACTION to store their label and row action in the page. Bellow there is a diagram with the button dynamic rendering using the Dynamic Execution engine.

Figure 27-28 Dynamic Execution Framework Dynamic Button Rendering Overview



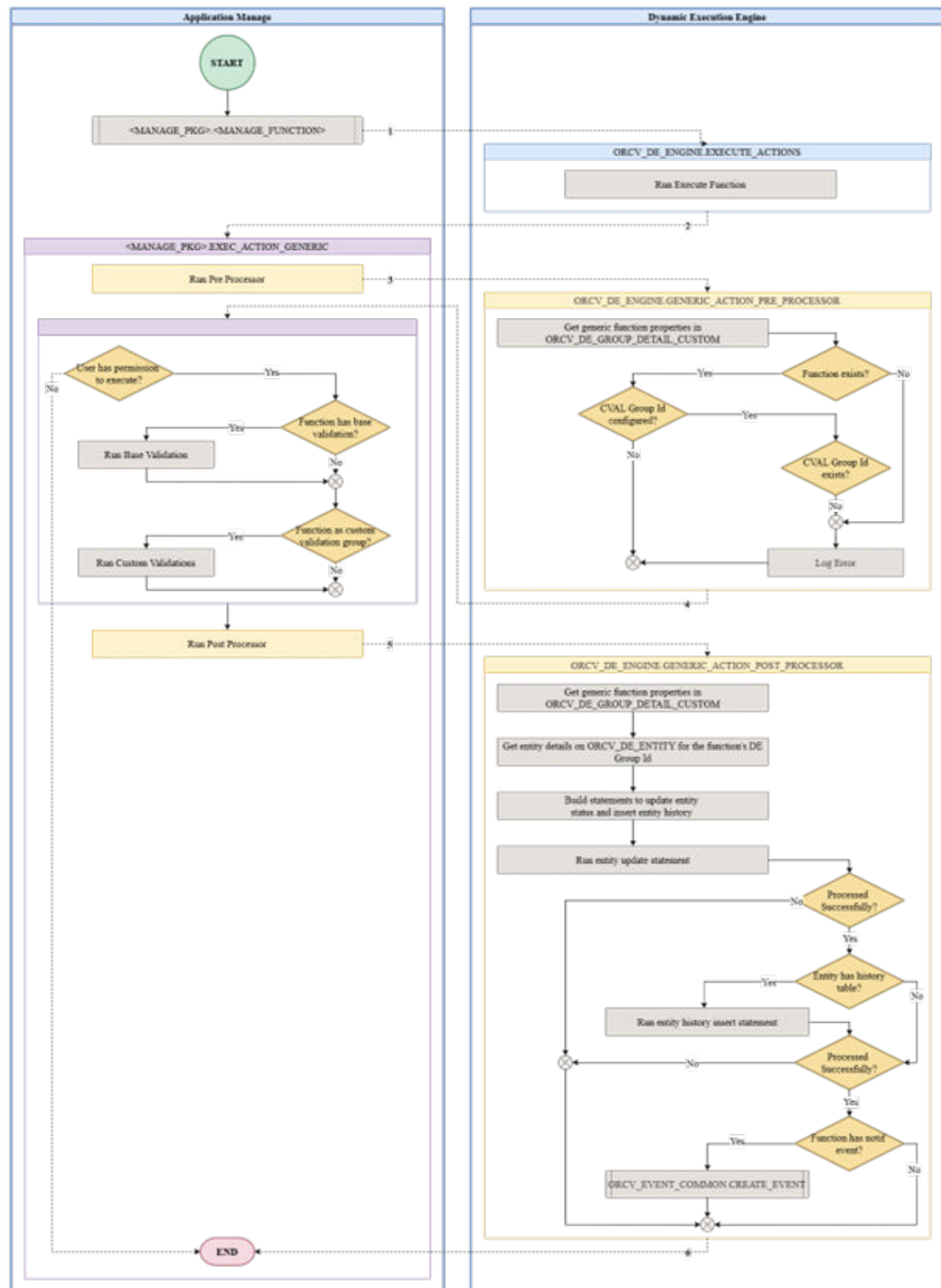
Extended Application Flow Helpers

The Application states flow extension using Dynamic Execution can be done in two ways:

- Develop a new function(s) from scratch that handles the behavior of the new state(s)
- Use the existing pre built helper functions available in DE and the Generic Function built on the Application side.
 - For cases where the new state just has the objective of changing the status of the entity, this approach does not require a new execute function.
 - To make this option available, when integrating the Dynamic Execution Engine into an Application, the generic function development must be coded in the Application side and use DE pre and post processor for generic steps.

When changing the status of an entity data integrity must be enforced from the previous state and run the base validation and custom validations that assure that guarantee.

Figure 27-29 Dynamic Execution Framework Extended Application Flow Helpers Overview



Database Objects

This section describes database objects.

Tables

Name	Description
ORCV_DE_ENTITY	Available entities in Dynamic Execution
ORCV_DE_ENTITY_KEY	Key column(s) definition for DE entities/entity tables
ORCV_DE_ENTITY_STEPS	Available steps for each DE Entity
ORCV_DE_GROUP_HEAD	DE Group header information
ORCV_DE_GROUP_DETAIL	Functions to be executed for DE groups
ORCV_DE_GROUP_DETAIL_CUSTOM	Custom Functions to be executed for DE groups
ORCV_DE_GROUP_DTL_COND	Configuration of valid steps for each group detail
ORCV_DE_GROUP_USER_GROUPS	User group's permissions for the group steps
ORCV_DE_ELEMENT_RENDER	Dynamic render conditions for application elements
ORCV_DE_ELEMENT_RENDER_FILTER	Dynamic render elements filters
ORCV_DE_ELEMENT_RND_FILTER_DETAIL	Dynamic render element filter parameters
ORCV_DE_PROCESS_CONTROL	Logs information about group execution
ORCV_DE_PROCESS_CTRL_EXE_CND	Configuration of valid steps for each group detail at the time of the DE execution request
ORCV_DE_PROCESS_PARAMETERS	Keeps information about the process parameters of a DE execution request
ORCV_DE_PROCESS_ATTRIBUTES	Keeps information about the attributes to be processed on a DE execution request
ORCV_DE_PROCESS_ATTR_CONTROL	Logs information about process attributes execution
ORCV_DE_PROCESS_AUX_ATTRIBUTES	Keeps information about the auxiliar attributes to be processed on a DE execution request

Views

Name	Description
ORCV_V_DE_PROC_ATTR_CTRL_PRG	View to be used to purge old data on DE process attributes control table
ORCV_V_DE_PROC_CTRL_EXEC_PRG	View to be used to purge old data on DE process control execution table
ORCV_V_DE_PROCESS_ATTR_PRG	View to be used to purge old data on DE process attributes table
ORCV_V_DE_PROCESS_AUX_ATTR_PRG	View to be used to purge old data on DE process aux attributes table
ORCV_V_DE_PROCESS_CONTROL_PRG	View to be used to purge old data on DE process control table
ORCV_V_DE_PROCESS_PARAM_PRG	View to be used to purge old data on DE process parameters table

Sequences

Name	Description
ORCV_DE_PROCESS_SEQ	Sequence to be used on a new process execution instance

Types

Name	Description
ORCV_DE_ACTION_OBJ	Represents an action/function to be executed on a DE process
ORCV_DE_ATTRIBUTES_CONTROL_OBJ	Represents the DE process attributes control data
ORCV_DE_ATTRIBUTES_OBJ	Represents the DE process auxiliar attributes data
ORCV_DE_AUX_ATTRIBUTES_OBJ	Represents the DE process auxiliar attributes data
ORCV_DE_ENTITY_KEY_OBJ	Represents the DE entity data
ORCV_DE_GROUP_DTL_CUSTOM_OBJ	Represents DE custom functions data
ORCV_DE_PARAMETERS_OBJ	Represents the DE process parameters data

Packages

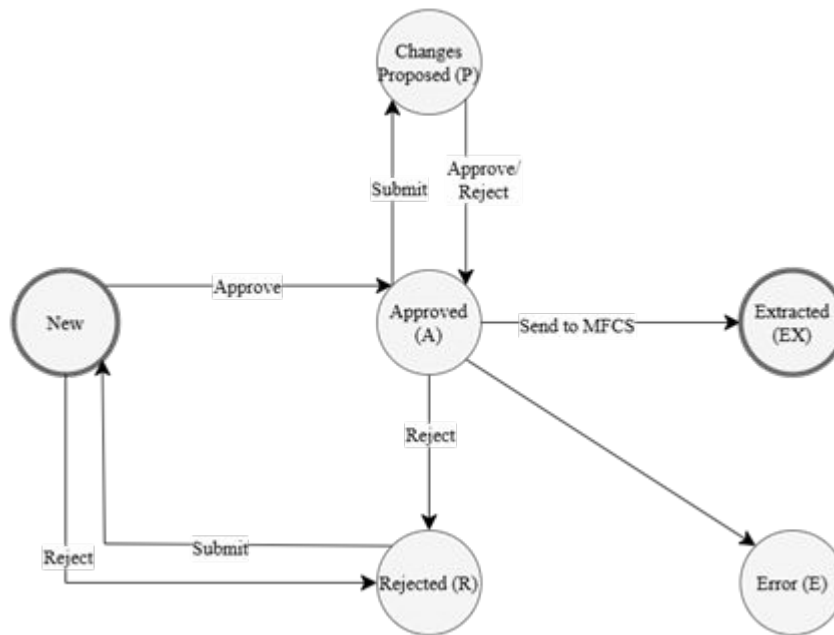
Table	Description
ORCV_DE_GV	Package that holds global variables
ORCV_DE_ERR	Package that holds keys for message errors
ORCV_DE_ENGINE	Package to manage dynamic executions

Configuration Example - Cost Changes Management

This example configuration is based on Cost Changes Management application available in Retail Supply Chain Hub and follows the below base workflow.

Before the creation of DE configurations for SPCC it is necessary to understand the status flow of the application. Below there a simplified version of SPCC status flow.

Figure 27-30 Dynamic Execution Framework Configuration Example Cost Changes Management Base Flow

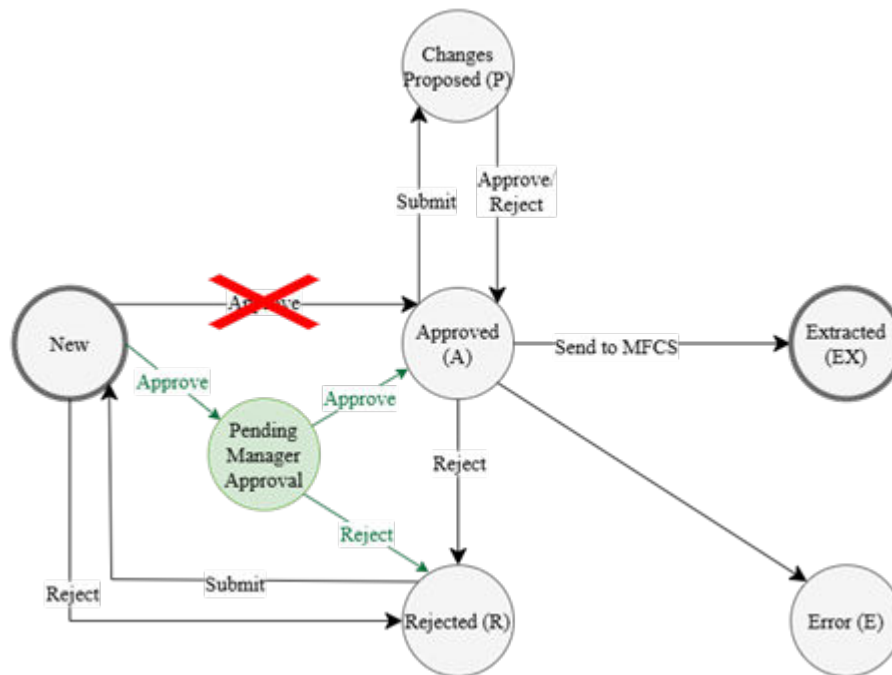


For Custom Flows it's possible to:

- Rename status: some Status should not be changed as they are part of the integral functionality of the Application, however they can be renamed if required. In Cost Changes Management: New and Approved
- Add Additional Steps:
 - If the additional steps only require validation (using Custom Validation Groups) and security control can be configured using a pre-existing generic function and setup.
 - * Generic Action Function: ORCV_SPCC_COSTCHG.EXEC_ACTION_GENERIC
 - * Security can be set at Role Level or Role + User Group Level to further detail permissions (Default workflow uses Role Level).
 - If not, then a custom function can be used and configured to be used instead

Example Custom Flow:

Figure 27-31 Dynamic Execution Framework Configuration Example Cost Changes Management Custom Flow



Step from New to Approved was removed and an additional step was added in the middle that requires an additional approval.

ORCV_CODE_DETAIL

Add a new Status code for the new flow.

CODE_TYPE	CODE	CODE_DESC
CCST	PM	Pending Manager Approval

ORCV_SEC_GROUPS

Add two additional groups to enable permissions for the additional Status.

GROUP_ID	GROUP_DESC
SPCC_APPR_MNGR	SPCC Approval Managers
SPCC_SUBMIT_USER	SPCC Submit Users

ORCV_DE_GROUP_HEAD

Enable the user group security by setting APPLY_USR_GRP_SEC_IND to Y.

ORCV_DE_ENTITY_STEPS

ENTITY_ID	STEP_ID	FROM_STATUS	ROW_ACTION	ACTIVE_IND
COSTCHG	STEP_PM_A	PM	A	A
COSTCHG	STEP_PM_M	PM	A	A
COSTCHG	STEP_PM_R	PM	A	A

ORCV_DE_GROUP_DETAIL

DETAIL_ID	GROUP_ID	EXEC_FUNCTION	EXEC_FUNCTION_DESC	SCOPE_FUNCTION	SEQ_NO	ACTIVE_IND	CUSTOM_IND
SPCC_EXEC_ACTION_SUBMIT	SPCC_COST_MGMT	ORCV_SPCC_COSTCHG.EXEC_ACTION_SUBMIT	Submit Cost Change	NA	10	A	N
SPCC_EXEC_ACTION_REJECT	SPCC_COST_MGMT	ORCV_SPCC_COSTCHG.EXEC_ACTION_REJECT	Reject Cost Change	NA	20	A	N
SPCC_EXEC_ACTION_N_APPROVE	SPCC_COST_MGMT	ORCV_SPCC_COSTCHG.EXEC_ACTION_GENERIC	Approve New Cost Change	NA	30	A	N
SPCC_EXEC_ACTION_APPROVE	SPCC_COST_MGMT	ORCV_SPCC_COSTCHG.EXEC_ACTION_APPROVE	Approve Cost Change	NA	31	A	N

ORCV_DE_GROUP_DETAIL_CUSTOM

DETAIL_ID	TO_STATUS	ROW_ACTION	HIST_REASON_CODE	VALIDATION_FUNCTION	CVAL_GROUP_ID	NOTIFICATION_TYPE	PERMISSION_TYPE
SPCC_EXEC_ACTION_N_APPROVE	PM	A	APRVCC	ORCV_SPCC_COSTCHG_VAL.VALIDATE_APPROVE	SPCC_COSTCHG_APPROVE	NA	APPR

ORCV_DE_GROUP_DTL_EXE_COND

DETAIL_ID	ENTITY_ID	STEP_ID
SPCC_EXEC_ACTION_SUBMIT	COSTCHG	STEP_PM_M
SPCC_EXEC_ACTION_REJECT	COSTCHG	STEP_PM_R
SPCC_EXEC_ACTION_APPROVE	COSTCHG	STEP_PM_A
SPCC_EXEC_ACTION_N_APPROVE	COSTCHG	STEP_N_A
SPCC_EXEC_ACTION_N_APPROVE	COSTCHG	STEP_N_M
SPCC_EXEC_ACTION_N_APPROVE	COSTCHG	STEP_R_M
SPCC_EXEC_ACTION_N_APPROVE	COSTCHG	STEP_NULL_C
SPCC_EXEC_ACTION_SUBMIT	COSTCHG	STEP_NULL_C
SPCC_EXEC_ACTION_SUBMIT	COSTCHG	STEP_N_M
SPCC_EXEC_ACTION_SUBMIT	COSTCHG	STEP_R_M

DETAIL_ID	ENTITY_ID	STEP_ID
SPCC_EXEC_ACTION_SUBMIT	COSTCHG	STEP_A_M
SPCC_EXEC_ACTION_SUBMIT	COSTCHG	STEP_P_M
SPCC_EXEC_ACTION_REJECT	COSTCHG	STEP_N_R
SPCC_EXEC_ACTION_REJECT	COSTCHG	STEP_P_R
SPCC_EXEC_ACTION_REJECT	COSTCHG	STEP_A_R
SPCC_EXEC_ACTION_APPROVE	COSTCHG	STEP_P_M
SPCC_EXEC_ACTION_APPROVE	COSTCHG	STEP_A_M
SPCC_EXEC_ACTION_APPROVE	COSTCHG	STEP_P_A

ORCV_DE_GROUP_USER_GROUPS

GROUP_ID	USER_GROUP_ID	ENTITY_ID	STEP_ID
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_A_M
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_A_R
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_NULL_C
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_N_A
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_N_M
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_N_R
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_PM_A
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_PM_M
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_PM_R
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_P_A
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_P_M
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_P_R
SPCC_COST_MGMT	SPCC_APP_MNGR	COSTCHG	STEP_R_M
SPCC_COST_MGMT	SPCC_SUMBIT_USER	COSTCHG	STEP_A_M
SPCC_COST_MGMT	SPCC_SUMBIT_USER	COSTCHG	STEP_A_R
SPCC_COST_MGMT	SPCC_SUMBIT_USER	COSTCHG	STEP_NULL_C
SPCC_COST_MGMT	SPCC_SUMBIT_USER	COSTCHG	STEP_N_A
SPCC_COST_MGMT	SPCC_SUMBIT_USER	COSTCHG	STEP_N_M
SPCC_COST_MGMT	SPCC_SUMBIT_USER	COSTCHG	STEP_N_R
SPCC_COST_MGMT	SPCC_SUMBIT_USER	COSTCHG	STEP_P_A
SPCC_COST_MGMT	SPCC_SUMBIT_USER	COSTCHG	STEP_P_M
SPCC_COST_MGMT	SPCC_SUMBIT_USER	COSTCHG	STEP_P_R
SPCC_COST_MGMT	SPCC_SUMBIT_USER	COSTCHG	STEP_R_M

ORCV_DE_ELEMENT_RENDER

ELEMENT_ID	ELEMENT_D ESC	APP_ALI AS	PAGE_ NO	STATIC_ID	ROW_ACTI ON	LABE L
SPCC_P200_B_APPROVE	Approve button.	SPCC	200	P200_B_APPROVE	A	Approve
SPCC_P200_B_SAVE	Save button.	SPCC	200	P200_B_SAVE	NA	NA
SPCC_P200_B_EDIT	Edit button.	SPCC	200	P200_B_EDIT	NA	NA

ELEMENT_ID	ELEMENT_DESCRIPTION	APP_ALIAS	PAGE_NO	STATIC_ID	ROW_ACTION	LABEL
SPCC_P200_B_REJECT	Reject button.	SPCC	200	P200_B_REJECT	NA	NA
SPCC_P200_B_APPROVE_PM	Approve button for PM Status.	SPCC	200	P200_B_APPROVE	A	Approve
SPCC_P200_B_EDIT_PM	Edit button for PM Status.	SPCC	200	P200_B_EDIT	NA	NA
SPCC_P200_B_REJECT_PM	Reject button for PM Status.	SPCC	200	P200_B_REJECT	NA	NA

ORCV_DE_ELEMENT_RENDER_FILTER

ELEMENT_ID	PARAMETER_NAME	PARAMETER_TYPE	PARAMETER_DELIMITER
ELEMENT_ID	PARAMETER_NAME	PARAMETER_TYPE	PARAMETER_DELIMITER
SPCC_P200_B_EDIT_PM	P200_MODE	VARCHAR	NA
SPCC_P200_B_EDIT_PM	P200_USER_PERM	VARCHAR	NA
SPCC_P200_B_EDIT_PM	P200_STATUS	VARCHAR	NA
SPCC_P200_B_EDIT_PM	P200_I_USER_GROUPS	VARCHAR	:
SPCC_P200_B_REJECT_PM	P200_MODE	VARCHAR	NA
SPCC_P200_B_REJECT_PM	P200_USER_PERM	VARCHAR	NA
SPCC_P200_B_REJECT_PM	P200_STATUS	VARCHAR	NA
SPCC_P200_B_REJECT_PM	P200_I_USER_GROUPS	VARCHAR	:
SPCC_P200_B_APPROVE_PM	P200_MODE	VARCHAR	NA
SPCC_P200_B_APPROVE_PM	P200_USER_PERM	VARCHAR	NA
SPCC_P200_B_APPROVE_PM	P200_STATUS	VARCHAR	NA
SPCC_P200_B_APPROVE_PM	P200_I_USER_GROUPS	VARCHAR	:
SPCC_P200_B_APPROVE	P200_I_USER_GROUPS	VARCHAR	:
SPCC_P200_B_REJECT	P200_I_USER_GROUPS	VARCHAR	:
SPCC_P200_B_EDIT	P200_I_USER_GROUPS	VARCHAR	:
SPCC_P200_B_SAVE	P200_I_USER_GROUPS	VARCHAR	:

ORCV_DE_ELEMENT_RND_FILTER_DTL

ELEMENT_ID	PARAMETER_NAME	PARAMETER_VALUE_VARCHAR
SPCC_P200_B_EDIT_PM	P200_MODE	VIEW
SPCC_P200_B_EDIT_PM	P200_USER_PERM	APPR

ELEMENT_ID	PARAMETER_NAME	PARAMETER_VALUE_VARCHAR
SPCC_P200_B_EDIT_PM	P200_STATUS	SPCC_APPRF_MNGR
SPCC_P200_B_EDIT_PM	P200_I_USER_GROUPS	PM
SPCC_P200_B_REJECT_PM	P200_MODE	VIEW
SPCC_P200_B_REJECT_PM	P200_USER_PERM	APPR
SPCC_P200_B_REJECT_PM	P200_STATUS	SPCC_APPRF_MNGR
SPCC_P200_B_REJECT_PM	P200_I_USER_GROUPS	PM
SPCC_P200_B_APPROVE_PM	P200_MODE	VIEW
SPCC_P200_B_APPROVE_PM	P200_USER_PERM	APPR
SPCC_P200_B_APPROVE_PM	P200_STATUS	SPCC_APPRF_MNGR
SPCC_P200_B_APPROVE_PM	P200_I_USER_GROUPS	PM
SPCC_P200_B_APPROVE	P200_I_USER_GROUPS	SPCC_SUBMIT_USER
SPCC_P200_B_APPROVE	P200_I_USER_GROUPS	SPCC_APPRF_MNGR
SPCC_P200_B_REJECT	P200_I_USER_GROUPS	SPCC_SUBMIT_USER
SPCC_P200_B_REJECT	P200_I_USER_GROUPS	SPCC_APPRF_MNGR
SPCC_P200_B_EDIT	P200_I_USER_GROUPS	SPCC_SUBMIT_USER
SPCC_P200_B_EDIT	P200_I_USER_GROUPS	SPCC_APPRF_MNGR
SPCC_P200_B_SAVE	P200_I_USER_GROUPS	SPCC_SUBMIT_USER
SPCC_P200_B_SAVE	P200_I_USER_GROUPS	SPCC_APPRF_MNGR

Integration

This section describes integrations.

Price Change Integration to RPCS

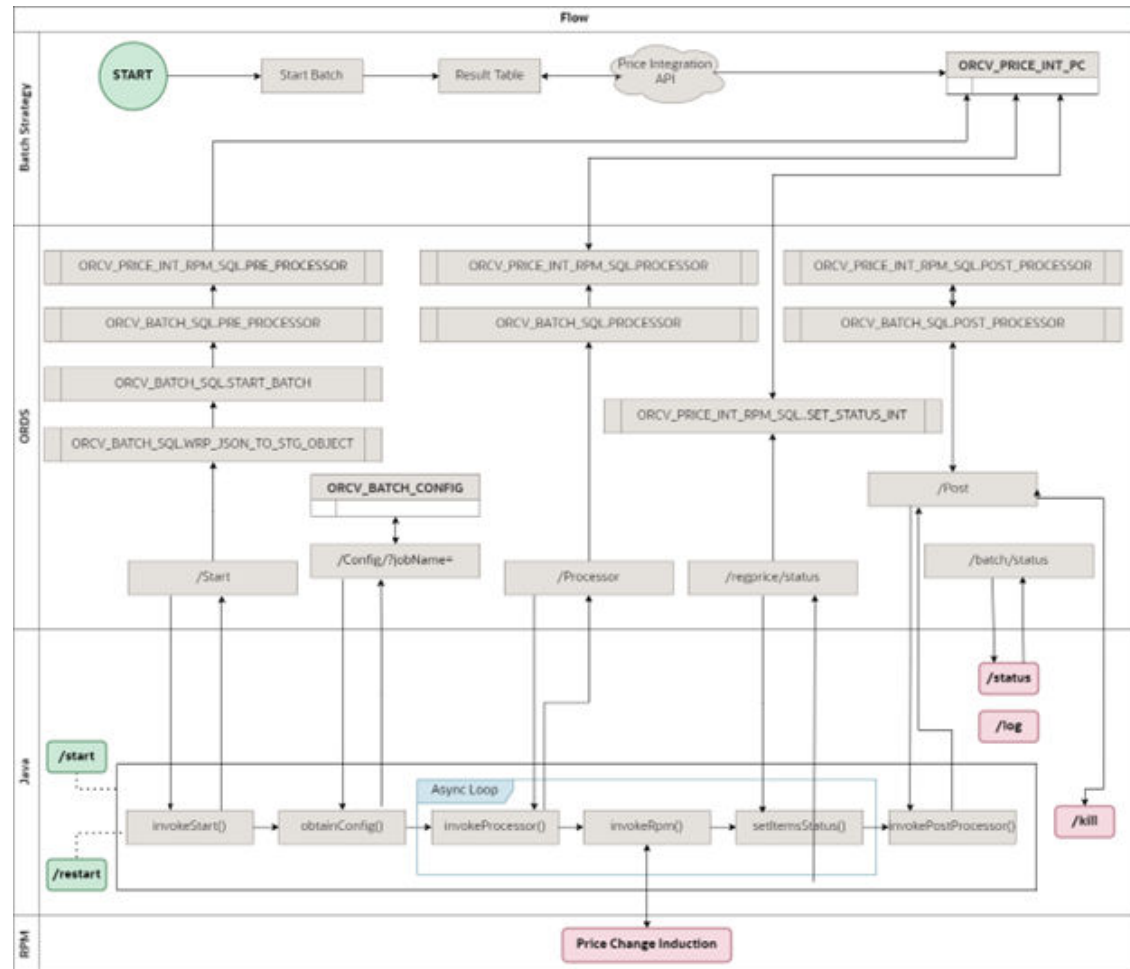
The price change integration batch has a maximum chunk size configuration and has a thread aggregator by effective date and new group batch, which is a reference to the event that generated the price recommendations.

As the chunk size determines the maximum number of records that will be sent in each request to RPCS, this means that in case the number of recommendations for the thread aggregator is higher than the chunk size, the integration will do two requests to RPCS price change induction service. The price changes from the second request will have the price change group description equal to the ones from the first request, but price change group id will be different.

Process Overview

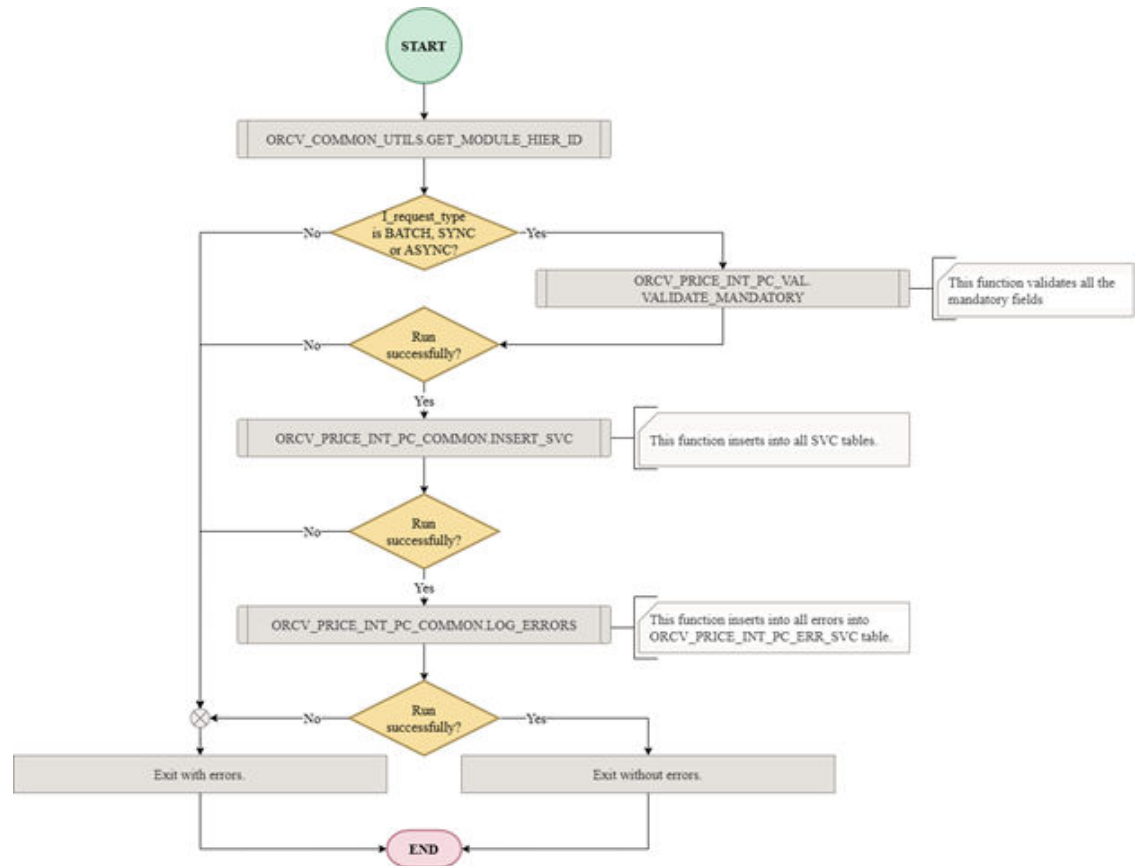
The Price Strategies integration will support the integration of price changes in RPCS through the induction service.

Figure 27-32 Price Change Integration Process Overview



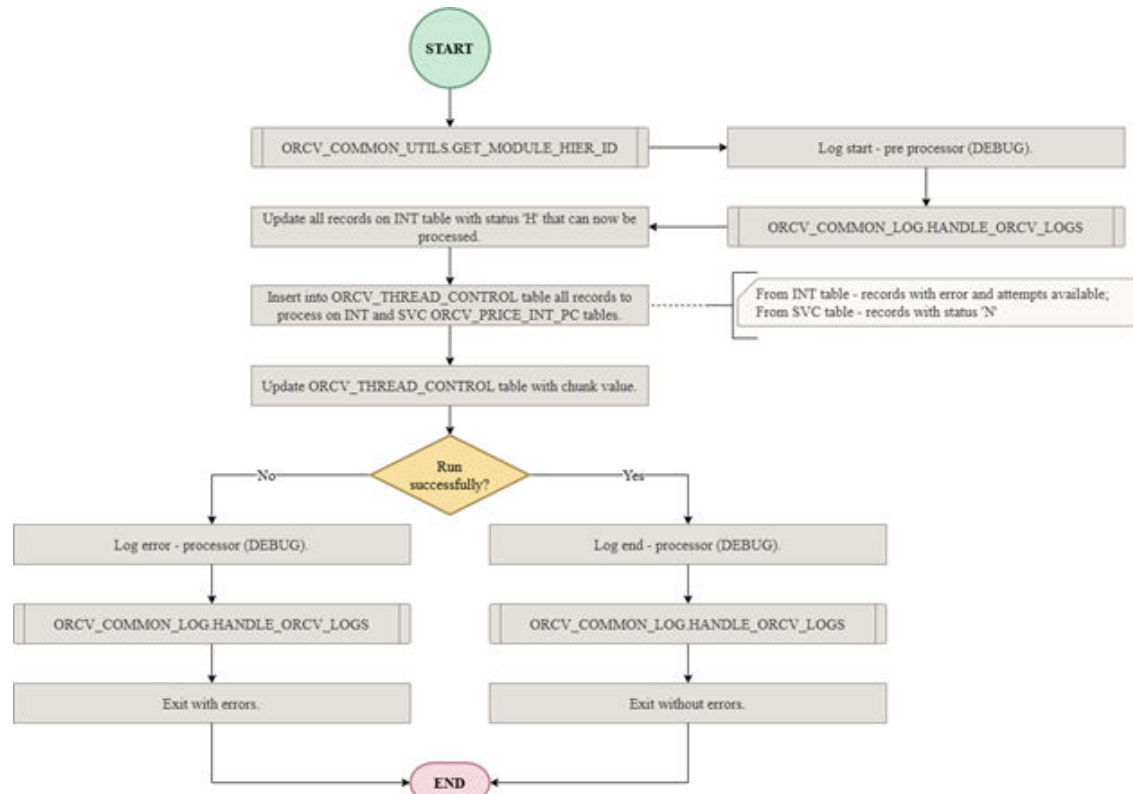
Process Price Changes

Figure 27-33 Price Changes Process Overview



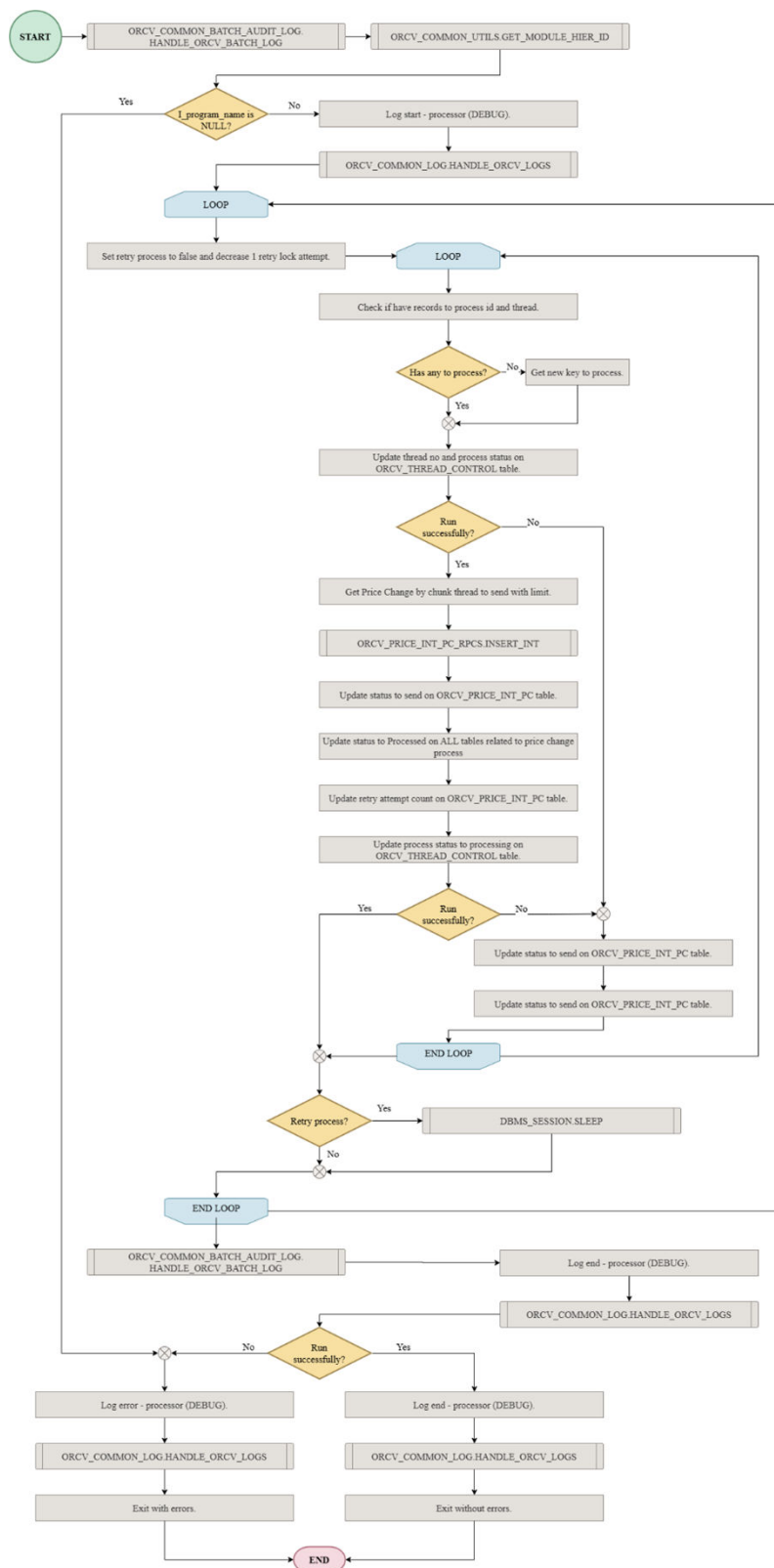
Pre Processor

Figure 27-34 Pre-Processor Price Changes for RPCS Integration



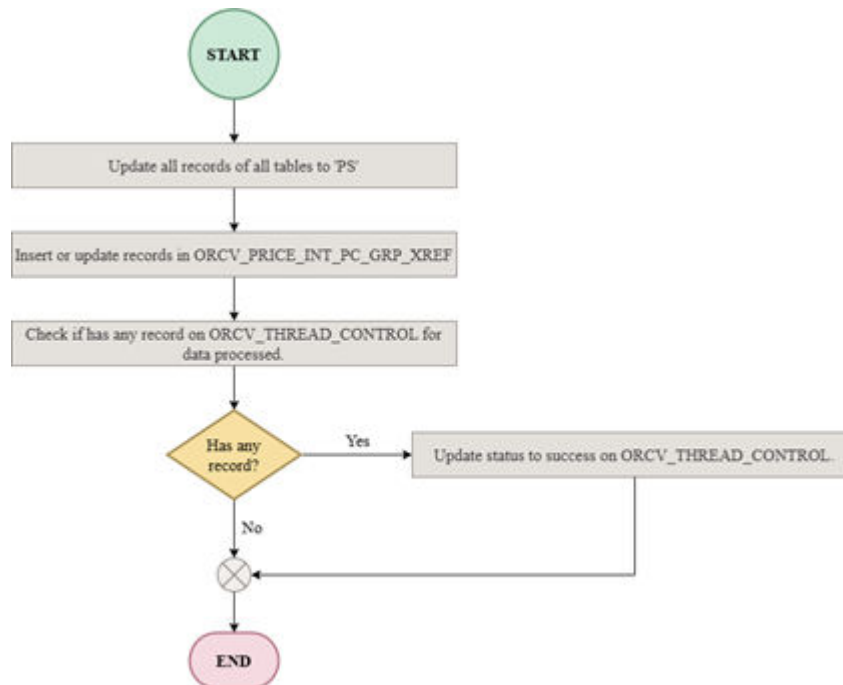
Processor

Figure 27-35 Processor Price Changes for RPCS Integration



Set Integration Status

Figure 27-36 Set Integration Status Flow for Price Change Integration to RPCS



Configuration

This section describes configurations.

Initial Batch Configuration for ORCV_PRICE_INT_PC_RPCS_JOB

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_PRICE_INT_PC_RPCS_JOB
MAX_CHUNK_SIZE	5000
MAX_CONCURRENT_THREADS	8
BULK_LIMIT	1000
RETRY_LOCK_ATTEMPTS	4
RETRY_WAIT_TIME	2
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

System Parameters

Listed below is the list of configurations at system level that are required by this integration.

The parameters are defined in table ORCV_SYSTEM_PARAMETERS.

Area	Parameter	Description	Value 1
AREA	PARAMETER	DESCRIPTION	VALUE_1
BATCH	MAX_CHUNK_SIZE	Default price max chunk size	1000
BATCH	THRESHOLD_INT	Default threshold limit	10
PURGE	PURGE_DAYS	Default purge Period.	ORCV_PRICE_INT_PC
PURGE	PURGE_DAYS	Default purge Period.	ORCV_PRICE_INT_PC_ERR_SVC
PURGE	PURGE_DAYS	Default purge Period.	ORCV_PRICE_INT_PC_EVENT
PURGE	PURGE_DAYS	Default purge Period.	ORCV_PRICE_INT_PC_GRP_XREF

Web Service

Frequency: Ad-hoc

Base URL: <BASE_URL>>/PromotionIntRpcJavaServicesWeb/services/orcv/merch/price/integration/promotionintrpcs/batch/<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_PRICE_INT_PC_RPCS_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_ENGINE_THREADS ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG ORCV_PRICE_INT_PC_SVC ORCV_PRICE_INT_PC_ERR_SVC ORCV_PRICE_INT_PC ORCV_PRICE_INT_PC_EVENT ORCV_PRICE_INT_PC_GRP_XREF
/restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "<EXEC_ID>", "jobName": "ORCV_PRICE_INT_PC_RPCS_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_ENGINE_THREADS ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG ORCV_PRICE_INT_PC_SVC ORCV_PRICE_INT_PC_ERR_SVC ORCV_PRICE_INT_PC ORCV_PRICE_INT_PC_EVENT ORCV_PRICE_INT_PC_GRP_XREF
/log	GET	URL query parameters: /log?jobName=ORCV_PRICE_INT_PC_RPCS_JOB& executionId=<EXEC_ID>&date=YYYY-MM-DD	N/A

URL Path	Request Method	Request Body	Affected Tables
/kill	POST	{ "executionId": <EXEC_ID>, "jobName":"ORCV_PRICE_INT_PC_RPCS_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
/status	GET	URL query parameters: /status?jobName=ORCV_PRICE_INT_PC_RPCS_JOB&executionId=<EXEC_ID>	N/A
/ping	GET	Empty	N/A

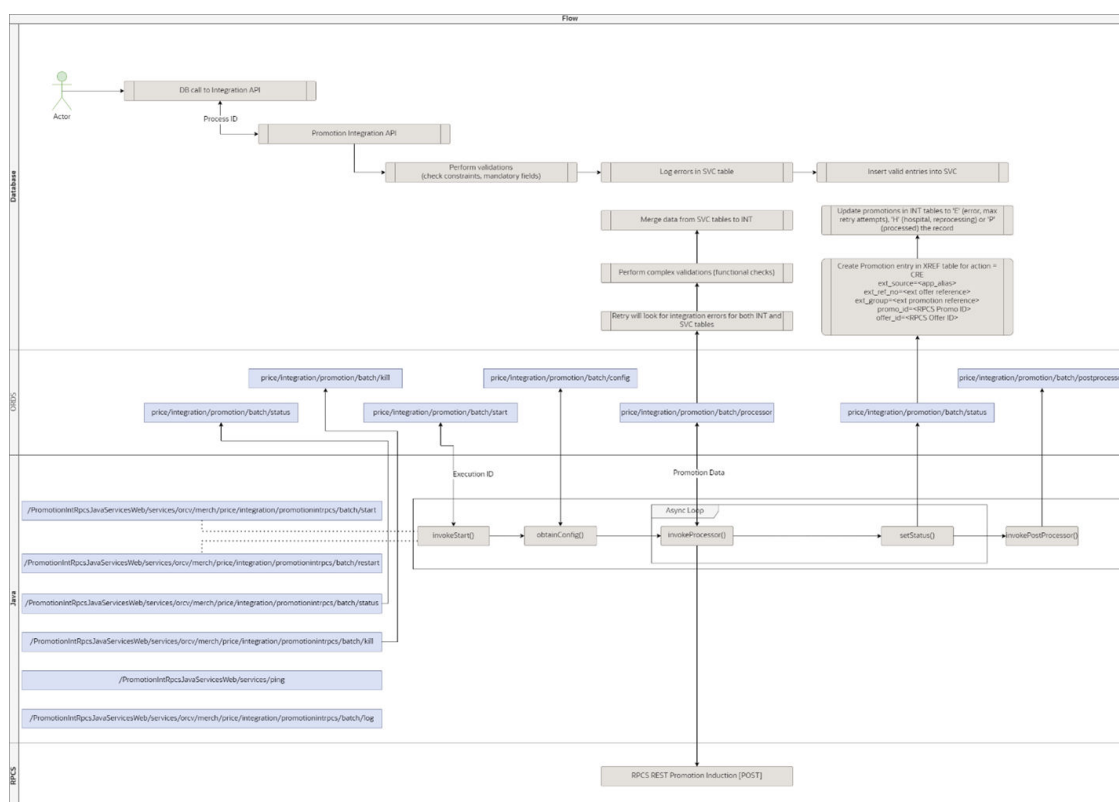
Promotion Induction to RPCS

This section describes the promotion induction to RPCS.

Process Overview

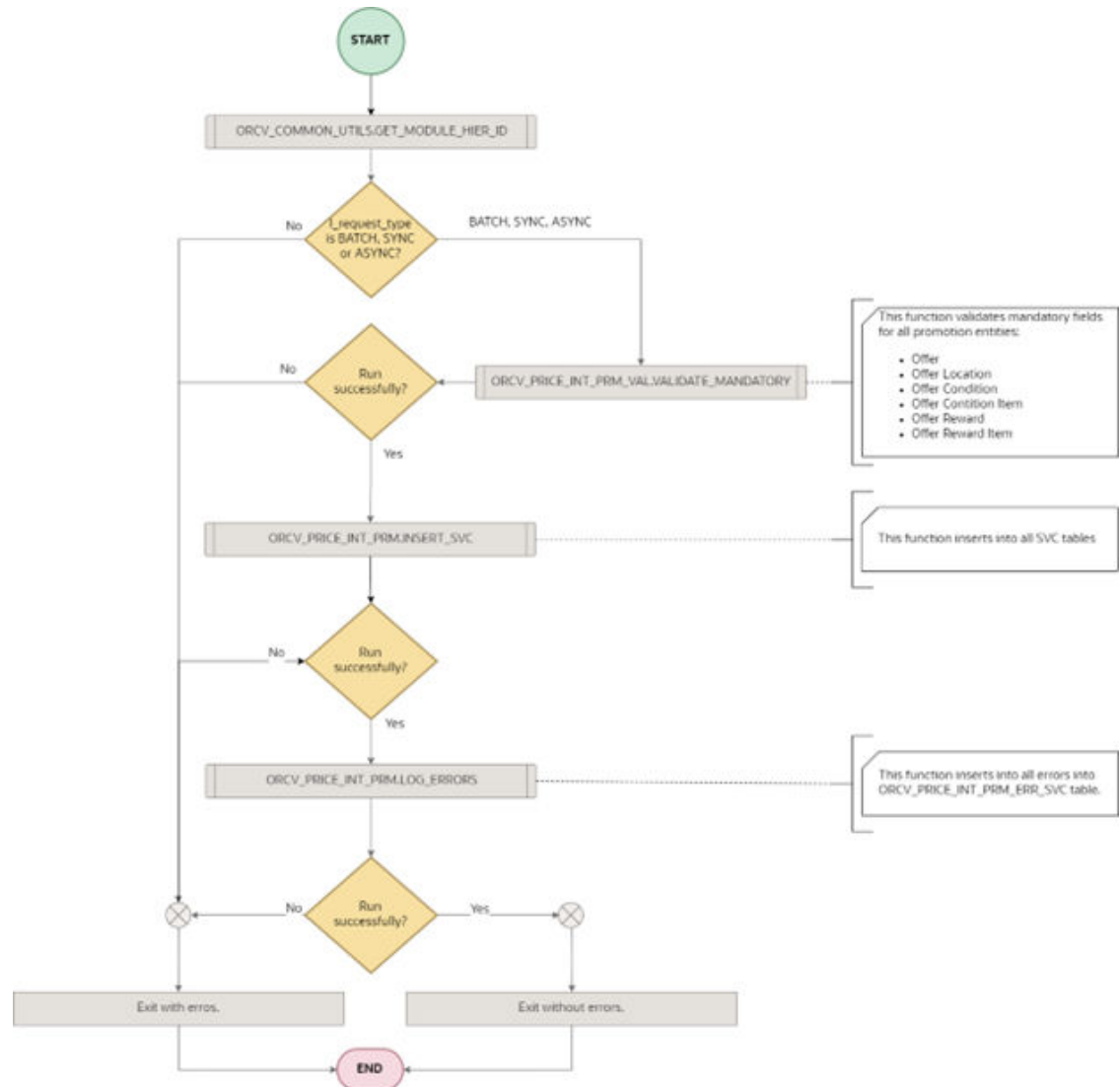
The Promotion induction batch service will support the integration of Promotions into RPCS through the induction REST service.

Figure 27-37 Promotion Induction to RPCS Batch Process Overview



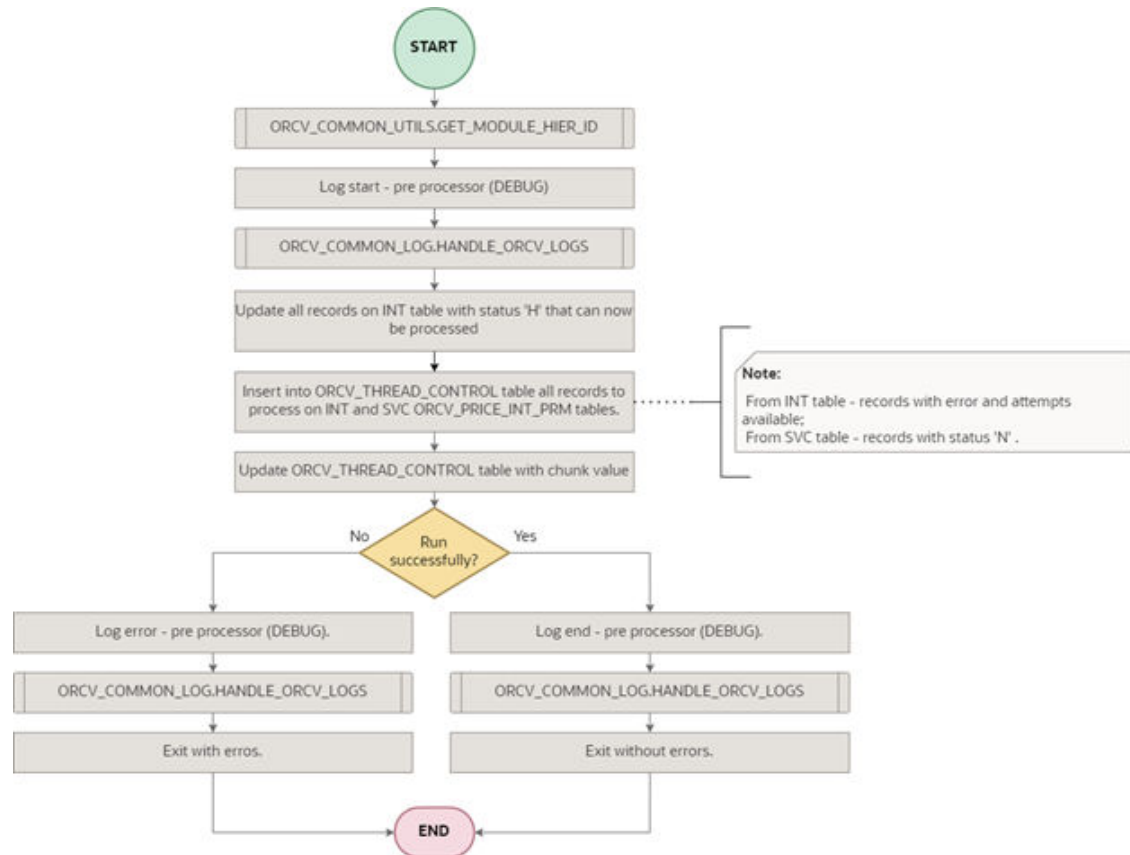
Process Promotion

Figure 27-38 Promotion Process Overview



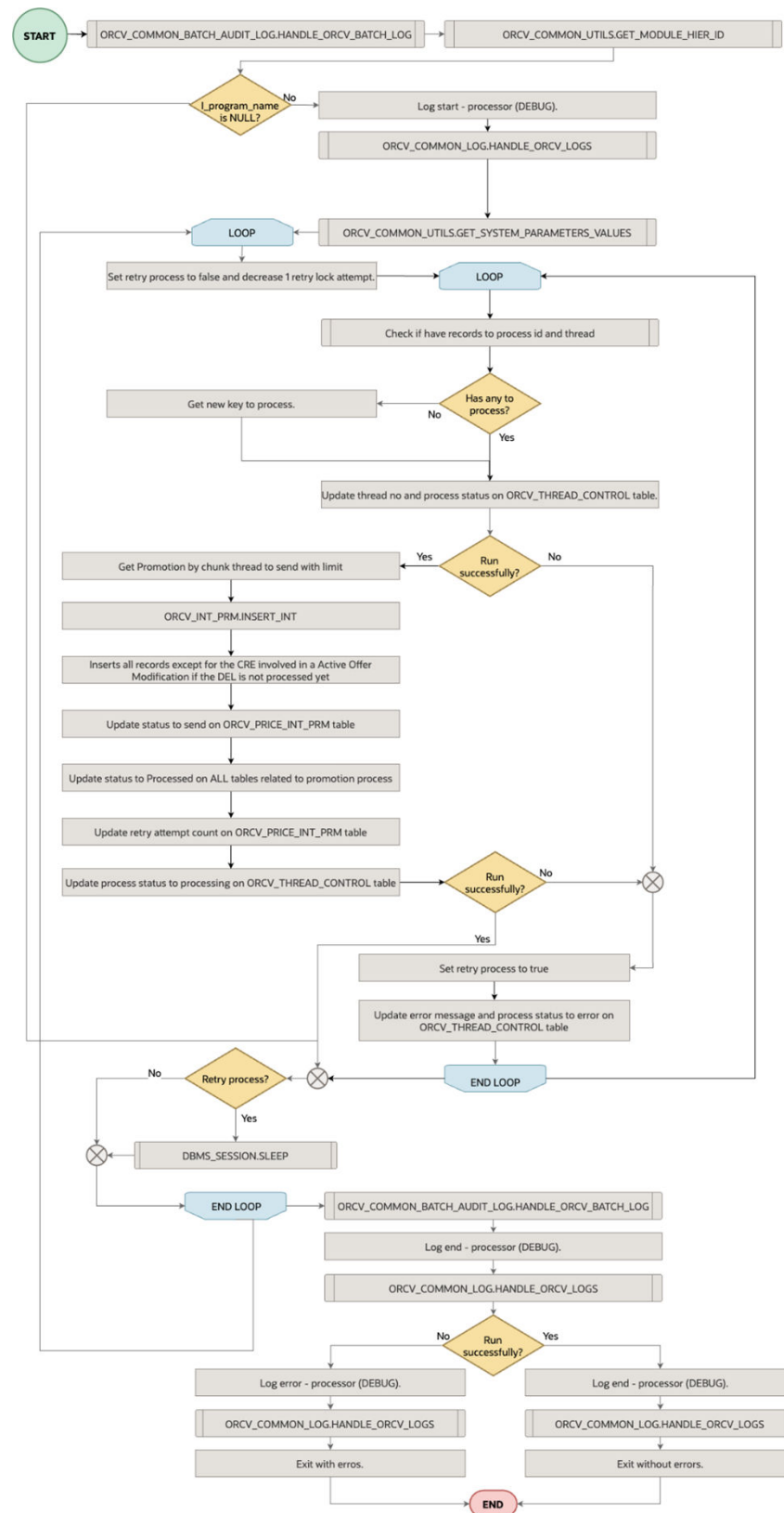
Pre Processor

Figure 27-39 Pre-Processor Promotion to RPCS Integration



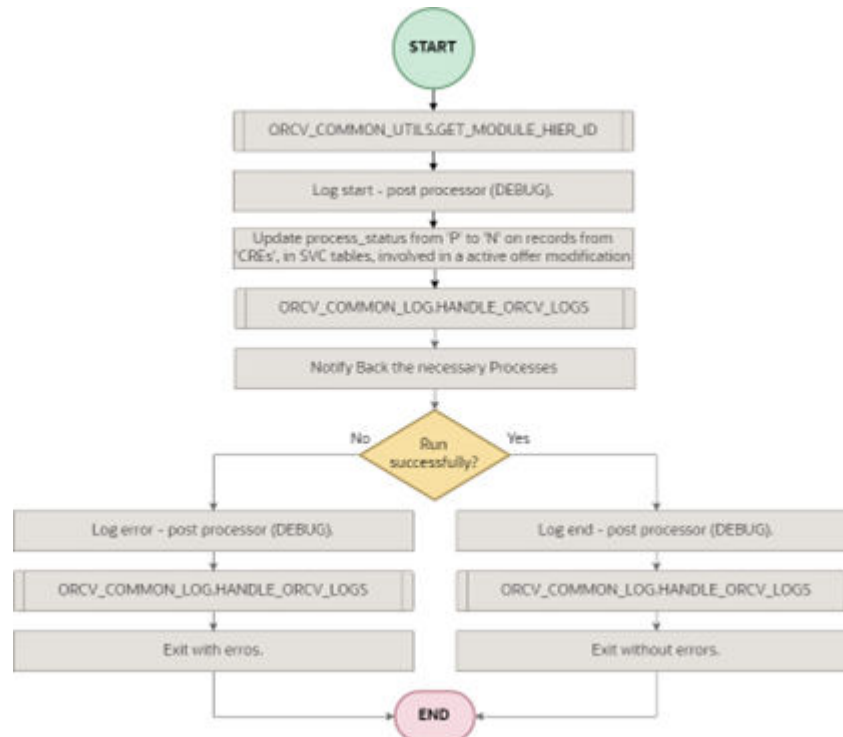
Processor

Figure 27-40 Processor Promotion induction to RPCS Integration



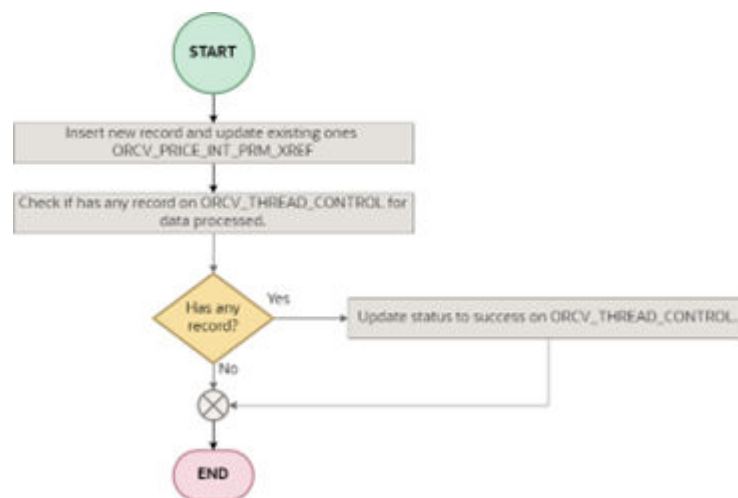
Post Processor

Figure 27-41 Post Processor Promotion induction to RPCS Integration



Set Integration Status

Figure 27-42 Set Integration Status Flow for Promotion induction to RPCS Integration



Configuration

This section describes configurations.

Initial Batch Configuration for ORCV_PRICE_INT_PRM_RPCS_JOB

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_PRICE_INT_PRM_RPCS_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
BULK_LIMIT	1000
RETRY_LOCK_ATTEMPTS	4
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

System Parameters

Listed below is the list of configurations at system level that are required by this integration.

The parameters are defined in table ORCV_SYSTEM_PARAMETERS.

Area	Parameter	Description	Value 1	Value 2
CONFIG	INDUCTION_TEMPLATE	Template key.	RPCS_PROMOTION - INDUCTION	NA
INTEGRATION	LOCK_PROMOTION_PROCESS	Promotion Ext Source that will not be processed by the batch.	SPDM	NA
PURGE	PURGE_DAYS	Default purge Period.	ORCV_PRICE_INT_PRM_XREF	30

Web Service

Frequency: Ad-hoc

Base URL: <BASE_URL>/PromotionIntRpcJavaServicesWeb/services/orcv/merch/price/integration/promotionintrpcs/batch/<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_PRICE_INT_PRM_RPCS_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_ENGINE_THREADS ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG ORCV_PRICE_INT_PRM_HEAD_SVC ORCV_PRICE_INT_PRM_HEAD ORCV_PRICE_INT_PRM_OF_SVC ORCV_PRICE_INT_PRM_OF ORCV_PRICE_INT_PRM_OF_SVC ORCV_PRICE_INT_PRM_OF ORCV_PRICE_INT_PRM_OF_ITM_SVC ORCV_PRICE_INT_PRM_OF_ITM ORCV_PRICE_INT_PRM_OF_SVC ORCV_PRICE_INT_PRM_OF ORCV_PRICE_INT_PRM_OF_ITM_SVC ORCV_PRICE_INT_PRM_OF_ITM ORCV_PRICE_INT_PRM_OFLOC_SVC ORCV_PRICE_INT_PRM_OFLOC ORCV_PRICE_INT_PRM_ERR_SVC ORCV_PRICE_INT_PRM_XREF
/restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_PRICE_INT_PRM_RPCS_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_ENGINE_THREADS ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG ORCV_PRICE_INT_PRM_HEAD_SVC ORCV_PRICE_INT_PRM_HEAD ORCV_PRICE_INT_PRM_OF_SVC ORCV_PRICE_INT_PRM_OF ORCV_PRICE_INT_PRM_OF_SVC ORCV_PRICE_INT_PRM_OF ORCV_PRICE_INT_PRM_OF_ITM_SVC ORCV_PRICE_INT_PRM_OF_ITM ORCV_PRICE_INT_PRM_OF_SVC ORCV_PRICE_INT_PRM_OF ORCV_PRICE_INT_PRM_OF_ITM_SVC ORCV_PRICE_INT_PRM_OF_ITM ORCV_PRICE_INT_PRM_OFLOC_SVC ORCV_PRICE_INT_PRM_OFLOC ORCV_PRICE_INT_PRM_ERR_SVC ORCV_PRICE_INT_PRM_XREF

URL Path	Request Method	Request Body	Affected Tables
/log	GET	URL query parameters: /log?jobName=ORCV_PRICE_INT_PRM_RPCS_JOB& executionId=<EXEC_ID>&date=YYYY-MM-DD	N/A
/kill	POST	{ "executionId": <EXEC_ID>, "jobName": "ORCV_PRICE_INT_PRM_RPCS_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
/status	GET	URL query parameters: /status?jobName=ORCV_PRICE_INT_PRM_RPCS_JOB&executionId=<EXEC_ID>	N/A
/ping	GET	Empty	N/A

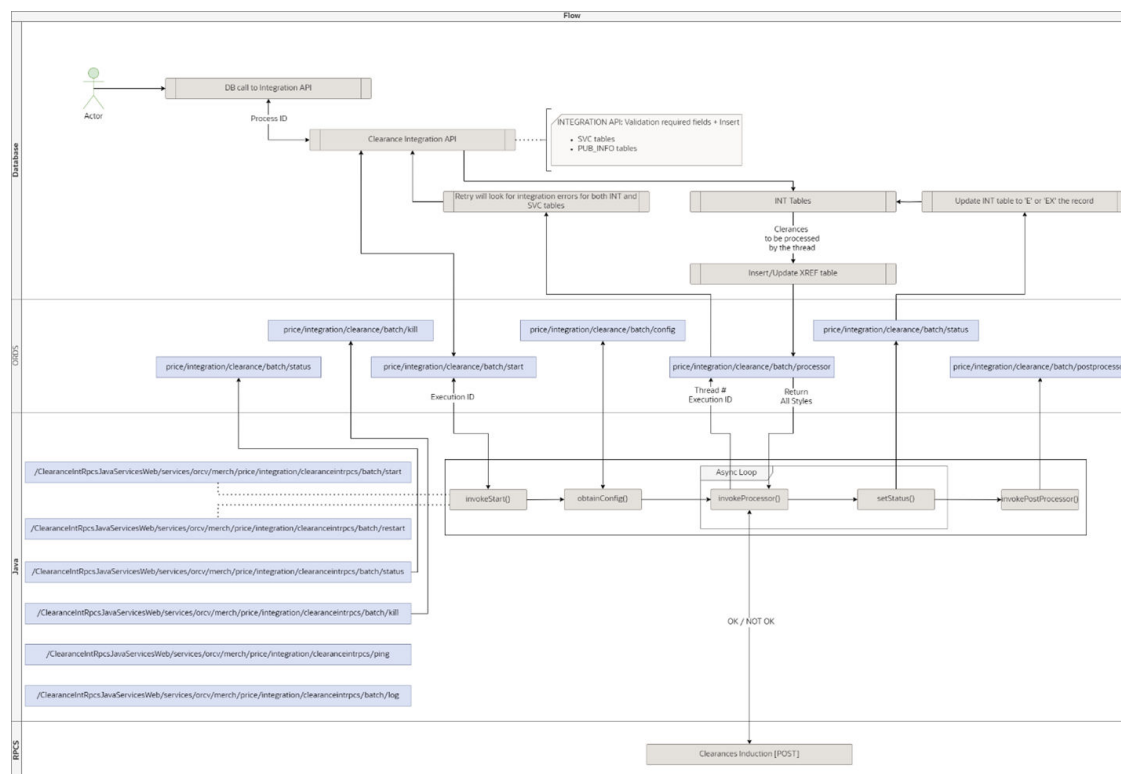
Clearance Induction to RPCS

This section describes the clearance induction to RPCS.

Process Overview

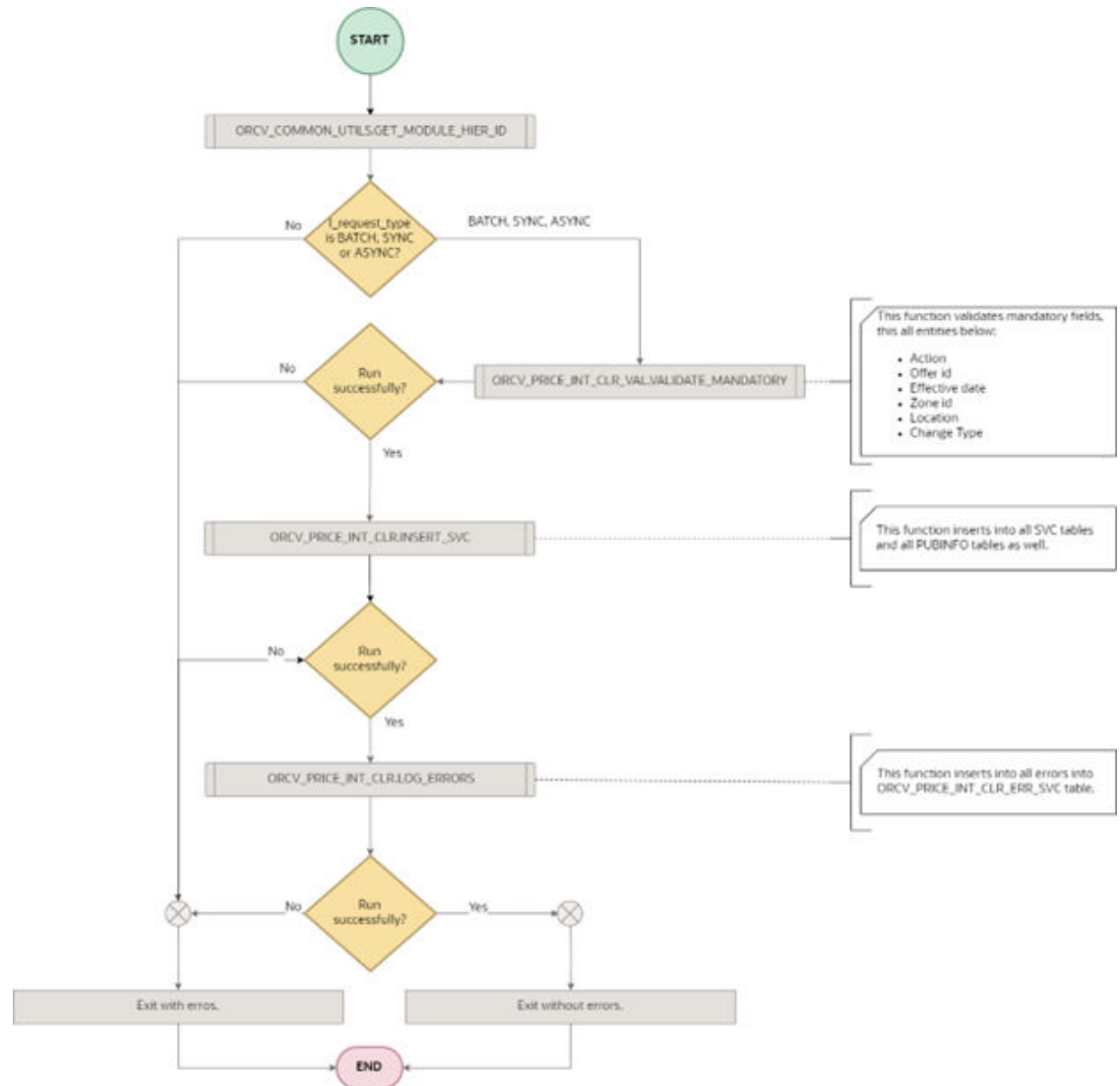
The Clearance induction batch service will support the integration of Clearances into RPCS through the induction REST service.

Figure 27-43 Clearance Induction to RPCS Batch Process Overview



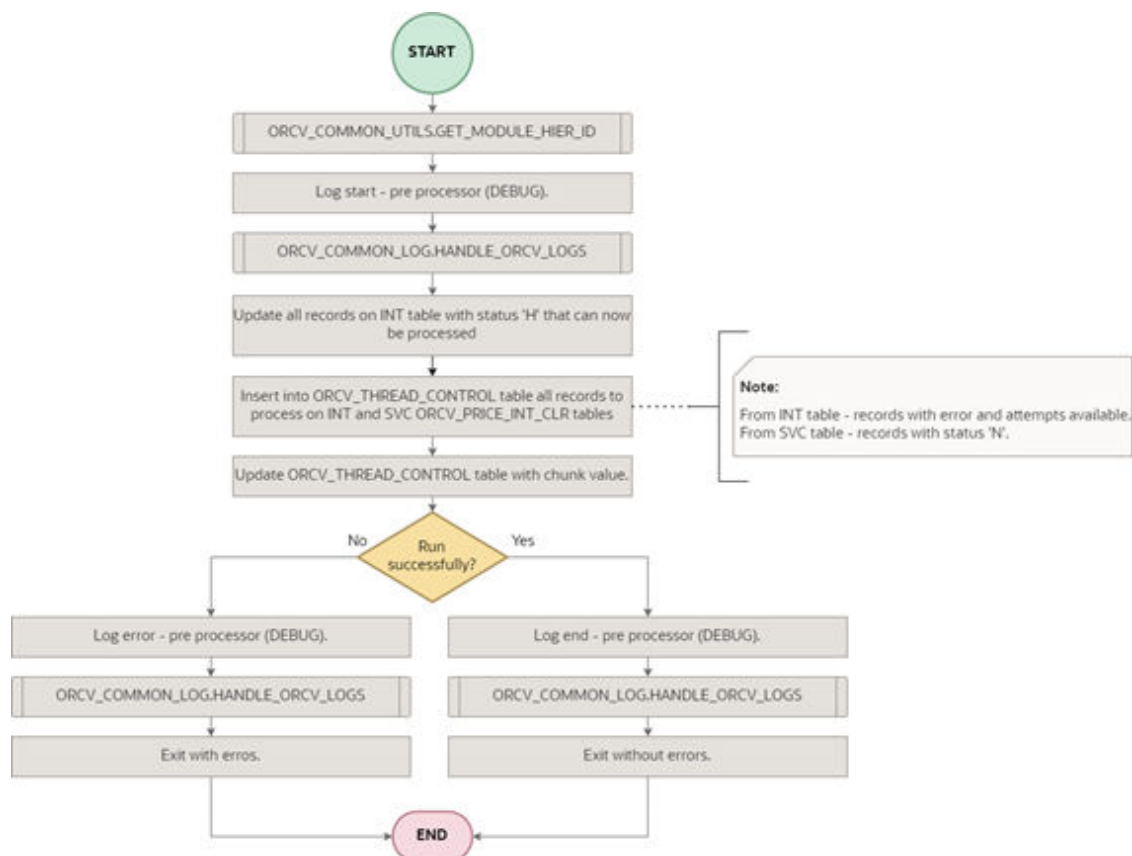
Process Clearance

Figure 27-44 Clearance Process Overview



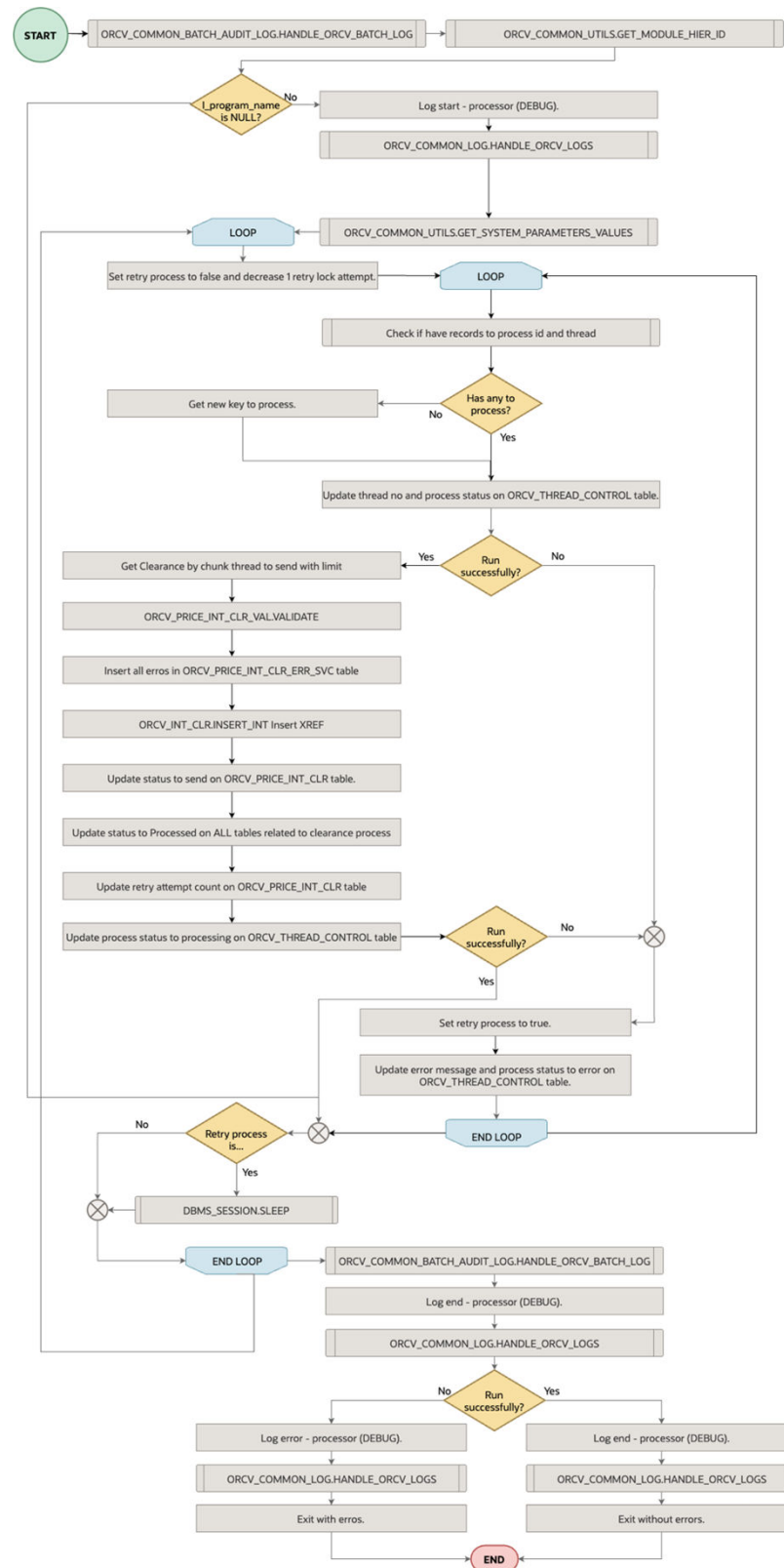
Pre Processor

Figure 27-45 Pre Processor Clearance to RPCS Integration



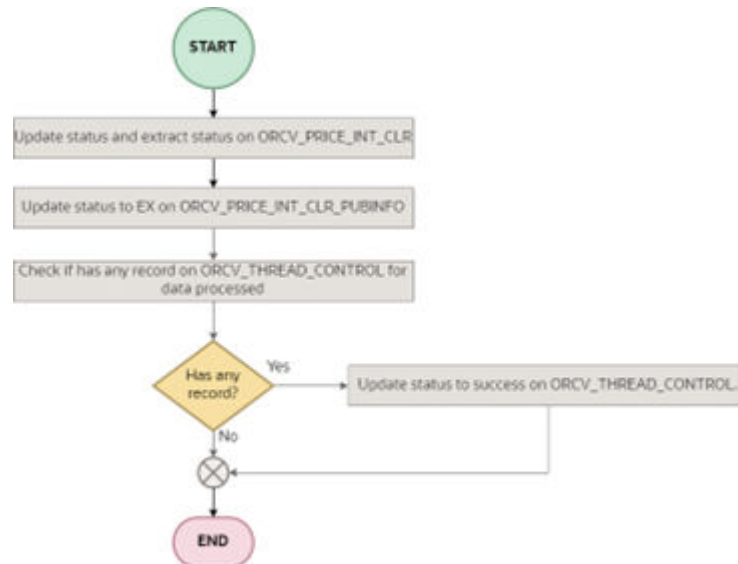
Processor

Figure 27-46 Processor Clearance induction to RPCS Integration



Set Integration Status

Figure 27-47 Set Integration Status Flow for Clearance induction to RPCS Integration



Configuration

This section describes configurations.

Initial Batch Configuration for ORCV_PRICE_INT_CLR_RPCS_JOB

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_PRICE_INT_CLR_RPCS_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
BULK_LIMIT	1000
RETRY_LOCK_ATTEMPTS	4
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

System Parameters

Listed below is the list of configurations at system level that are required by this integration.

The parameters are defined in table ORCV_SYSTEM_PARAMETERS.

Area	Parameter	Description	Value 1	Value 2
CONFIG	CLEAR_MARKDOWN – NBR_DEFAULT	Markdown NBR for clearance. Default set with -999 (means NULL).Valid values: MFCS Code Type MKDN(except "reset")	-999	NA
PURGE	PURGE_DAYS	Default purge Period.	ORCV_PRICE_INT_CLEARANCE	30
PURGE	PURGE_DAYS	Default purge Period.	ORCV_PRICE_INT_CLR_XREF	30

Web Service

Clearance Induction Batch Service

Frequency: Ad hoc

Base URL: <BASE_URL>/ClearanceIntRpcJavaServicesWeb/services/orcv/merch/price/integration/

clearanceintrpcs/URL_PATH>

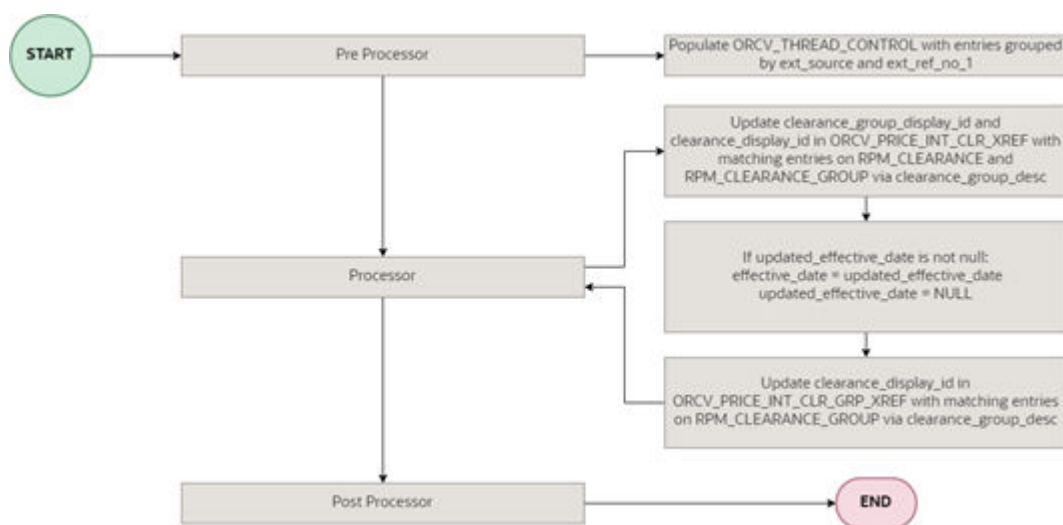
URL Path	Request Method	Request Body	Affected Tables
batch/ start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_PRICE_INT_CLR_RPCS_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_ENGINE_THREADS ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG ORCV_PRICE_INT_CLEARANCE ORCV_PRICE_INT_CLR_GRP_XREF ORCV_PRICE_INT_CLEARANCE_SVC ORCV_PRICE_INT_CLR_XREF ORCV_PRICE_INT_CLEARANCE_ERROR
batch/ restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_PRICE_INT_CLR_RPCS_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_ENGINE_THREADS ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG ORCV_PRICE_INT_CLEARANCE ORCV_PRICE_INT_CLR_GRP_XREF ORCV_PRICE_INT_CLEARANCE_SVC ORCV_PRICE_INT_CLR_XREF ORCV_PRICE_INT_CLEARANCE_ERROR

URL Path	Request Method	Request Body	Affected Tables
batch/log	GET	URL query parameters: /log? jobName=ORCV_PRICE_INT_CLR_RPC S_JOB&executionId=<EXEC_ID>&date =YYYY-MM-DD	N/A
batch/kill	POST	{ "executionId": <EXEC_ID>, "jobName": "ORCV_PRICE_INT_CLR_ RPCS_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
batch/status	GET	URL query parameters: /status? jobName=ORCV_PRICE_INT_CLR_RPC S_JOB&executionId=<EXEC_ID>	N/A
/ping	GET	Empty	N/A

Clearance XREF Batch Service

The Clearance XREF batch service allow synchronization of RPCS clearance group display id and clearance display id to the external reference integrated via Clearance Induction to RPCS batch service.

Figure 27-48 Clearance XREF Batch Service Overview



Initial Batch Configuration for ORCV_PRICE_INT_CLR_XREF_JOB

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_PRICE_INT_CLR_XREF_JOB

Column Name	Default Values
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	4
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Web Service

Frequency: Ad hoc

Base URL: <BASE_URL>/ords/mfcs/orcv/merch/price/integration/clearance/xrefupdate/
<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
batch/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_PRICE_INT_CLR_XREF_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_ENGINE_THREADS ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG ORCV_PRICE_INT_CLR_GRP_XREF ORCV_PRICE_INT_CLR_XREF
batch/restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "<EXEC_ID>", "jobName": "ORCV_PRICE_INT_CLR_XREF_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_ENGINE_THREADS ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG ORCV_PRICE_INT_CLR_GRP_XREF ORCV_PRICE_INT_CLR_XREF
batch/log	GET	URL query parameters: /log? jobName=ORCV_PRICE_INT_CLR_XREF_JOB& executionId=<EXEC_ID>&logDate=YY YY-MM-DD	N/A

URL Path	Request Method	Request Body	Affected Tables
batch/kill	POST	{ "executionId": <EXEC_ID>, "jobName": "ORCV_PRICE_INT_CLR_XREF_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
batch/status	GET	URL query parameters: /status?jobName=ORCV_PRICE_INT_CLR_XREF_JOB&executionId=<EXEC_ID>	N/A
/ping	GET	Empty	N/A

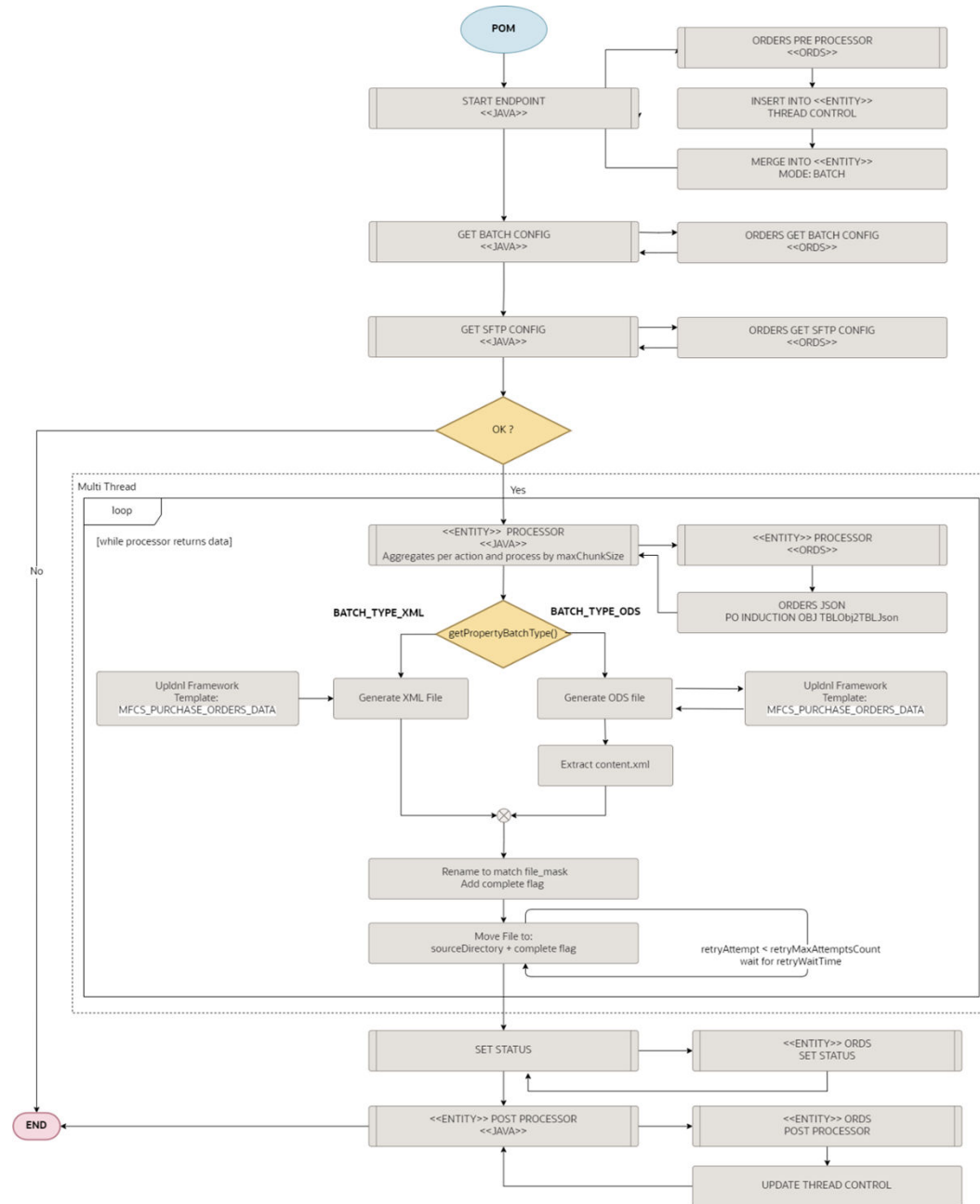
Purchase Order Induction to MFCS

The Purchase Order induction to MFCS batch has a maximum chunk size configuration and has a thread aggregator that is a combination of master purchase order and purchase order. The chunk size determines the maximum number of master purchase order and purchase order that will be sent to MFCS per purchase order induction ODS file, and then consumed by MFCS POINDBATCH_JOB.

Process Overview

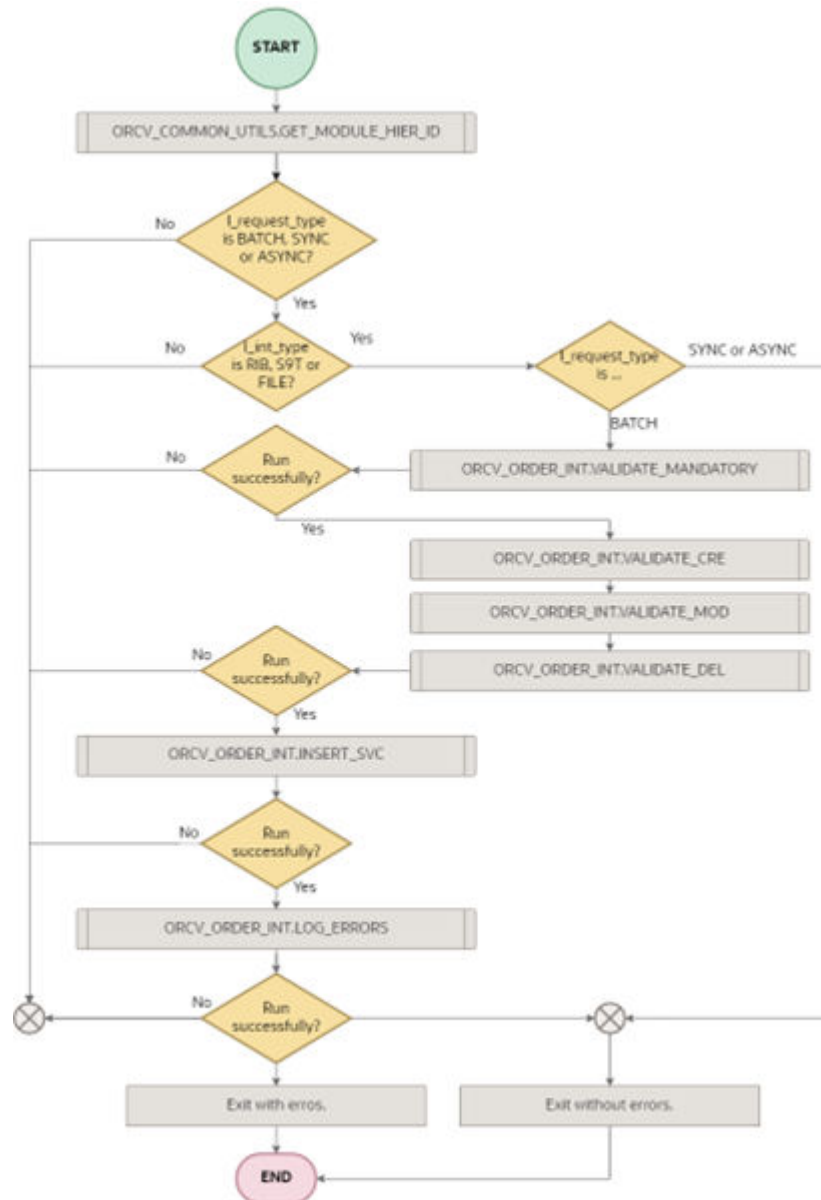
The Purchase Order induction service will support the integration of Purchase Orders into MFCS through induction batch.

Figure 27-49 Purchase Order Induction to MFCS Process Overview



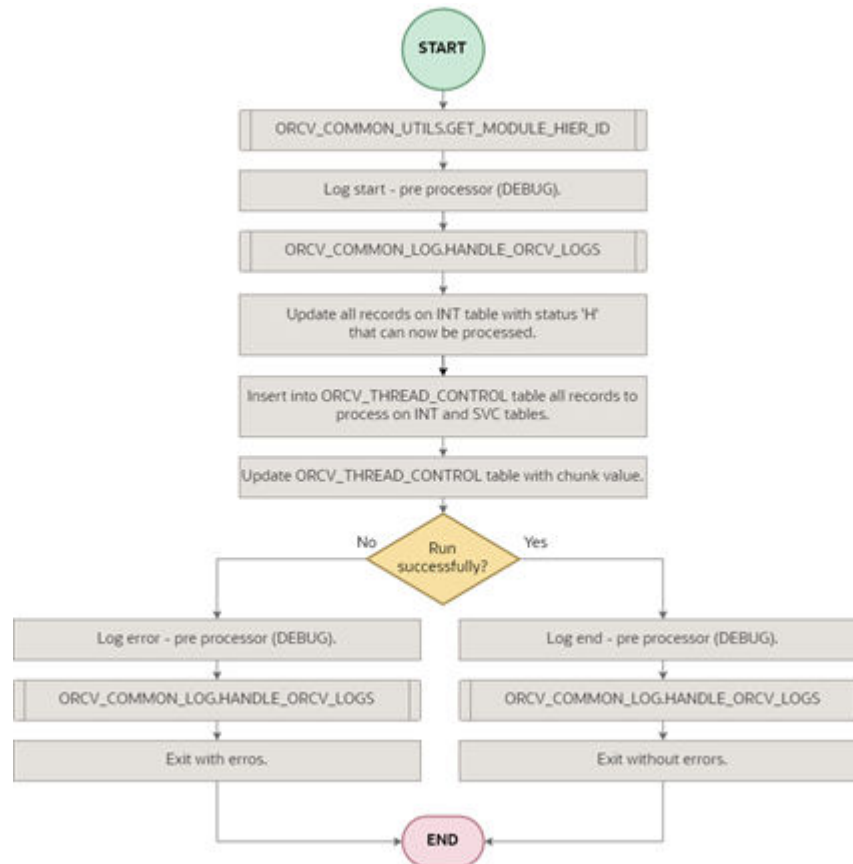
Process Purchase Order

Figure 27-50 Process Purchase Order Function



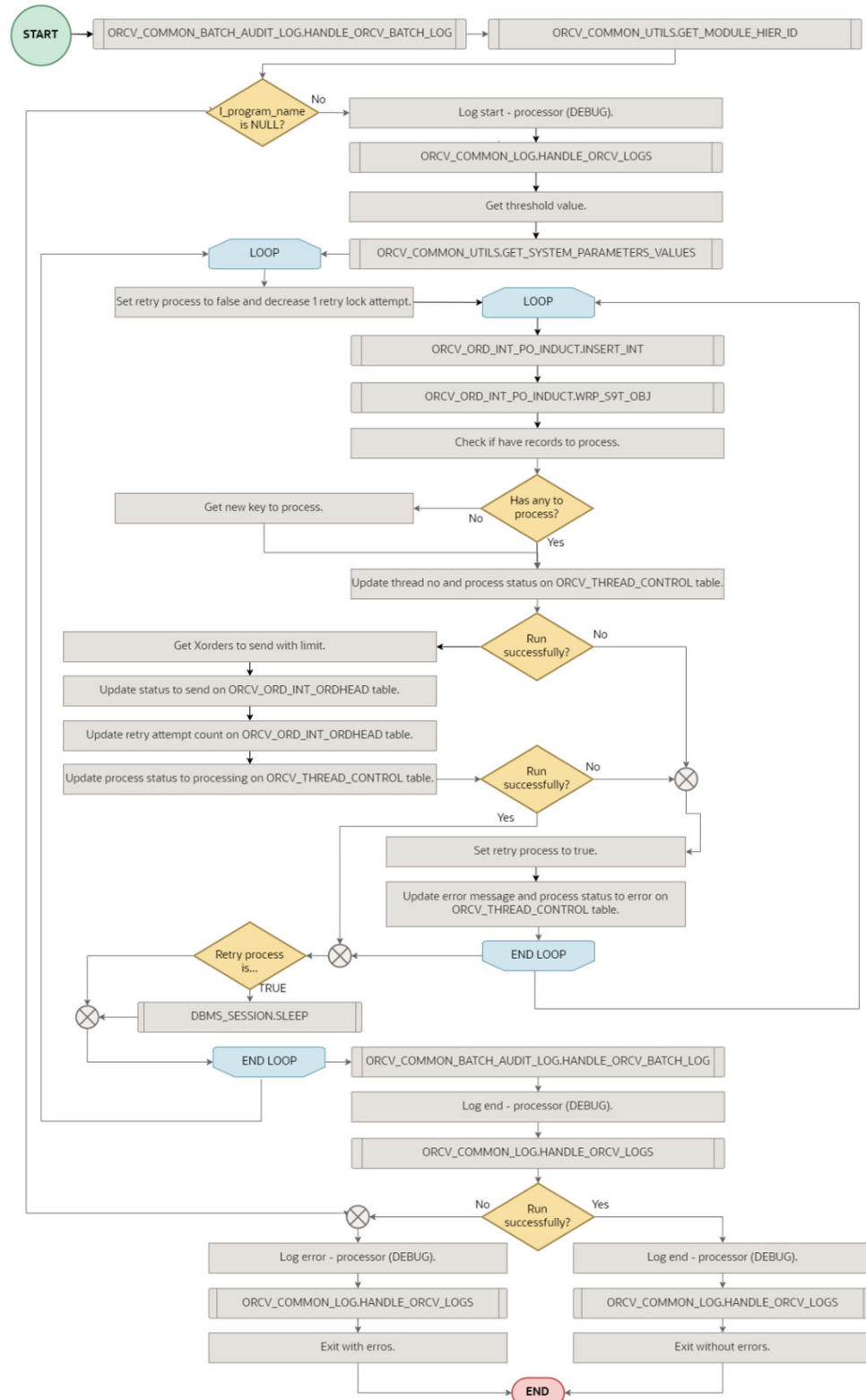
Pre Processor

Figure 27-51 Pre-Processor Purchase Order Function



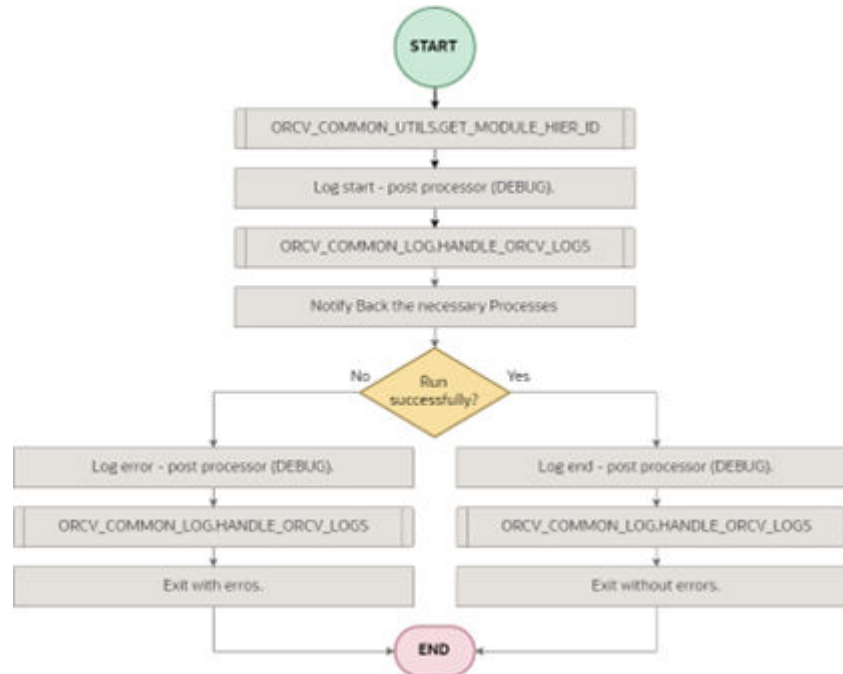
Processor

Figure 27-52 Processor Purchase Order Function



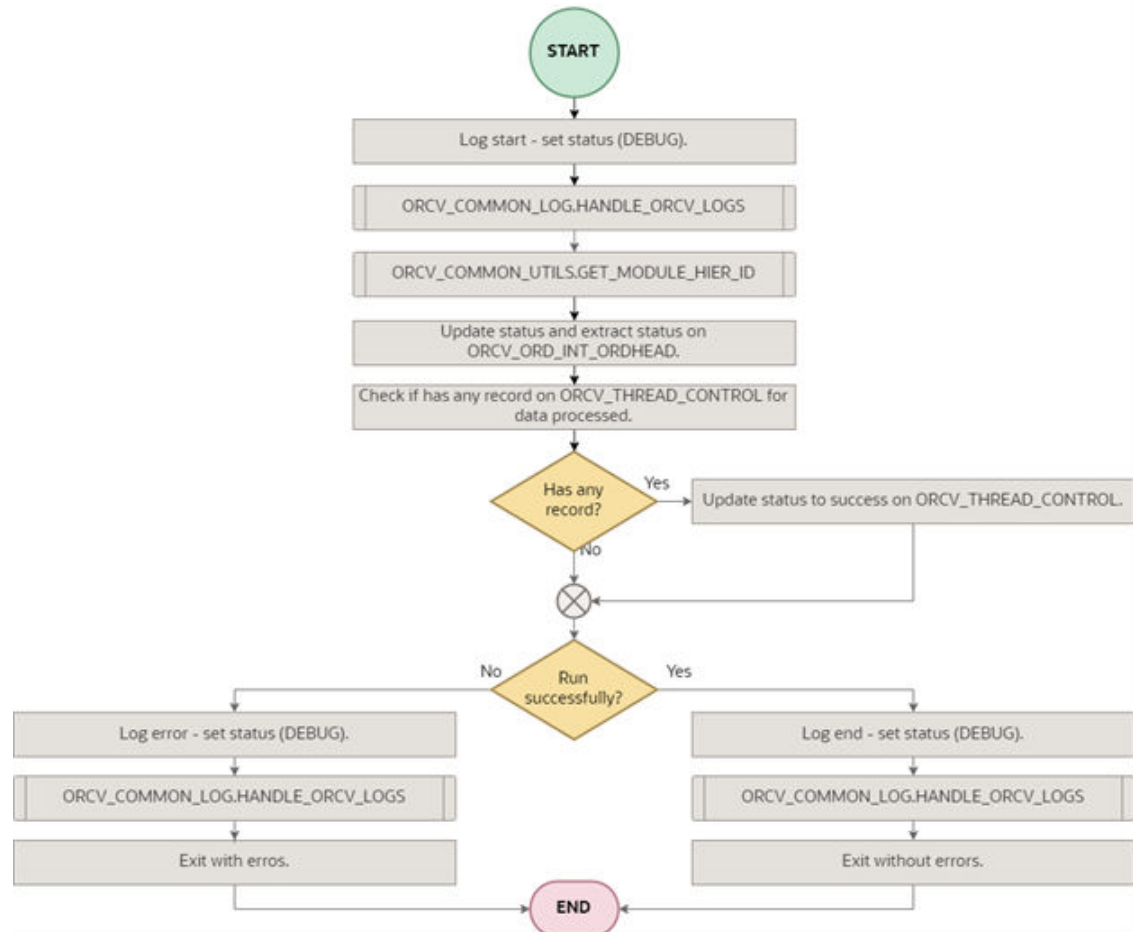
Post Processor

Figure 27-53 Post Processor Purchase Order Function



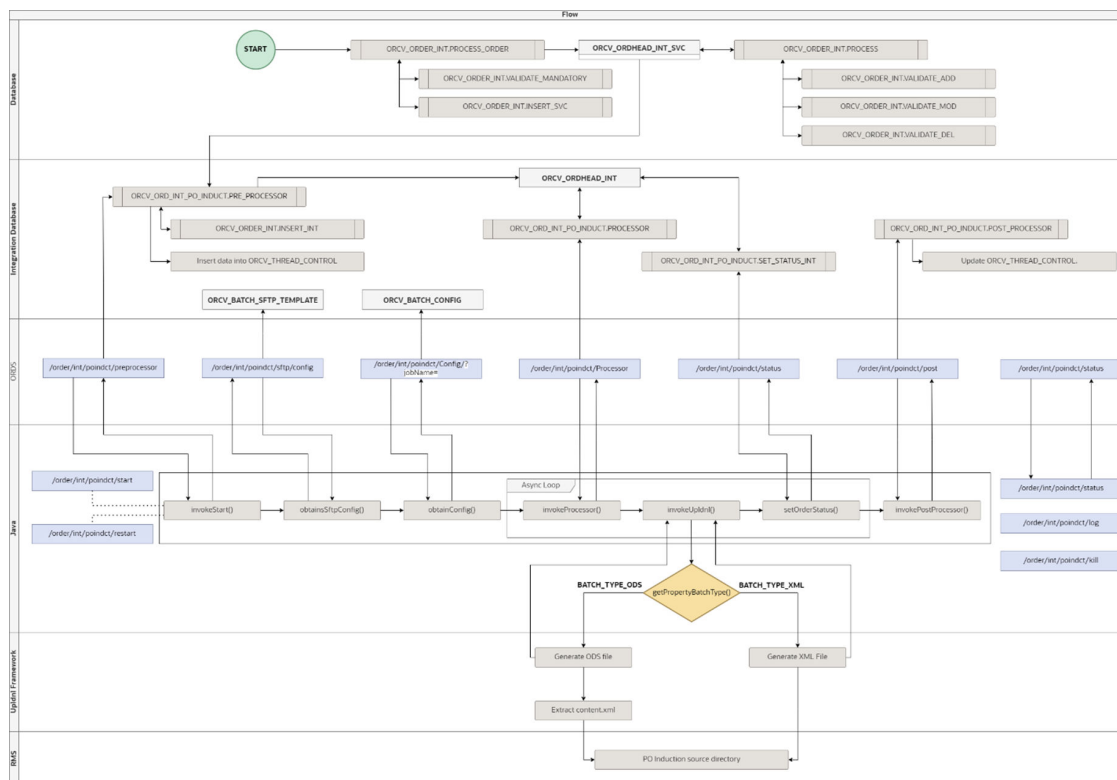
Set Integration Status

Figure 27-54 Set Integration Status Purchase Order Function



Batch Flow

Figure 27-55 Purchase Order integration Batch Flow



Configuration

The Purchase Order integration service has a prerequisite of the interface with ID POINDUCTION being configured in ORCV_BATCH_SFTP and ORCV_BATCH_SFTP_TARGET tables.



Note:

POM configuration for MFCS POINDBATCH_JOB set to run at a high frequency if required and with a final target the merchandising tables.

Initial Batch Configuration for ORCV_ORD_INT_POINDUCTION_JOB

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_ORD_INT_POINDUCTION_JOB
MAX_CHUNK_SIZE	10000
MAX_CONCURRENT_THREADS	4

Column Name	Default Values
BULK_LIMIT	1000
RETRY_LOCK_ATTEMPTS	4
RETRY_WAIT_TIME	2
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Web Service

Frequency: Ad hoc

Authorization Method: OAuth 2.0

Base URL: <BASE_URL>/PoInductionJavaServicesWeb/services/orcv/merch/integration/
<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
batch/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_ORD_INT_ POINDUCTION_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_ORD_INT_ORDHEAD_SVC ORCV_ORD_INT_ORDSKU_SVC ORCV_ORD_INT_ORDLOC_SVC ORCV_ORD_INT_ORDER_CFA_SV C ORCV_ORD_INT_ERR_SVC ORCV_ORD_INT_ORDHEAD ORCV_ORD_INT_ORDSKU ORCV_ORD_INT_ORDLOC ORCV_ORD_INT_ORDHEAD_CFA ORCV_ORD_INT_ORDSKU_CFA ORCV_ORD_INT_ORDLOC_CFA

URL Path	Request Method	Request Body	Affected Tables
batch/restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_ORD_INT_POINDUCTION_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_ORD_INT_ORDHEAD_SVC ORCV_ORD_INT_ORDSKU_SVC ORCV_ORD_INT_ORDLOC_SVC ORCV_ORD_INT_ORDER_CFA_SVC ORCV_ORD_INT_ERR_SVC ORCV_ORD_INT_ORDHEAD ORCV_ORD_INT_ORDSKU ORCV_ORD_INT_ORDLOC ORCV_ORD_INT_ORDHEAD_CFA ORCV_ORD_INT_ORDSKU_CFA ORCV_ORD_INT_ORDLOC_CFA
batch/log	GET	URL query parameters: /log?jobName=ORCV_ORD_INT_POINDUCTION_JOB &executionId=<EXEC_ID>&date=YYYY-MM-DD	N/A
batch/kill	POST	{ "executionId": <EXEC_ID>, "jobName": "ORCV_ORD_INT_POINDUCTION_JOB", }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
batch/status	GET	URL query parameters: /status? jobName=ORCV_ORD_INT_POINDUCTION_JOB &executionId=<EXEC_ID>	N/A
/ping	GET	Empty	N/A

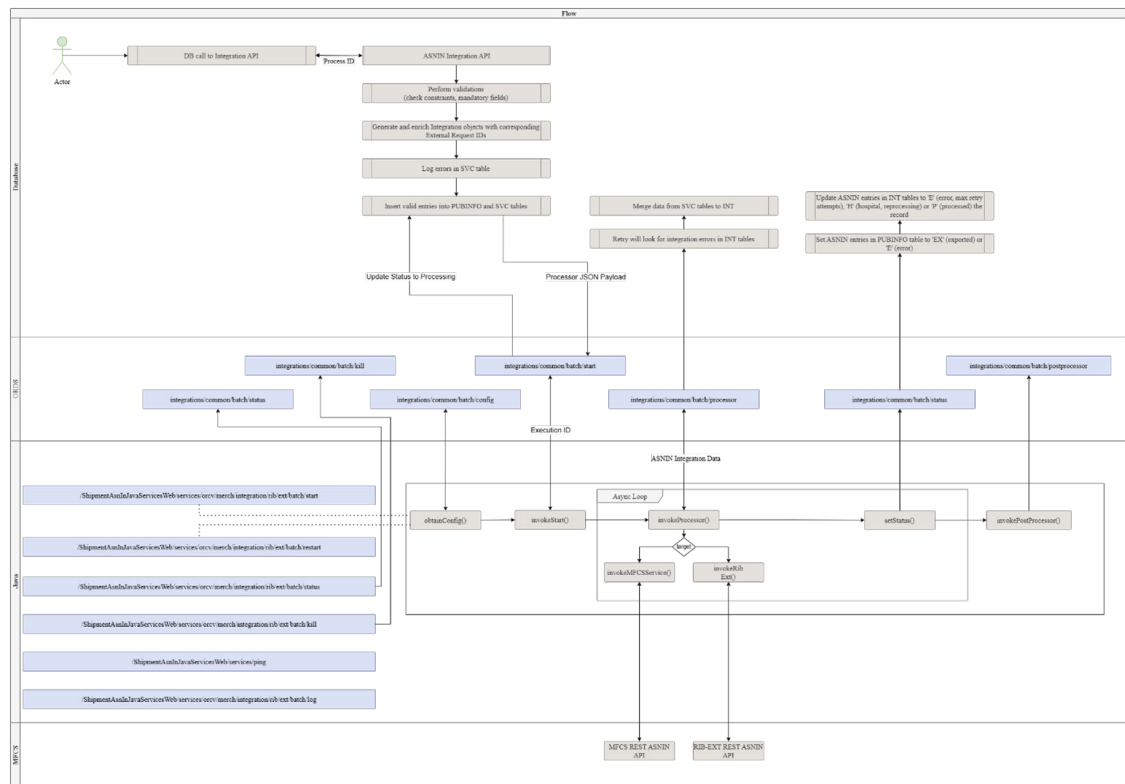
Shipment ASNIN Integration to RICS

The Shipments ASNIN integration service supports the integration of ASNINs into MFCS through MFCS MerchIntegrations Rest Service and RIB-EXT Rest Service.

This integration has a maximum chunk size configuration and has a thread aggregator per number of ASN numbers. The chunk size determines the maximum number of ASNIN records that will be sent to MFCS.

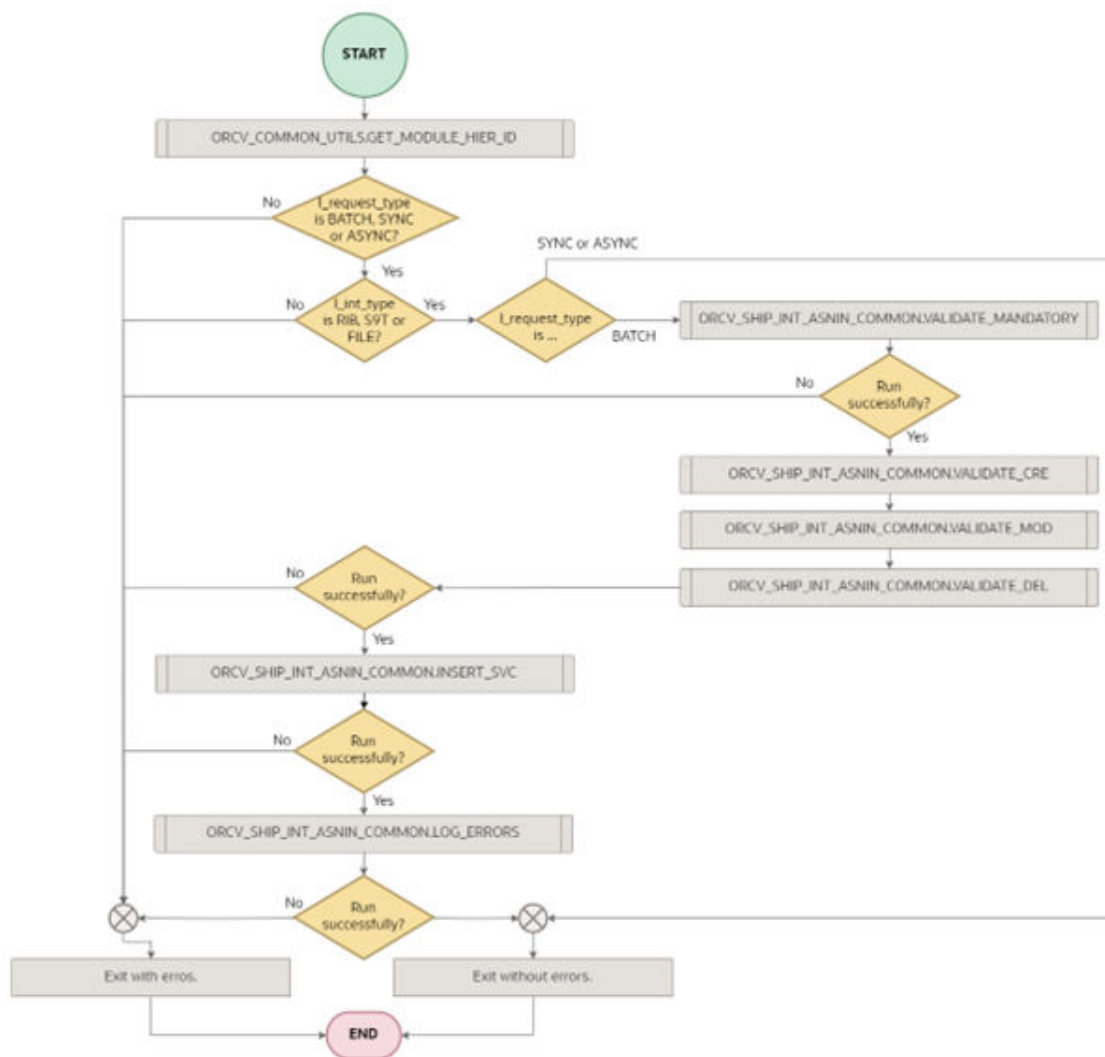
Process Overview

Figure 27-56 Shipment ASNin Integration to MFCS via MerchIntegrations and RIB-EXT Process Overview



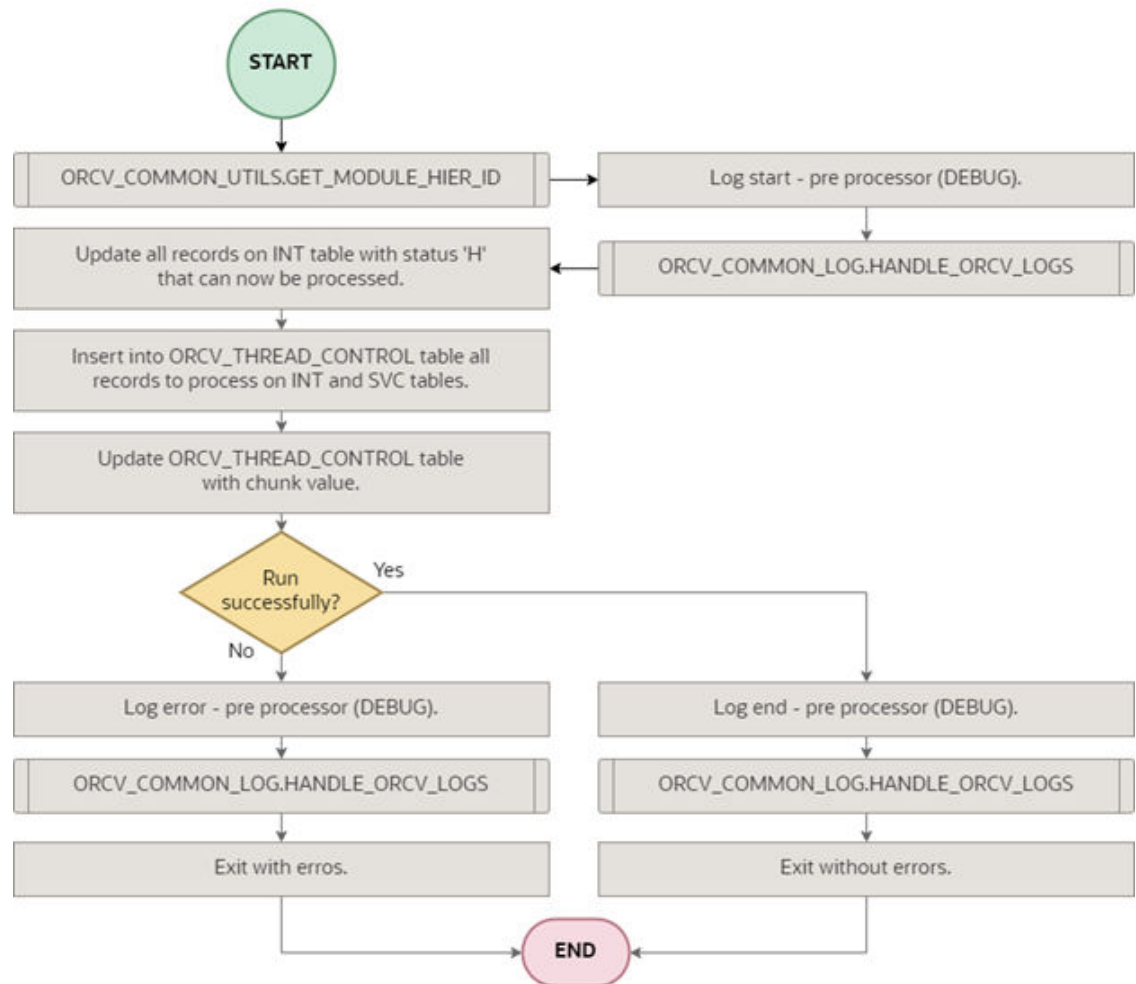
Process Shipment ASNIN

Figure 27-57 Process Shipment ASNIN



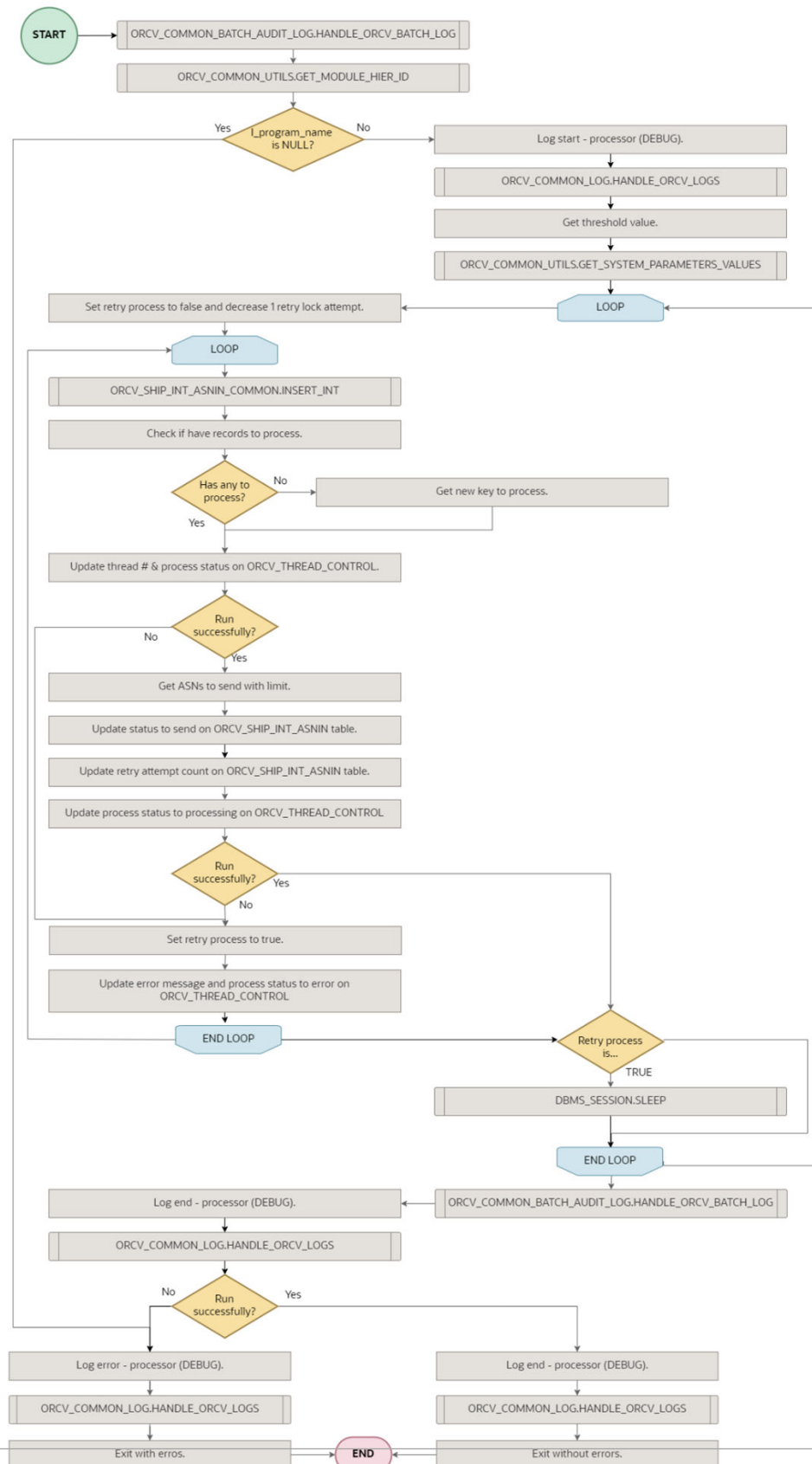
Pre Processor

Figure 27-58 Pre-Processor Shipment ASNIn function



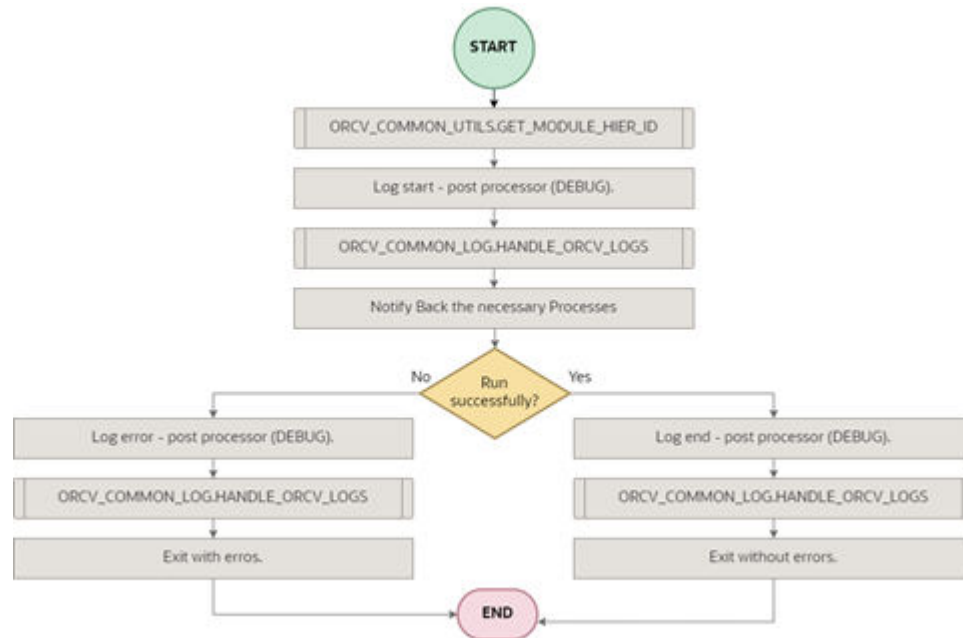
Processor

Figure 27-59 Processor Shipment ASNIn Function



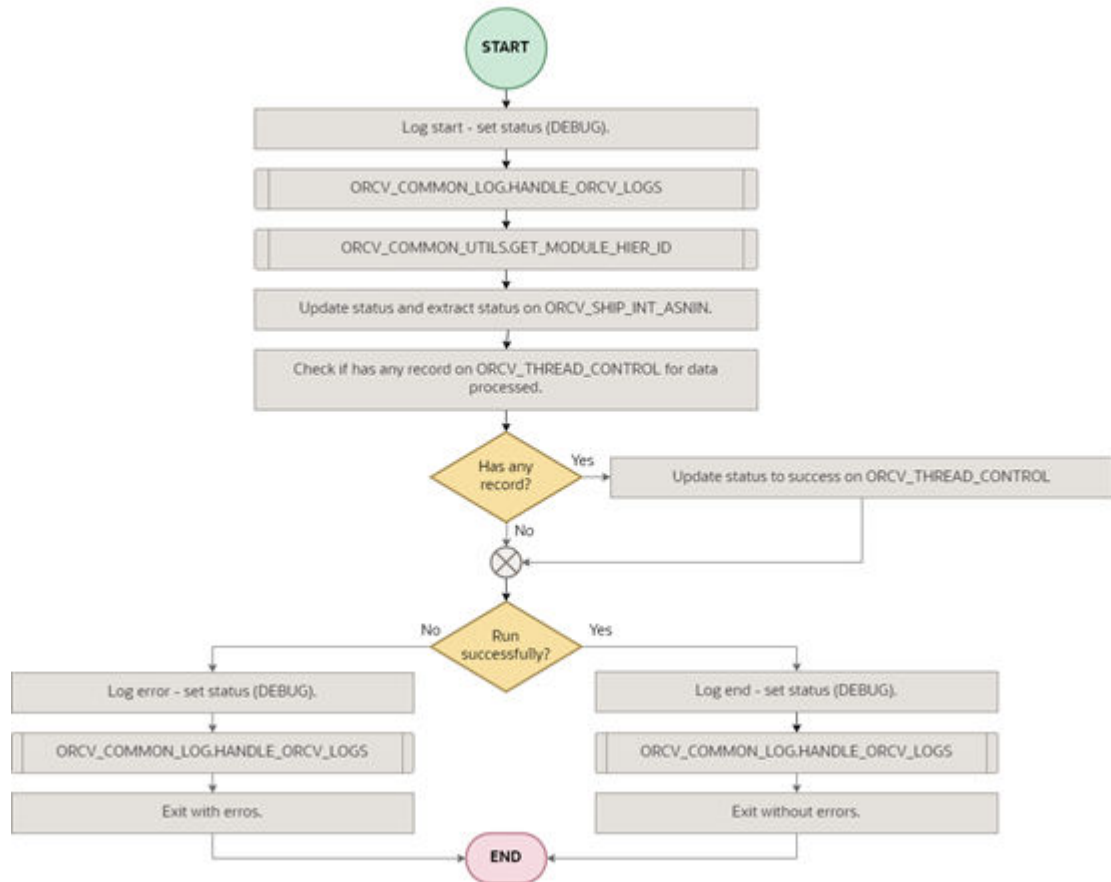
Post Processor

Figure 27-60 Post Processor Shipment ASNIn Function



Set Integration Status

Figure 27-61 Set Integration Status for Shipments ASNIn



ASNIN Targets Configuration

It is possible to configure the targets for ASNINs in table `ORCV_BATCH_CONFIG_PROPS` if the Java Service is configured with `DATABASE_PROPS_ENABLED` set to true and the target id for MFCS is defined as `MFCS_TARGET` key and RICS as `RIB_TARGET` key. After setting the Java Service to use the database configurations the details must be added to `ORCV_BATCH_CONFIG_PROPS` with program name `ORCV_SHIP_INT_ASNIN_JOB`. Examples are shared below for MFCS MechIntegrations Rest Service and RICS RIB -EXT Rest Service.

Java Service Properties Configuration

Parameter	Description	Example
DATABASE_PROPS_ENABLED	If False, target is RIB-EXT and properties are defined within the JAVA application properties file. If True, properties configuration rely on what is defined on ORCV_BATCH_CONFIG_PROPS, accepts multiple targets at the same time.	true
RIB_TARGET	Needs to match the KEY defined on ORCV_BATCH_CONFIG_PROPS for VALUE = 'TARGET'	RIB_EXT
MFCS_TARGET	Needs to match the KEY defined on ORCV_BATCH_CONFIG_PROPS for VALUE = 'TARGET'	MFCS_ASNIN_REST

TABLE: ORCV_BATCH_CONFIG_PROPS

MFCS MechIntegrations Rest Service Example.

Program Name	Key	Value
ORCV_SHIP_INT_ASNIN_JOB	MFCS_ASNIN_REST	TARGET
ORCV_SHIP_INT_ASNIN_JOB	MFCS_ASNIN_REST.URL	<MFCS Base URL>/MerchIntegrations/services/inventory/shipmentsAndReceipts/inboundAsn/manage
ORCV_SHIP_INT_ASNIN_JOB	MFCS_ASNIN_REST.ALL_OWEDSCOPES	rgbu:merch:MFCS-<EnvId>
ORCV_SHIP_INT_ASNIN_JOB	MFCS_ASNIN_REST.CLIENT_MAP	oracle.retail.apps.platform.services.mfcs.security.idcs
ORCV_SHIP_INT_ASNIN_JOB	MFCS_ASNIN_REST.CLIENT_KEY	MFCSAlias

RICS RIB -EXT Rest Service Example.

Program Name	Key	Value
ORCV_SHIP_INT_ASNIN_JOB	RIB_EXT	TARGET
ORCV_SHIP_INT_ASNIN_JOB	RIB_EXT.URL	<RICS Base URL>/rib-ext-services-web/resources/publisher/publish
ORCV_SHIP_INT_ASNIN_JOB	RIB_EXT.ALLOWEDSCOPES	rgbu:rics:RICS-<EnvId>
ORCV_SHIP_INT_ASNIN_JOB	RIB_EXT.CLIENT_MAP	oracle.retail.apps.platform.services.rics.security.idcs
ORCV_SHIP_INT_ASNIN_JOB	RIB_EXT.CLIENT_KEY	RICSAlias

Configuration

This section describes configurations.

Initial Batch Configuration for ORCV_SHIP_INT_ASNIN_JOB

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_SHIP_INT_ASNIN_JOB
MAX_CHUNK_SIZE	10000
MAX_CONCURRENT_THREADS	4
BULK_LIMIT	1000
RETRY_LOCK_ATTEMPTS	4
RETRY_WAIT_TIME	2
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Web Service

Frequency: Ad hoc

Authorization Method: OAuth 2.0

Base URL: <BASE_URL>/ShipmentAsnInJavaServicesWeb/services/orcv/merch/
integration/rib/ext/batch/<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": " ORCV_SHIP_INT_ASNIN_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SHIP_INT_ASNIN_SVC ORCV_SHIP_INT_ASNIN_PO_SVC ORCV_SHIP_INT_ASNIN_CTN_SVC ORCV_SHIP_INT_ASNIN_ITEM_SVC ORCV_SHIP_INT_ASNIN_UIN_SVC ORCV_SHIP_INT_ASNIN_CFA_SVC ORCV_SHIP_INT_ASNIN_ERR_SVC ORCV_SHIP_INT_ASNIN ORCV_SHIP_INT_ASNIN_PO ORCV_SHIP_INT_ASNIN_CTN ORCV_SHIP_INT_ASNIN_ITEM ORCV_SHIP_INT_ASNIN_UIN ORCV_SHIP_INT_ASNIN_CFA ORCV_SHIP_INT_ASNIN_CTN_CFA

URL Path	Request Method	Request Body	Affected Tables
/restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_SHIP_INT_ASNIN_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_SHIP_INT_ASNIN_SVC ORCV_SHIP_INT_ASNIN_PO_SVC ORCV_SHIP_INT_ASNIN_CTN_SVC ORCV_SHIP_INT_ASNIN_ITEM_SVC ORCV_SHIP_INT_ASNIN_UIN_SVC ORCV_SHIP_INT_ASNIN_CFA_SVC ORCV_SHIP_INT_ASNIN_ERR_SVC ORCV_SHIP_INT_ASNIN ORCV_SHIP_INT_ASNIN_PO ORCV_SHIP_INT_ASNIN_CTN ORCV_SHIP_INT_ASNIN_ITEM ORCV_SHIP_INT_ASNIN_UIN ORCV_SHIP_INT_ASNIN_CFA ORCV_SHIP_INT_ASNIN_CTN_CFA
/log	GET	URL query parameters: /log? jobName=ORCV_SHIP_INT_ASNIN_JOB &executionId=<EXEC_ID>&date=YYYY-MM-DD	N/A
/kill	POST	{ "executionId": <EXEC_ID>, "jobName": "ORCV_SHIP_INT_ASNIN_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
/status	GET	URL query parameters: /status? jobName=ORCV_SHIP_INT_ASNIN_JOB &executionId=<EXEC_ID>	N/A
/ping	GET	Empty	N/A

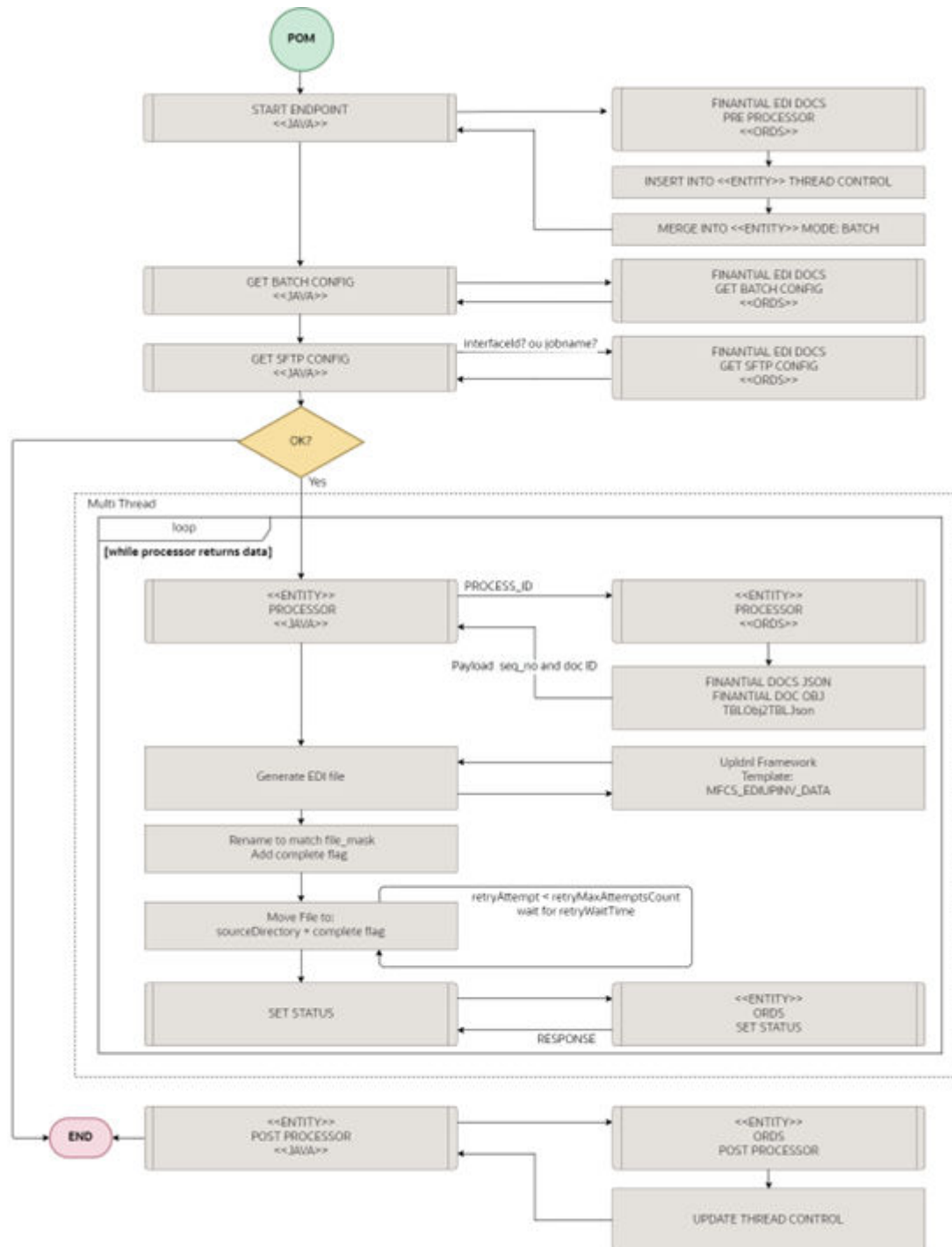
Finance EDI Documents to IMCS

This integration has a maximum chunk size configuration and has a thread aggregator per number of documents for each document type. The chunk size determines the maximum number of documents of a certain type that will be sent per EDI file to IMCS.

Process Overview

The Finance EDI Documents Service batch will support the integration of Financial Documents into IMCS through the EDI injection batch.

Figure 27-62 Finance EDI Documents to IMCS Process Overview



System Parameters

Listed below is the list of configurations at system level that are used by this service.

The parameters are defined in table ORCV_SYSTEM_PARAMETERS.

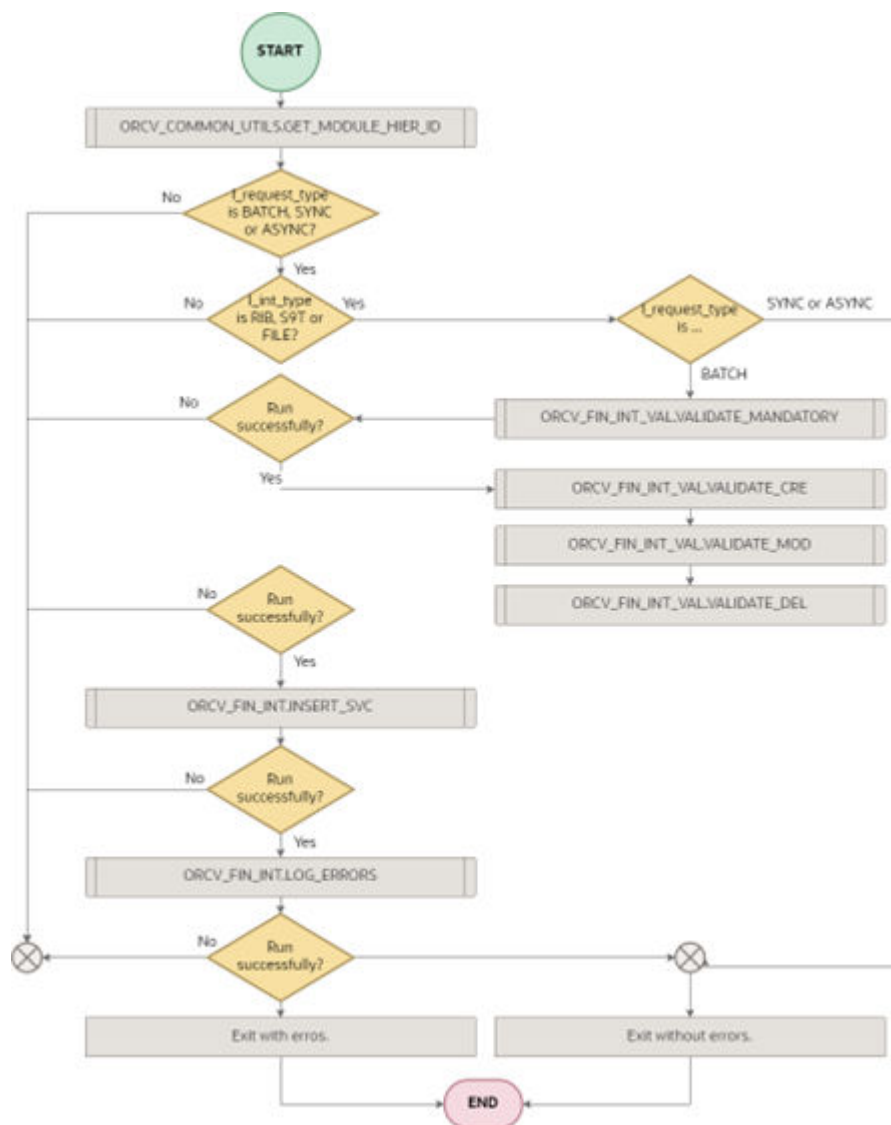
Area	Parameter	Description	Value 1
INTEGRATION	INDUCTION_TEMPLATE	Finance Integration template key	Tax Type = 'SALES' MFCS_EDIPINV_SALES_DATA Tax Type = 'SVAT' MFCS_EDIPINV_SVAT_DATA Tax Type = 'GTS' MFCS_EDIPINV_GTS_DATA

**Note:**

The MFCS system option (defined in the localization_config_options table) sets the default tax type used in the environment.

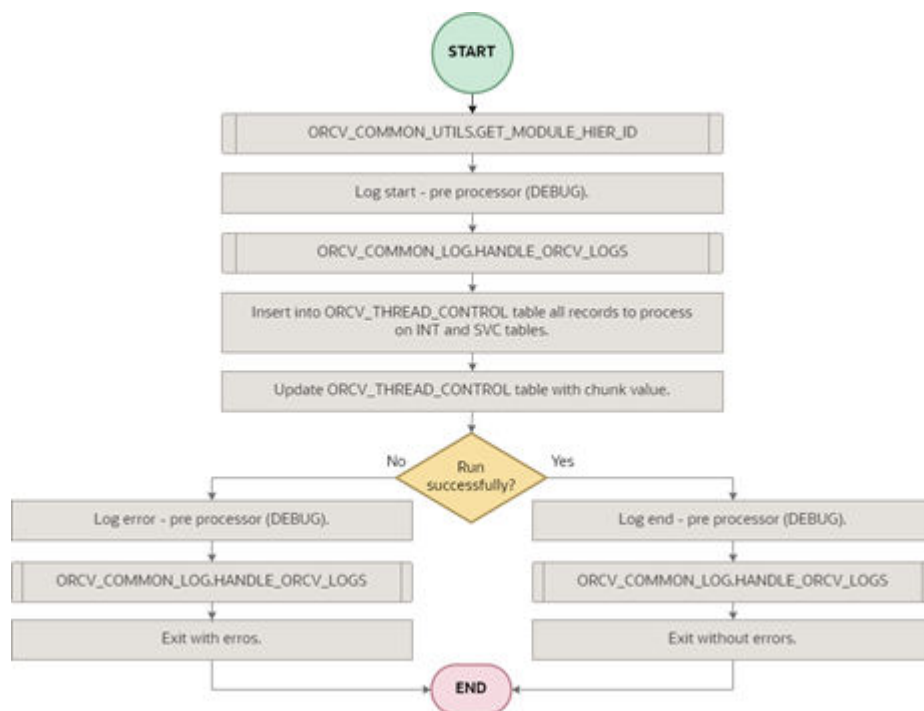
Process EDI

Figure 27-63 Process EDI Overview



Pre Processor

Figure 27-64 Pre-Processor Finance Documents EDI integration



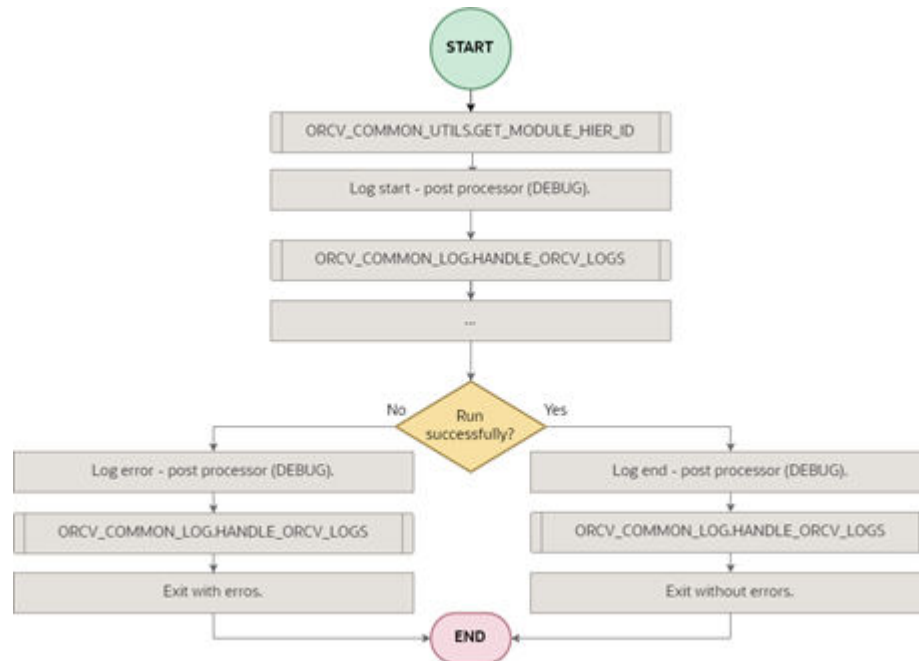
```

graph TD
    START([START]) --> LogStart[ORCV_COMMON_BATCH_AUDIT_LOG.HANDLE_ORCV_BATCH_LOG]
    LogStart --> GetModuleHierId[ORCV_COMMON_UTILS.GET_MODULE_HIER_ID]
    GetModuleHierId --> IsProgramNameNull{1_program_name is NULL?}
    
    IsProgramNameNull -- No --> LogStartProcessor[Log start - processor (DEBUG).]
    LogStartProcessor --> LogHandle[ORCV_COMMON_LOG.HANDLE_ORCV_LOGS]
    LogHandle --> GetThreshold[Get threshold value.]
    GetThreshold --> GetSystemParameters[ORCV_COMMON_UTILS.GET_SYSTEM_PARAMETERS_VALUES]
    GetSystemParameters --> Loop1((LOOP))
    
    IsProgramNameNull -- Yes --> Loop1
    
    Loop1 --> SetRetryFalse[Set retry process to false and decrease 1 retry lock attempt.]
    SetRetryFalse --> Loop1
    
    Loop1 --> InsertInt[ORCV_FIN_INT.INSERT_INT]
    InsertInt --> CheckRecords[Check if have records to process.]
    CheckRecords --> HasAnyToProcess{Has any to process?}
    
    HasAnyToProcess -- No --> GetNewKey[Get new key to process.]
    GetNewKey --> UpdateThread[Update thread no and process status on ORCV_THREAD_CONTROL]
    UpdateThread --> RunSuccessfully1{Run successfully?}
    
    RunSuccessfully1 -- No --> SetRetryTrue[Set retry process to true.]
    SetRetryTrue --> UpdateErrorMessage[Update error message and process status to error on ORCV_THREAD_CONTROL table.]
    UpdateErrorMessage --> EndLoop1((END LOOP))
    EndLoop1 --> Loop1
    
    RunSuccessfully1 -- Yes --> GetEDIDocs[Get EDI Docs to send with limit.]
    GetEDIDocs --> UpdateStatus[Update status to send on ORCV_FIN_INT, tables.]
    UpdateStatus --> UpdateRetryCount[Update retry attempt count on ORCV_FIN_INT, tables.]
    UpdateRetryCount --> UpdateProcessStatus[Update process status to processing on ORCV_THREAD_CONTROL]
    UpdateProcessStatus --> RunSuccessfully2{Run successfully?}
    
    RunSuccessfully2 -- No --> SetRetryTrue
    SetRetryTrue --> UpdateErrorMessage
    UpdateErrorMessage --> EndLoop1
    
    RunSuccessfully2 -- Yes --> DBMSSESSIONSLEEP[DBMS_SESSION.SLEEP]
    DBMSSESSIONSLEEP --> EndLoop1
    
    EndLoop1 --> LogEnd[ORCV_COMMON_BATCH_AUDIT_LOG.HANDLE_ORCV_BATCH_LOG]
    LogEnd --> LogEndProcessor[Log end - processor (DEBUG).]
    LogEndProcessor --> LogHandleEnd[ORCV_COMMON_LOG.HANDLE_ORCV_LOGS]
    LogHandleEnd --> RunSuccessfully3{Run successfully?}
    
    RunSuccessfully3 -- No --> LogError[Log error - processor (DEBUG).]
    LogError --> LogHandleError[ORCV_COMMON_LOG.HANDLE_ORCV_LOGS]
    LogHandleError --> ExitWithErrors[Exit with errors.]
    ExitWithErrors --> END([END])
    
    RunSuccessfully3 -- Yes --> LogEndProcessor
    LogEndProcessor --> LogHandleEnd
    LogHandleEnd --> ExitWithoutErrors[Exit without errors.]
    ExitWithoutErrors --> END
  
```

The flowchart illustrates the process of handling ORCV batch audit logs. It begins with a 'START' terminal, leading to the 'ORCV_COMMON_BATCH_AUDIT_LOG.HANDLE_ORCV_BATCH_LOG' process. This is followed by 'ORCV_COMMON_UTILS.GET_MODULE_HIER_ID'. A decision point '1_program_name is NULL?' determines the next steps. If 'No', it proceeds to 'Log start - processor (DEBUG)', 'ORCV_COMMON_LOG.HANDLE_ORCV_LOGS', 'Get threshold value.', and 'ORCV_COMMON_UTILS.GET_SYSTEM_PARAMETERS_VALUES', entering a 'LOOP'. If 'Yes', it also enters the 'LOOP'. Inside the 'LOOP', 'Set retry process to false and decrease 1 retry lock attempt.' is executed. The process then moves to 'ORCV_FIN_INT.INSERT_INT' and 'Check if have records to process.'. A decision 'Has any to process?' follows. If 'No', it goes to 'Get new key to process.', then 'Update thread no and process status on ORCV_THREAD_CONTROL', and a 'Run successfully?' decision. If 'No', it sets the retry process to true, updates the error message and process status to error on 'ORCV_THREAD_CONTROL' table, and ends the loop. If 'Yes', it proceeds to 'Get EDI Docs to send with limit.', 'Update status to send on ORCV_FIN_INT, tables.', 'Update retry attempt count on ORCV_FIN_INT, tables.', and 'Update process status to processing on ORCV_THREAD_CONTROL'. Another 'Run successfully?' decision follows. If 'No', it sets the retry process to true, updates the error message and process status to error on 'ORCV_THREAD_CONTROL' table, and ends the loop. If 'Yes', it executes 'DBMS_SESSION.SLEEP' and ends the loop. After the loop, it logs the end of the process ('ORCV_COMMON_BATCH_AUDIT_LOG.HANDLE_ORCV_BATCH_LOG', 'Log end - processor (DEBUG)', 'ORCV_COMMON_LOG.HANDLE_ORCV_LOGS') and checks 'Run successfully?'. If 'No', it logs an error ('Log error - processor (DEBUG)', 'ORCV_COMMON_LOG.HANDLE_ORCV_LOGS') and exits with errors. If 'Yes', it logs the end of the process and exits without errors. The process concludes at the 'END' terminal.

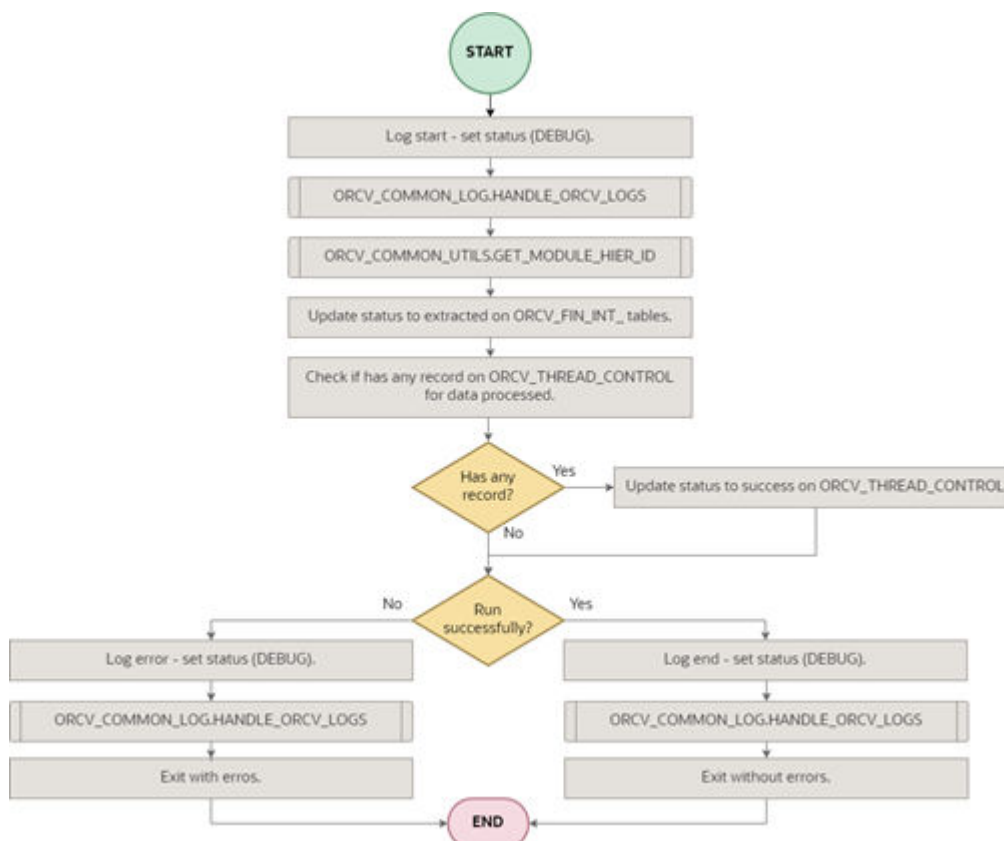
Post Processor

Figure 27-66 Post Processor Finance Documents EDI integration



Set Integration Status

Figure 27-67 Set Integration Status Flow for Finance Documents EDI Integration



Configuration

The finance EDI documents integration service has a prerequisite of the interface with ID FINEDI being configured in ORCV_BATCH_SFTP and ORCV_BATCH_SFTP_TARGET tables.



Note:

POM configuration for IMCS REIM_EDI_INJECTOR_JOB set to run at a high frequency if required.

Initial Batch Configuration for ORCV_FIN_INT_EDI_JOB

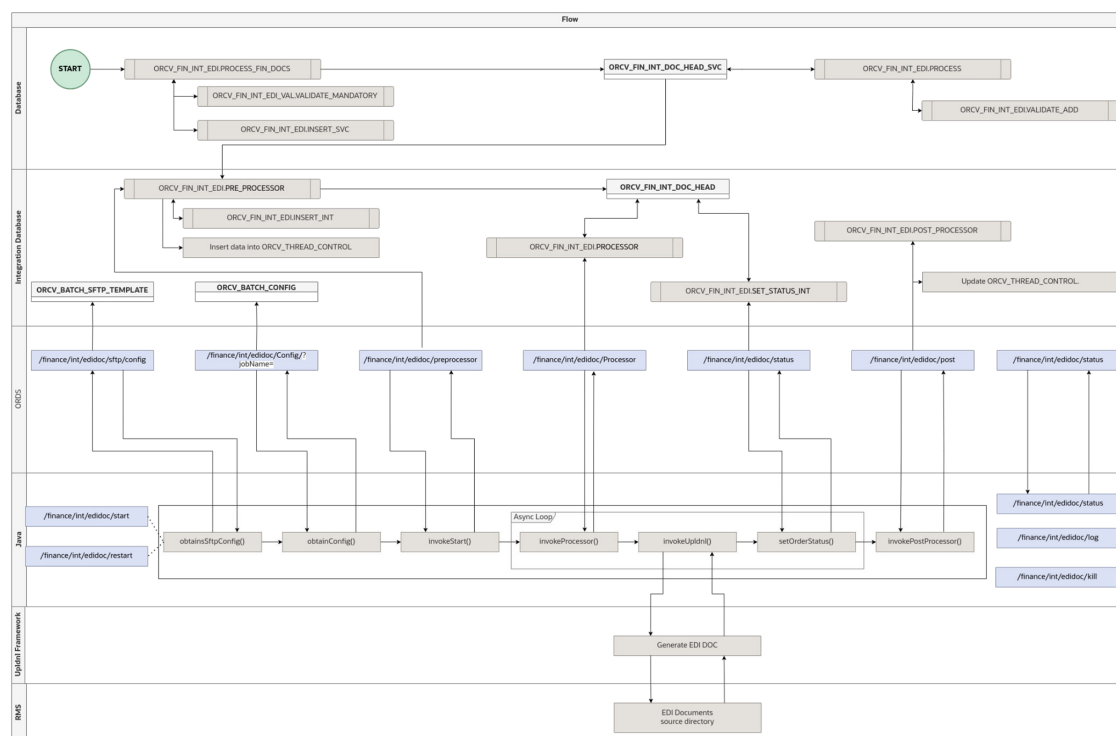
Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_FIN_INT_EDI_JOB
MAX_CHUNK_SIZE	10000

Column Name	Default Values
MAX_CONCURRENT_THREADS	4
BULK_LIMIT	1000
RETRY_LOCK_ATTEMPTS	4
RETRY_WAIT_TIME	2
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Batch Flow

Figure 27-68 Finance Documents EDI integration Batch Flow



Web Service

Frequency: Ad hoc

Authorization Method: OAuth 2.0

Base URL: <BASE_URL>/ FinanceEdiDocsJavaServicesWeb/services/orcv/merch/finance/integration/

edidocs/<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
batch/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_FIN_INT_EDI_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_FIN_INT_DOC_HEAD_SVC ORCV_FIN_INT_DOC_HEAD_CMTS_SVC ORCV_FIN_INT_DOC_TX_SVC ORCV_FIN_INT_DOC_DTL_CMTS_SVC ORCV_FIN_INT_DOC_DTL_RC_SVC ORCV_FIN_INT_DOC_DTL_TAX_SVC ORCV_FIN_INT_DOC_NON_MERCH_SVC ORCV_FIN_INT_DOC_N_MCH_TX_SVC ORCV_FIN_INT_DOC_PO_SVC ORCV_FIN_INT_INV_DTL_SVC ORCV_FIN_INT_ERR_SVC ORCV_FIN_INT_DOC_HEAD ORCV_FIN_INT_DOC_HEAD_CMTS ORCV_FIN_INT_DOC_NON_MERCH ORCV_FIN_INT_DOC_NON_MERCH_TX ORCV_FIN_INT_DOC_PO ORCV_FIN_INT_DOC_TX ORCV_FIN_INT_DOC_XREF ORCV_FIN_INT_DOC_DTL_CMTS ORCV_FIN_INT_DOC_DTL_RC ORCV_FIN_INT_DOC_DTL_TAX ORCV_FIN_INT_DOC_PO ORCV_FIN_INT_INV_DTL

URL Path	Request Method	Request Body	Affected Tables
batch/restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_FIN_INT_EDI_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_FIN_INT_DOC_HEAD_SVC ORCV_FIN_INT_DOC_HEAD_CMTS_SVC ORCV_FIN_INT_DOC_TX_SVC ORCV_FIN_INT_DOC_DTL_CMTS_SVC ORCV_FIN_INT_DOC_DTL_RC_SVC ORCV_FIN_INT_DOC_DTL_TAX_SVC ORCV_FIN_INT_DOC_NON_MERCH_SVC ORCV_FIN_INT_DOC_N_MCH_TX_SVC ORCV_FIN_INT_DOC_PO_SVC ORCV_FIN_INT_INV_DTL_SVC ORCV_FIN_INT_ERR_SVC ORCV_FIN_INT_DOC_HEAD ORCV_FIN_INT_DOC_HEAD_CMTS ORCV_FIN_INT_DOC_NON_MERCH ORCV_FIN_INT_DOC_NON_MERCH_TX ORCV_FIN_INT_DOC_PO ORCV_FIN_INT_DOC_TX ORCV_FIN_INT_DOC_XREF ORCV_FIN_INT_DOC_DTL_CMTS ORCV_FIN_INT_DOC_DTL_RC ORCV_FIN_INT_DOC_DTL_TAX ORCV_FIN_INT_DOC_PO ORCV_FIN_INT_INV_DTL
batch/log	GET	URL query parameters: /log? jobName=ORCV_FIN_INT_EDI_JOB &executionId=<EXEC_ID>&date=YYYY-MM-DD	N/A
batch/kill	POST	{ "executionId": <EXEC_ID>, "jobName": "ORCV_FIN_INT_EDI_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
batch/status	GET	URL query parameters: /status? jobName=ORCV_FIN_INT_EDI_JOB &executionId=<EXEC_ID>	N/A

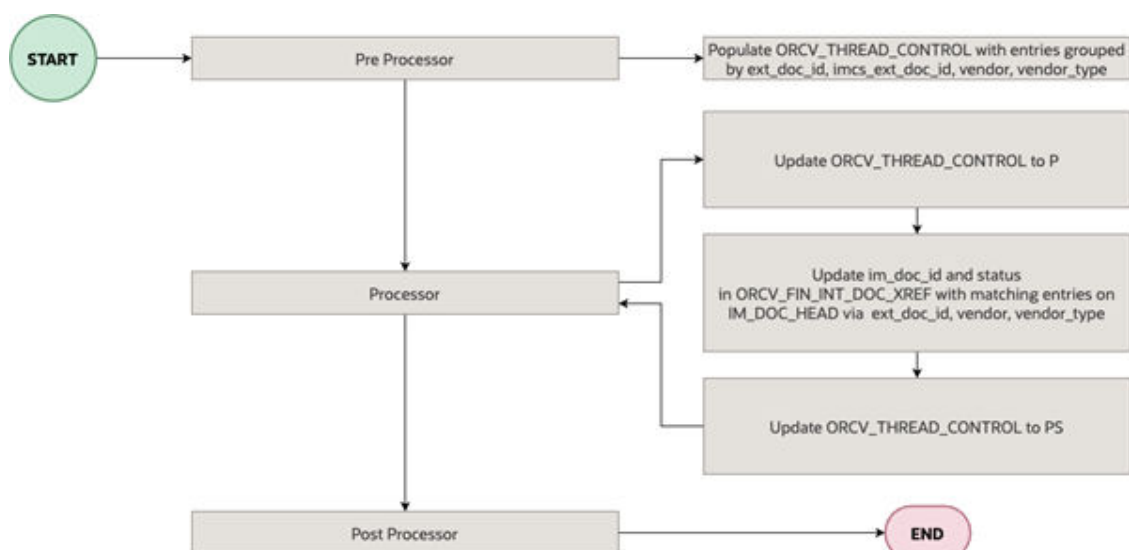
URL Path	Request Method	Request Body	Affected Tables
/ping	GET	Empty	N/A

Finance EDI Document XREF Service

Finance EDI Document XREF service updates the integrated Finance Document ids from IMCS in table ORCV_FIN_INT_DOC_XREF.

- Job: ORCV_FIN_INT_EDI_XREF_JOB
- Frequency: Ad hoc
- Dependency: None
- Inter Scheduler Dependency: None

Figure 27-69 Finance EDI Document XREF Batch Service Overview



Initial Batch Configuration

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_FIN_INT_EDI_XREF_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	3
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Web Service

Authentication Method: OAuth2

BaseURL: <BASE_URL>/orcv/merch/finance/integration/xrefupdate/batch/<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_FIN_INT_EDIXREF_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_FIN_INT_DOC_XREF
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_FIN_INT_EDIXREF_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_FIN_INT_DOC_XREF
log	GET	URL query parameters: /log? jobName=ORCV_FIN_INT_EDIXREF_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A
kill	POST	Empty	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
status	GET	URL query parameters: /status? jobName=ORCV_FIN_INT_EDIXREF_JOB &executionId=<EXEC_ID>&logDate=YYYY-DD-MM	N/A
ping	GET	Empty	N/A

Item Induction to MFCS

This integration batch has a maximum chunk size configuration and has a thread aggregator per number of level-one items to process. The chunk size represents the maximum number of

level-one items that each thread will pick per iteration and maps it to the number of level-one items each output induction ODS file will have and that will be integrated to MFCS through IINDBATCH_JOB.



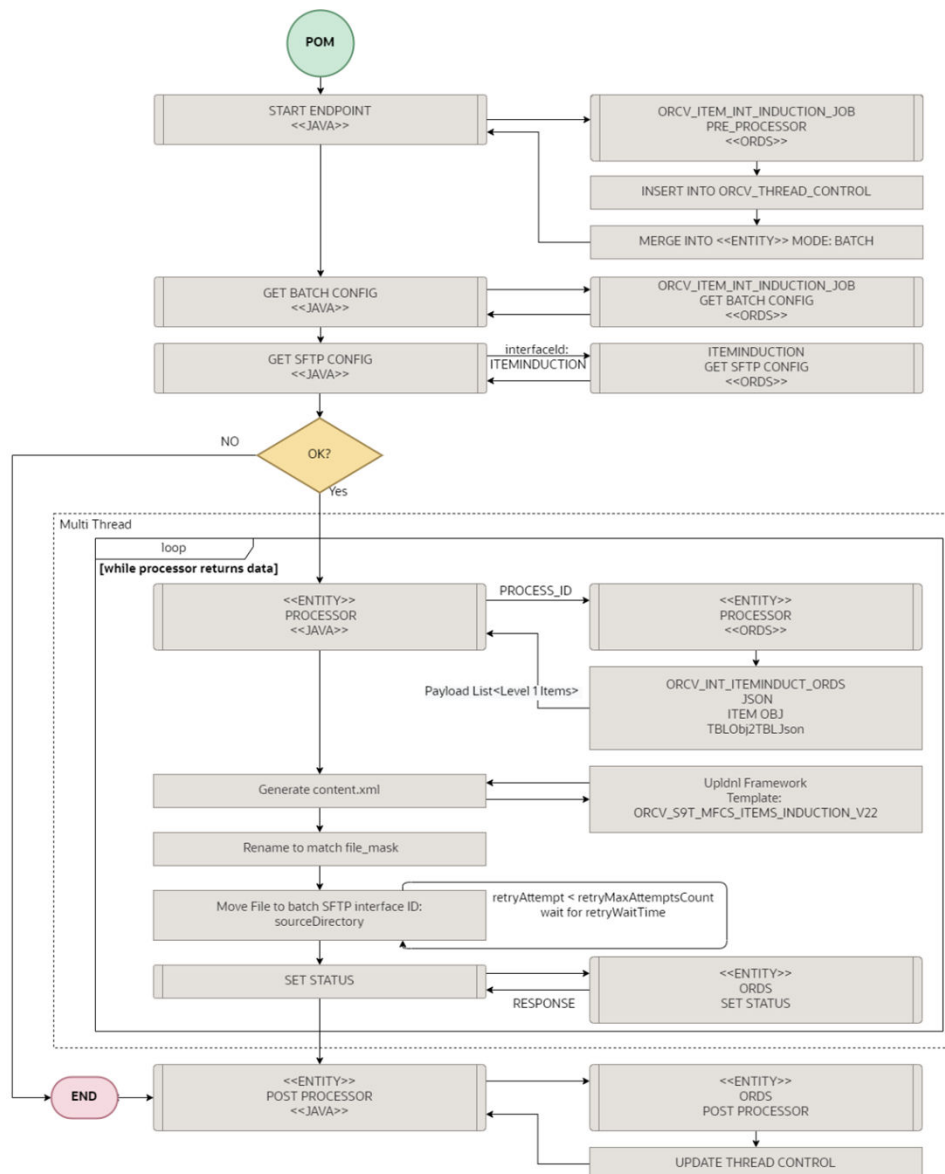
Note:

Item Induction has different configurations depending on the tax type defined in MFCS, in case the tax type is changed in MFCS the template should also be updated.

Process Overview

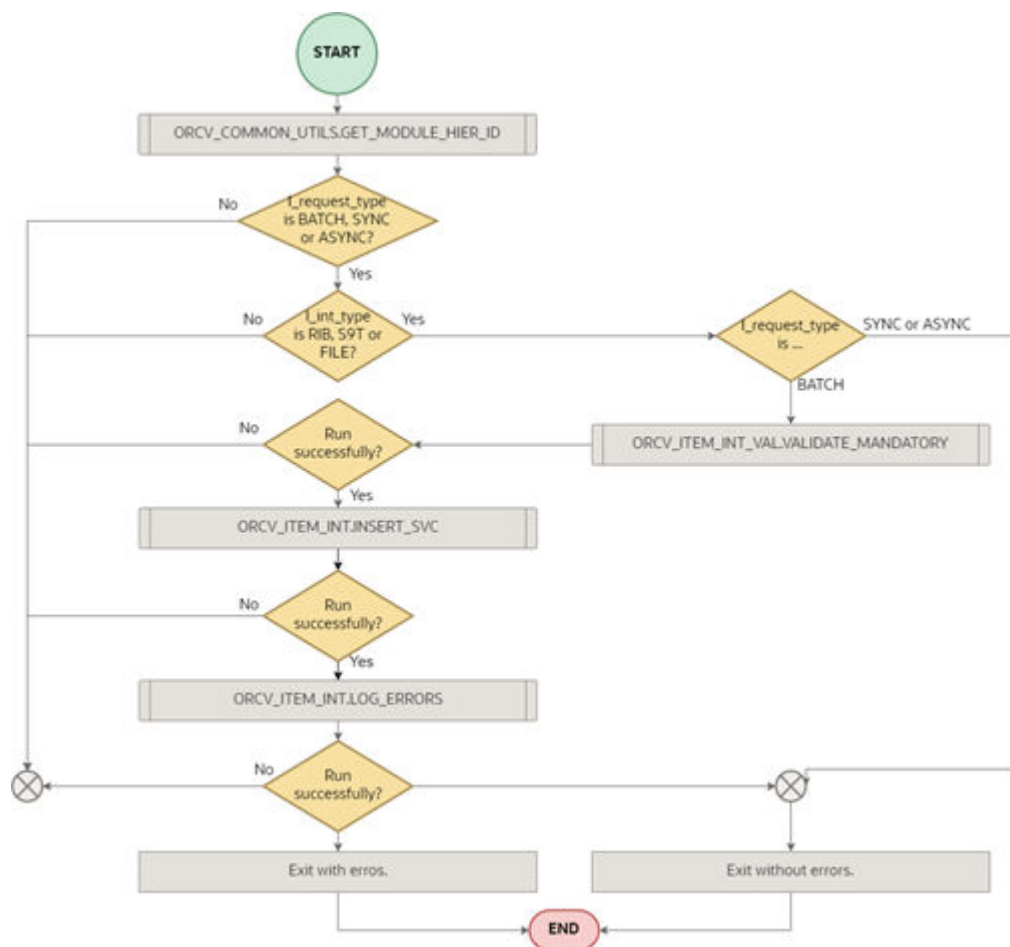
The Item induction service will support the integration of Items into MFCS through the induction batch.

Figure 27-70 Item Induction to MFCS Process Overview



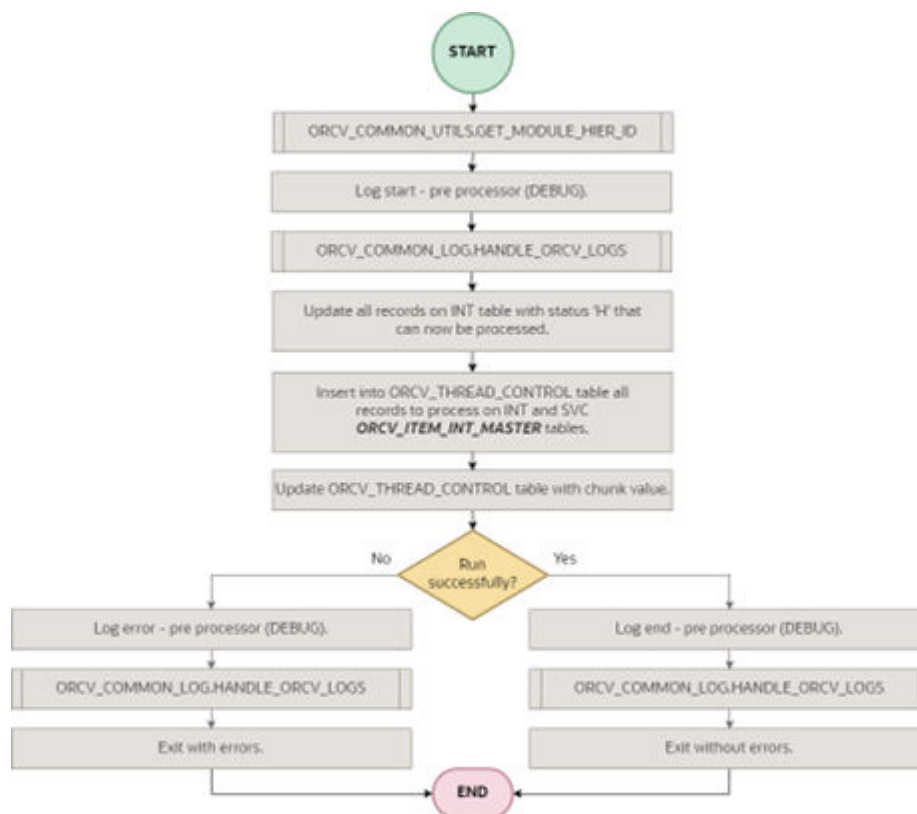
Process Item

Figure 27-71 Process Item Overview



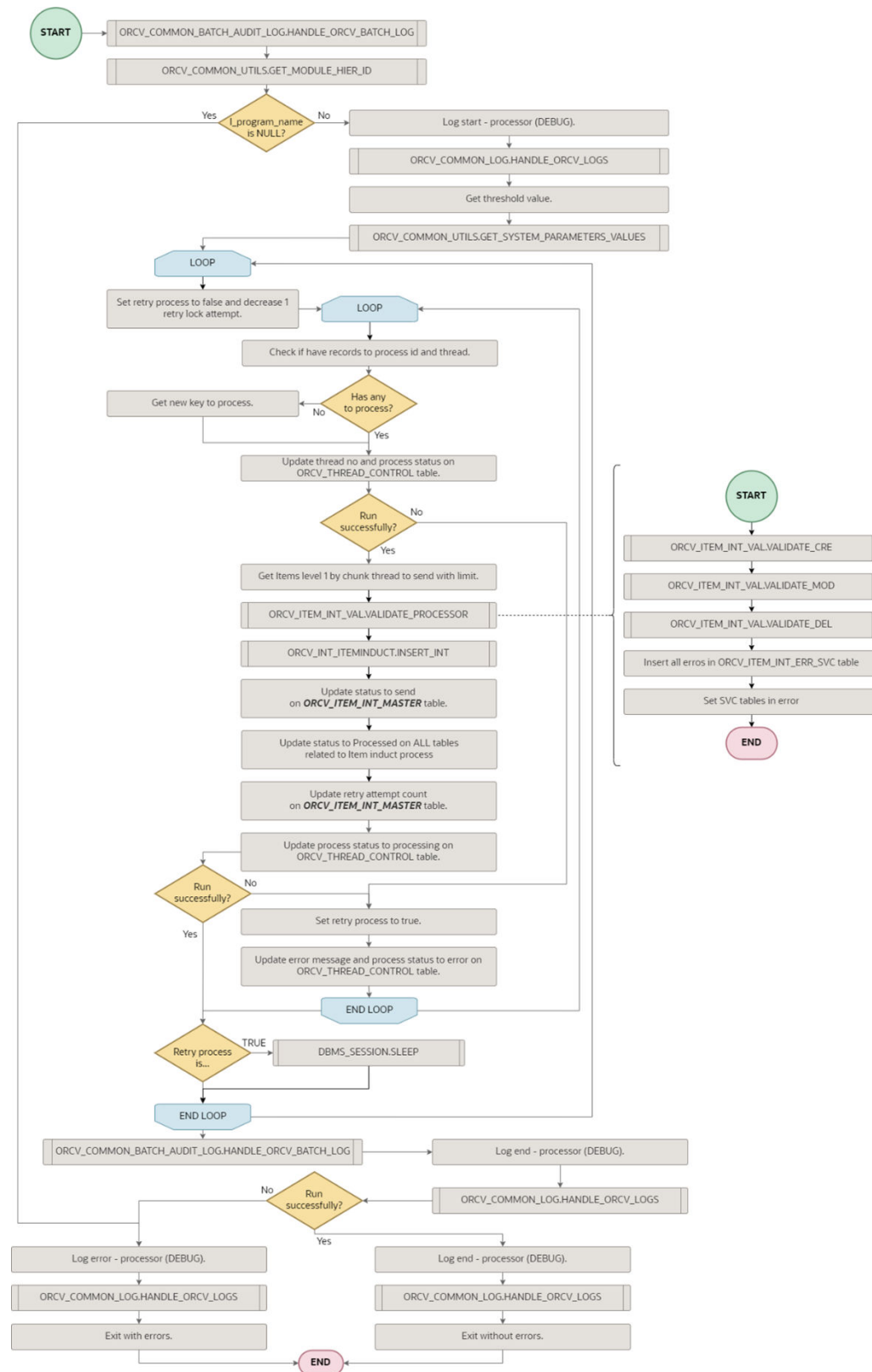
Pre-Processor

Figure 27-72 Pre-Processor Item Induction MFCS Integration



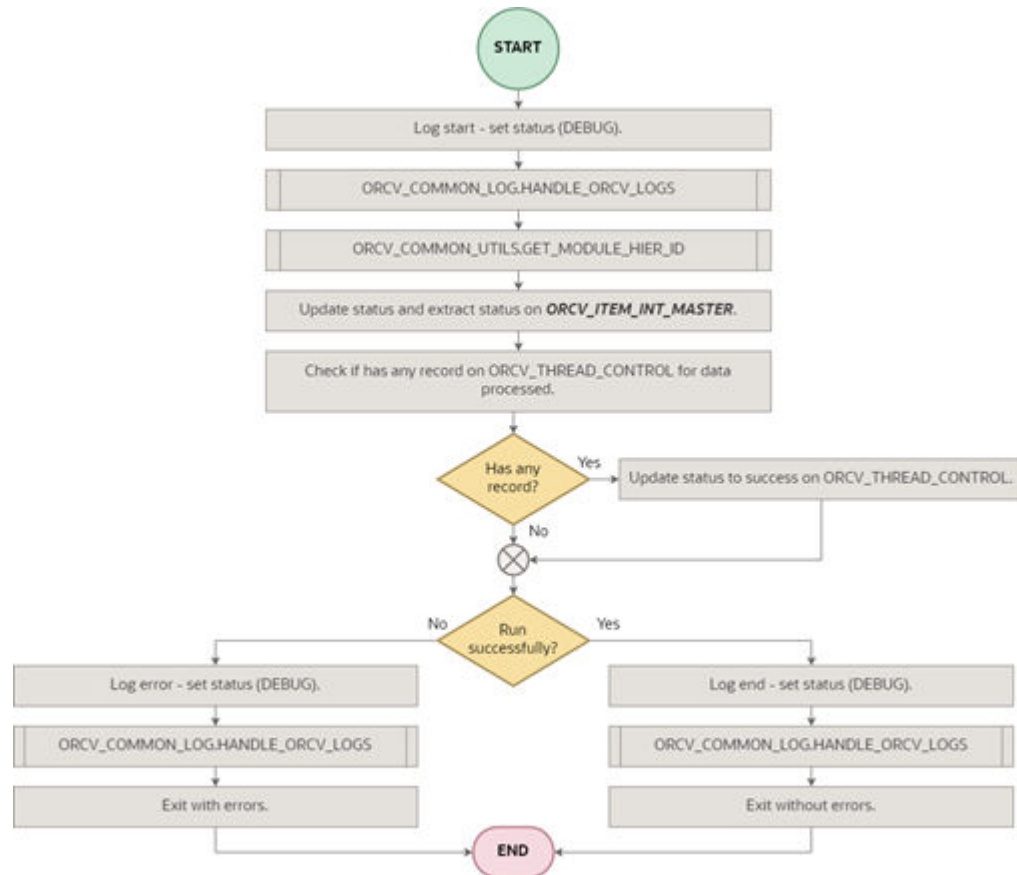
Processor

Figure 27-73 Processor Item Induction MFCS Integration



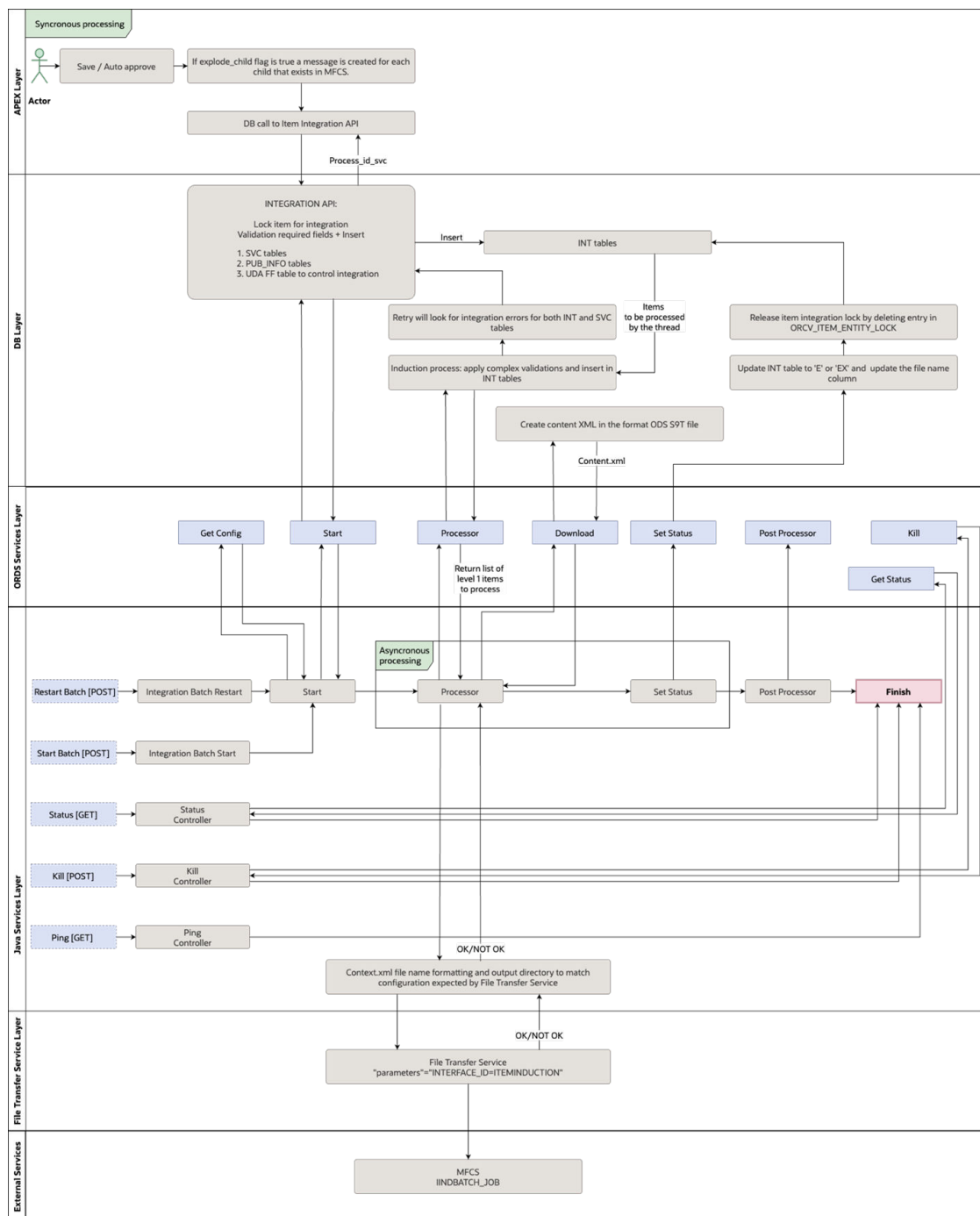
Set Integration Status

Figure 27-74 Set Integration Status Flow for Item Induction MFCS Integration



Batch Flow

Figure 27-75 Item Induction MFCS Integration Batch Flow



Configuration

The Item Induction integration service has a prerequisite of the interface with ID ITEMINDUCTION being configured in ORCV_BATCH_SFTP and ORCV_BATCH_SFTP_TARGET tables.

**Note:**

POM configuration for MFCS IINDBATCH_JOB set to run at a high frequency if required.

Initial Batch Configuration for ORCV_ITEM_INT_INDUCTION_JOB

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_ITEM_INT_INDUCTION_JOB
MAX_CHUNK_SIZE	1
MAX_CONCURRENT_THREADS	4
BULK_LIMIT	1000
RETRY_LOCK_ATTEMPTS	4
RETRY_WAIT_TIME	2
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

**Note:**

Worksheet VAT_ITEM will not be displayed if system option (defined in IMFCS localization_config_options table) has GTS or SALES value

System Parameters

Listed below is the list of configurations at system level that are required by this integration.

The parameters are defined in table ORCV_SYSTEM_PARAMETERS.

Area	Parameter	Description	Value 1
INTEGRATION	INDUCTION_TEMPLAT E	Template key.	ORCV_S9T_MFCS_ITEMS_INDUCTION_V 24

Web Service

Frequency: Ad hoc

Authorization Method: OAuth 2.0

Base URL: <BASE_URL>/ItemInductionJavaServicesWeb/services/orcv/merch/item/
integration/iteminduct /<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
batch/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_ITEM_INT_INDUCTION_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_ITEM_INT_IMAGE_SVC ORCV_ITEM_INT_SEASONS_SVC ORCV_ITEM_INT HTS_SVC ORCV_ITEM_INT HTS_ASSESS_SVC ORCV_ITEM_INT_IMAGE_TL_SVC ORCV_ITEM_INT_SUP_TL_SVC ORCV_ITEM_INT_MASTER_TL_SVC ORCV_ITEM_INT_XFORM_H_TL_SVC ORCV_ITEM_INT_EXPENSES_SVC ORCV_ITEM_INT_TICKET_SVC ORCV_ITEM_INT_CHRG_SVC ORCV_ITEM_INT_MASTER ORCV_ITEM_INT_SUP ORCV_ITEM_INT_SUPCTRY ORCV_ITEM_INT_SUPCTRY_DIM ORCV_ITEM_INT_SUP_CTRYLOC ORCV_ITEM_INT_SUP_MN_CTRY ORCV_ITEM_INT_SUP_UOM ORCV_ITEM_INT_XFORM_HEAD ORCV_ITEM_INT_XF_H_TL_PUBINFO ORCV_ITEM_INT_EXPENSES_PUBINFO ORCV_ITEM_INT HTS_PUBINFO ORCV_ITEM_INT_TICKET_PUBINFO ORCV_ITEM_INT_CHRG_PUBINFO ORCV_ITEM_INT_EXPENSES ORCV_ITEM_INT_TICKET ORCV_ITEM_INT_CHRG ORCV_ITEM_INT_ERR_SVC ORCV_ITEM_INT_MASTER_PUBINFO ORCV_ITEM_INT_SUP_PUBINFO ORCV_ITEM_INT_SUPCTRY_PUBINFO ORCV_ITEM_INT_SUPCY_DM_PUBINFO ORCV_ITEM_INT_SUP_CYLC_PUBINFO ORCV_ITEM_INT_SUP_UOM_PUBINFO ORCV_ITEM_INT_SUP_MNCY_PUBINFO ORCV_ITEM_INT_XF_HEAD_PUBINFO ORCV_ITEM_INT_XF_DTL_PUBINFO ORCV_ITEM_INT_PACKITEM_PUBINFO

URL Path	Request Method	Request Body	Affected Tables
			ORCV_ITEM_INT_IZP_PUBINFO ORCV_ITEM_INT_VAT_ITEM_PUBINFO ORCV_ITEM_INT_UDA_LOV_PUBINFO ORCV_ITEM_INT_UDA_DATE_PUBINFO ORCV_ITEM_INT_UDA_FF_PUBINFO ORCV_ITEM_INT_SEASONS_PUBINFO ORCV_ITEM_INT_IMAGE_PUBINFO ORCV_ITEM_INT_IMAGE_TL_PUBINFO ORCV_ITEM_INT HTS_A_PUBINFO ORCV_ITEM_INT_IEM_TL_PUBINFO ORCV_ITEM_INT_SUP_TL_PUBINFO ORCV_ITEM_INT_XFORM_DETAIL ORCV_ITEM_INT_PACKITEM ORCV_ITEM_INT_IZP ORCV_ITEM_INT_VAT_ITEM ORCV_ITEM_INT_UDA_LOV ORCV_ITEM_INT_UDA_DATE ORCV_ITEM_INT_UDA_FF ORCV_ITEM_INT_IMAGE ORCV_ITEM_INT_SEASONS ORCV_ITEM_INT HTS ORCV_ITEM_INT HTS_ASSESS ORCV_ITEM_INT_IMAGE_TL ORCV_ITEM_INT_SUP_TL ORCV_ITEM_INT_MASTER_TL ORCV_ITEM_INT_XFORM_H_TL ORCV_ITEM_INT_MASTER_SVC ORCV_ITEM_INT_SUP_SVC ORCV_ITEM_INT_SUPCTRY_SVC ORCV_ITEM_INT_SUPCTRY_DIM_SVC ORCV_ITEM_INT_SUP_CTRYLOC_SVC ORCV_ITEM_INT_SUP_MN_CTRY_SVC ORCV_ITEM_INT_SUP_UOM_SVC ORCV_ITEM_INT_XFORM_HEAD_SVC ORCV_ITEM_INT_XFORM_DETAIL_SVC ORCV_ITEM_INT_PACKITEM_SVC ORCV_ITEM_INT_IZP_SVC ORCV_ITEM_INT_VAT_ITEM_SVC ORCV_ITEM_INT_UDA_LOV_SVC ORCV_ITEM_INT_UDA_DATE_SVC ORCV_ITEM_INT_UDA_FF_SVC

URL Path	Request Method	Request Body	Affected Tables
batch/ restart	POST	{ "executionId":<EXEC_ID>, "jobName":"ORCV_ITEM_INT_ INDUCTION_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_ITEM_INT_IMAGE_SVC ORCV_ITEM_INT_SEASONS_SVC ORCV_ITEM_INT HTS_SVC ORCV_ITEM_INT HTS_ASSESS_SVC ORCV_ITEM_INT_IMAGE_TL_SVC ORCV_ITEM_INT_SUP_TL_SVC ORCV_ITEM_INT_MASTER_TL_SVC ORCV_ITEM_INT_XFORM_H_TL_SVC ORCV_ITEM_INT_EXPENSES_SVC ORCV_ITEM_INT_TICKET_SVC ORCV_ITEM_INT_CHRG_SVC ORCV_ITEM_INT_MASTER ORCV_ITEM_INT_SUP ORCV_ITEM_INT_SUPCTRY ORCV_ITEM_INT_SUPCTRY_DIM ORCV_ITEM_INT_SUP_CTRYLOC ORCV_ITEM_INT_SUP_MN_CTRY ORCV_ITEM_INT_SUP_UOM ORCV_ITEM_INT_XFORM_HEAD ORCV_ITEM_INT_XF_H_TL_PUBINFO ORCV_ITEM_INT_EXPENSES_PUBINFO ORCV_ITEM_INT HTS_PUBINFO ORCV_ITEM_INT_TICKET_PUBINFO ORCV_ITEM_INT_CHRG_PUBINFO ORCV_ITEM_INT_EXPENSES ORCV_ITEM_INT_TICKET ORCV_ITEM_INT_CHRG ORCV_ITEM_INT_ERR_SVC ORCV_ITEM_INT_MASTER_PUBINFO ORCV_ITEM_INT_SUP_PUBINFO ORCV_ITEM_INT_SUPCTRY_PUBINFO ORCV_ITEM_INT_SUPCY_DM_PUBINFO ORCV_ITEM_INT_SUP_CYLC_PUBINFO ORCV_ITEM_INT_SUP_UOM_PUBINFO ORCV_ITEM_INT_SUP_MNCY_PUBINFO ORCV_ITEM_INT_XF_HEAD_PUBINFO ORCV_ITEM_INT_XF_DTL_PUBINFO

URL Path	Request Method	Request Body	Affected Tables
			ORCV_ITEM_INT_PACKITEM_PUBINFO ORCV_ITEM_INT_IZP_PUBINFO ORCV_ITEM_INT_VAT_ITEM_PUBINFO ORCV_ITEM_INT_UDA_LOV_PUBINFO ORCV_ITEM_INT_UDA_DATE_PUBINFO ORCV_ITEM_INT_UDA_FF_PUBINFO ORCV_ITEM_INT_SEASONS_PUBINFO ORCV_ITEM_INT_IMAGE_PUBINFO ORCV_ITEM_INT_IMAGE_TL_PUBINFO ORCV_ITEM_INT HTS_A_PUBINFO ORCV_ITEM_INT_IEM_TL_PUBINFO ORCV_ITEM_INT_SUP_TL_PUBINFO ORCV_ITEM_INT_XFORM_DETAIL ORCV_ITEM_INT_PACKITEM ORCV_ITEM_INT_IZP ORCV_ITEM_INT_VAT_ITEM ORCV_ITEM_INT_UDA_LOV ORCV_ITEM_INT_UDA_DATE ORCV_ITEM_INT_UDA_FF ORCV_ITEM_INT_IMAGE ORCV_ITEM_INT_SEASONS ORCV_ITEM_INT HTS ORCV_ITEM_INT HTS_ASSESS ORCV_ITEM_INT_IMAGE_TL ORCV_ITEM_INT_SUP_TL ORCV_ITEM_INT_MASTER_TL ORCV_ITEM_INT_XFORM_H_TL ORCV_ITEM_INT_MASTER_SVC ORCV_ITEM_INT_SUP_SVC ORCV_ITEM_INT_SUPCTRY_SVC ORCV_ITEM_INT_SUPCTRY_DIM_SVC ORCV_ITEM_INT_SUP_CTRYLOC_SVC ORCV_ITEM_INT_SUP_MN_CTRY_SVC ORCV_ITEM_INT_SUP_UOM_SVC ORCV_ITEM_INT_XFORM_HEAD_SVC ORCV_ITEM_INT_XFORM_DETAIL_SVC ORCV_ITEM_INT_PACKITEM_SVC ORCV_ITEM_INT_IZP_SVC ORCV_ITEM_INT_VAT_ITEM_SVC ORCV_ITEM_INT_UDA_LOV_SVC ORCV_ITEM_INT_UDA_DATE_SVC

URL Path	Request Method	Request Body	Affected Tables
			ORCV_ITEM_INT_UDA_FF_SVC
batch/log	GET	URL query parameters: /log?jobName=ORCV_ITEM_INT_INDUCATION_JOB &executionId=<EXEC_ID>&date=YYYY-MM-DD	N/A
batch/kill	POST	{ "executionId":<EXEC_ID>, "jobName": "ORCV_ITEM_INT_INDUCATION_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
batch/status	GET	URL query parameters: /status?jobName=ORCV_ITEM_INT_INDUCATION_JOB&executionId=<EXEC_ID>	N/A
/ping	GET	Empty	N/A

Item CFAs Rest to MFCS

Item integration to MFCS MerchIntegration extends the Database Integration Framework to publish Items CFAs for Item Master, Item Supplier and Item Supplier Country.

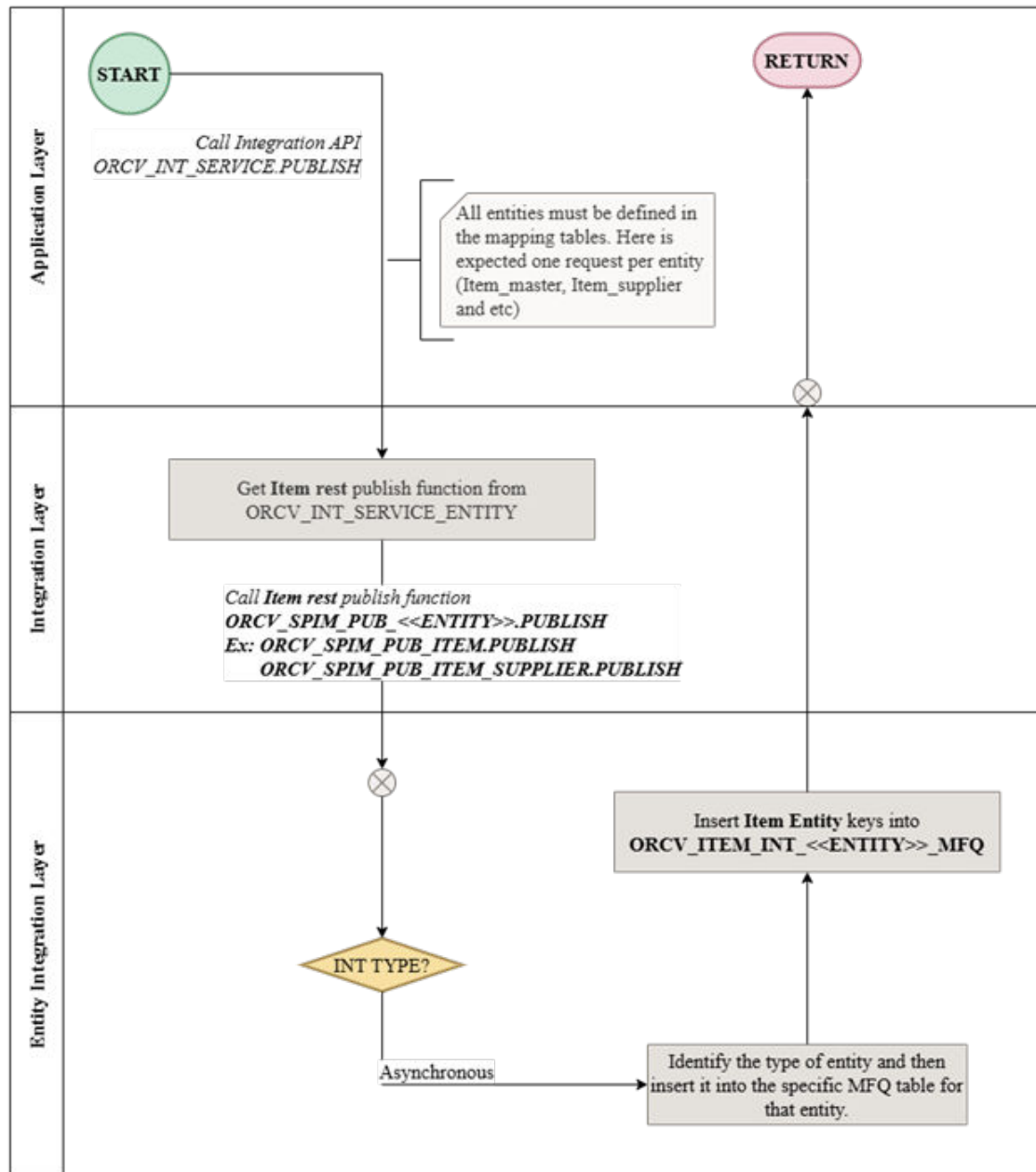
Process Overview

This section describes the process overview.

Publish Flow

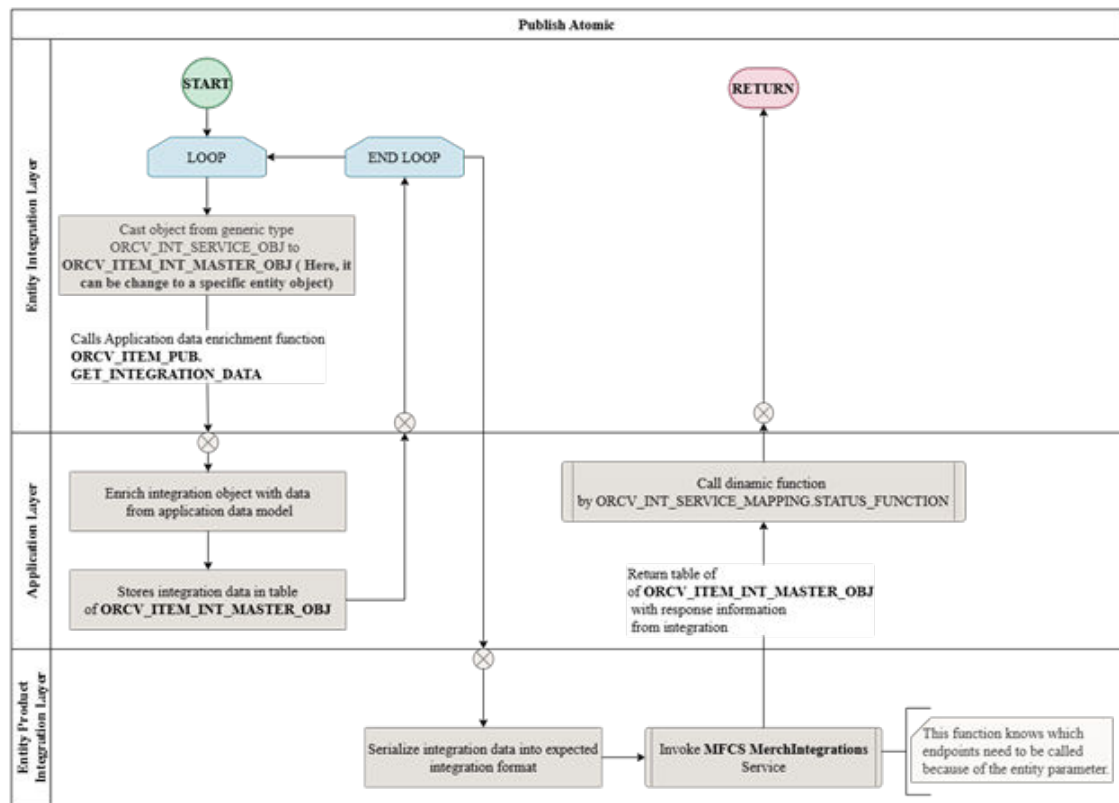
Publish allows integration endpoints to be called in either synchronous or asynchronous mode. These calls can be atomic or non-atomic. Atomic calls ensure that all entries are integrated together as a bulk operation, while non-atomic calls handle each entry individually.

Figure 27-76 Items CFAs to MFCS Publish Flow



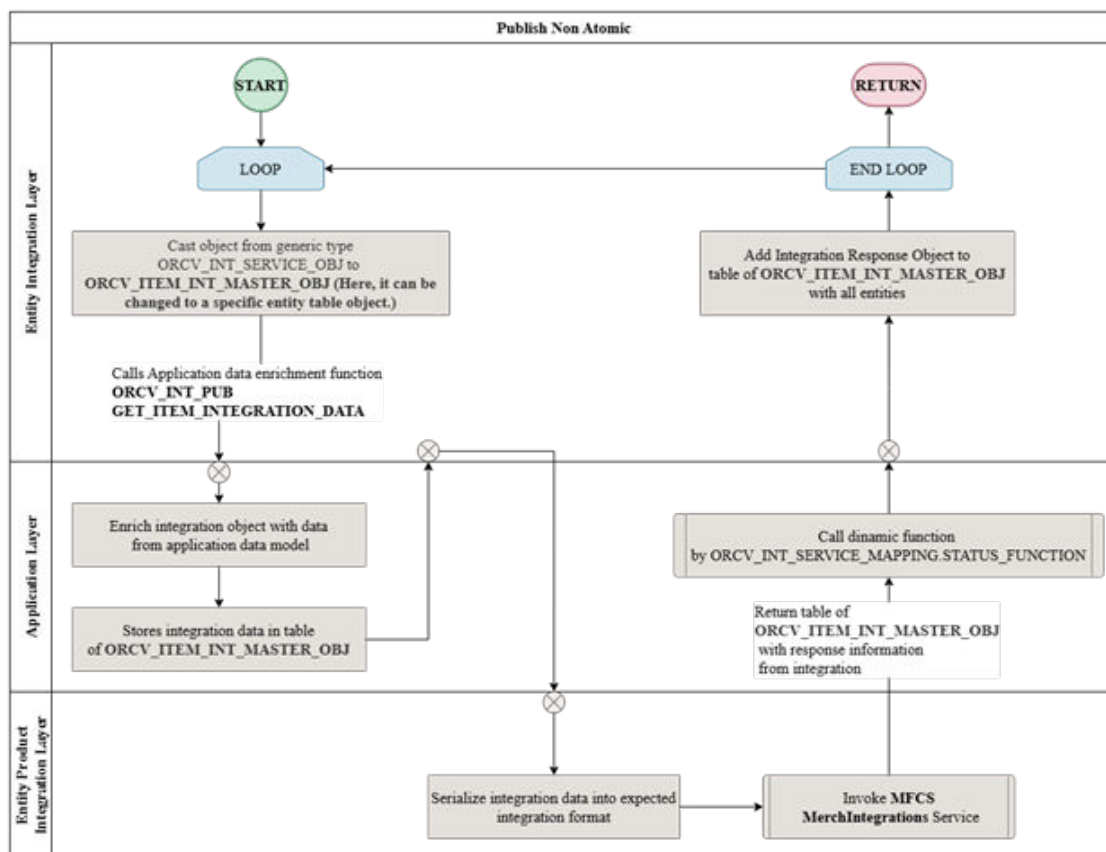
Publish Atomic

Figure 27-77 Items CFAs to MFCS Publish Atomic



Publish Non Atomic

Figure 27-78 Items CFAs to MFCS Publish Non Atomic



Configuration

Items publish Items CFAs with the following configurations in Database Integration Framework.

Table ORCV_INT_SERVICE_MAPPING

Item Master

Configuration	Value
Application Alias	ITEM
Application Entity Get Data Function	ORCV_ITEM_INT_ML_WS.GET_ITEM_INTEGRATION_DATA
Application Status Function	NA
Family	ITEM_MASTER
Atomic Indicator	N

Item Supplier

Configuration	Value
Application Alias	ITEM
Application Entity Get Data Function	ORCV_ITEM_INT_ML_WS.GET_ITEM_INTEGRATION_DATA
Application Status Function	NA
Family	ITEM_SUPPLIER
Atomic Indicator	N

Item Supplier Country

Configuration	Value
Application Alias	ITEM
Application Entity Get Data Function	ORCV_ITEM_INT_ML_WS.GET_ITEM_SUPCTRY_INT_DATA
Application Status Function	NA
Family	ITEM_SUPPLIER_CTRY
Atomic Indicator	N

Table ORCV_WS_ENDPOINT**Item Master**

Configuration	Value
Endpoint ID	MFCS_MERCH_ITEM_UPDATE
Credential Static ID	OAUTH_MFCS_INTEGRATION
URL	<MFCS_MerchIntegrations_URL>/services/items/update
Action	PUT
Request Type	REST

Item Supplier

Configuration	Value
Endpoint ID	MFCS_MERCH_SUP_UPDATE
Credential Static ID	OAUTH_MFCS_INTEGRATION
URL	<MFCS_MerchIntegrations_URL>/services/item/supplier/update
Action	PUT
Request Type	REST

Item Supplier Country

Configuration	Value
Endpoint ID	MFCS_MERCH_SUPCTRY_UPDATE
Credential Static ID	OAUTH_MFCS_INTEGRATION
URL	<MFCS_MerchIntegrations_URL>/services/item/suppliers/countries/update

Configuration	Value
Action	PUT
Request Type	REST

Item Rest Publish Asynchronous Batch

This batch processes all entries to be integrated in MFQueue tables for Rest Integrations, it requires the parameter ENTITY to be passed defining the entity to be processed:

- ITEM_MASTER for Item Master CFAs
- ITEM_SUPPLIER for Item Supplier CFAs
- ITEM_SUPPLIER_CTRY for Item Supplier Country CFAs

For those entries to be integrated, this batch enriches the data from the Application Layer generating the payload to call the MFCS MerchIntegrations Rest update endpoints in mode Atomic / Non Atomic depending on configuration setup.

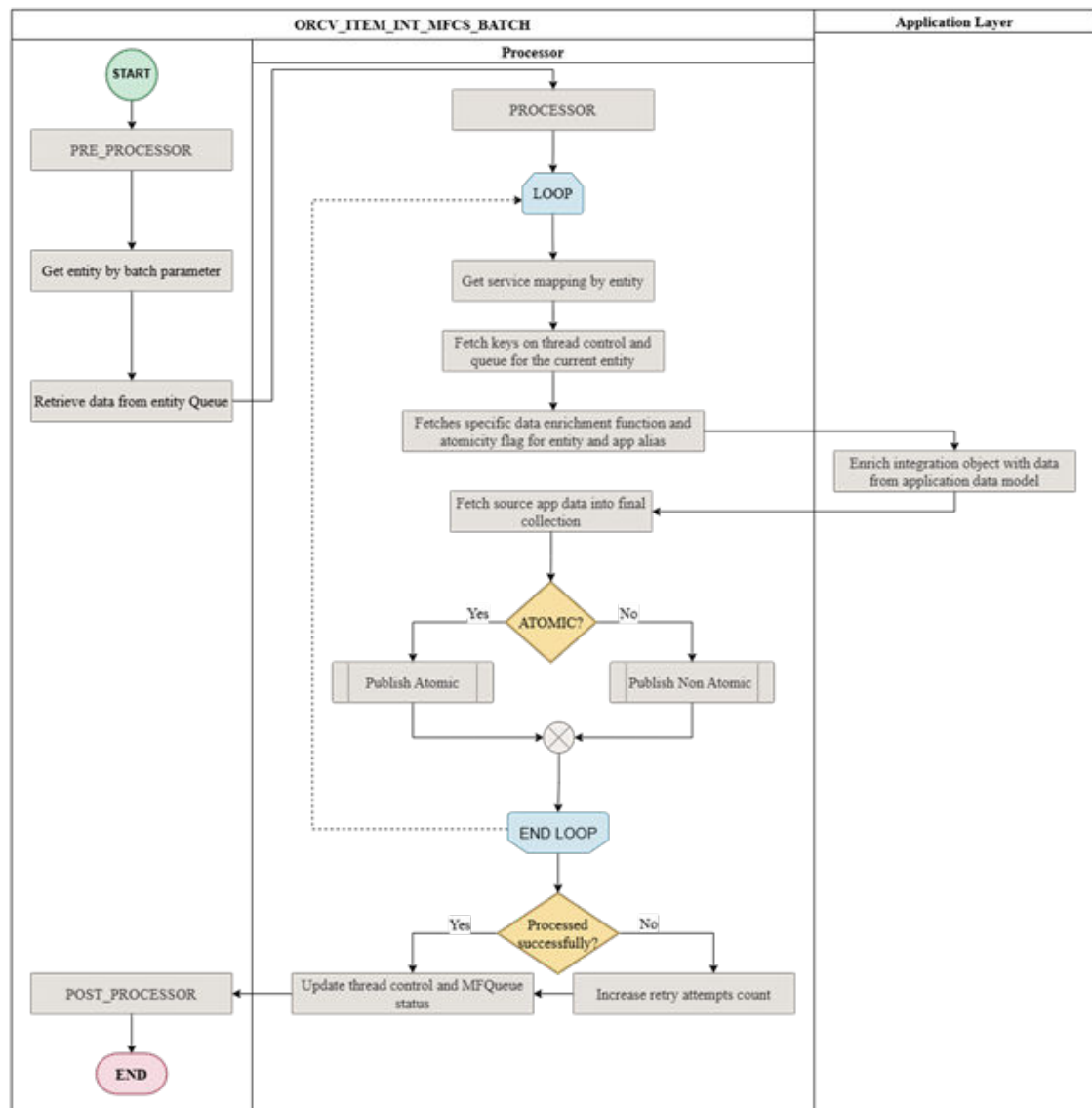
Configuration

Batch configuration table: ORCV_BATCH_CONFIG

Column Name	Default Values
PROGRAM_NAME	ORCV_ITEM_INT_MFCS_JOB
MAX_CHUNK_SIZE	10000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	4
RETRY_WAIT_TIME	2
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Process Overview

Figure 27-79 Item CFAs Rest to MFCS Publish Batch Overview



Operations

Authentication Method: OAuth2

Base URL: /ords/mfcs/orcv/merch/item/integration/item/merchint/batch/<URL Path>

URL Path	Request Method	Request Body	Affected Tables
start	POST	<pre>{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_ITEM_INT_MFCS_JOB", "agentExecutionId": "", "parameters": "ENTITY=ITEM_MASTER ITEM_SUPPLIER ITEM_SUPPLIER_CTRY" }</pre>	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_ITEM_INT_SUPCTRY_CFA ORCV_ITEM_INT_SUPCTRY_SVC ORCV_ITEM_INT_SUPCTRY ORCV_ITEM_INT_SUP_CTRYLOC ORCV_ITEM_INT_SUPCTRY_DIM ORCV_ITEM_INT_MFQUEUE ORCV_ITEM_INT_MASTER_SVC ORCV_ITEM_INT_MASTER ORCV_ITEM_INT_MASTER_CFA_SVC ORCV_ITEM_INT_MASTER_CFA ORCV_ITEM_INT_SUP_SVC ORCV_ITEM_INT_SUP ORCV_ITEM_INT_SUPPLIER_CFA_SVC ORCV_ITEM_INT_SUPPLIER_CFA ORCV_ITEM_INT_SUPCTRY_CFA_SVC
restart	POST	<pre>{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "<EXEC_ID>", "jobName": "ORCV_ITEM_INT_MFCS_JOB", "agentExecutionId": "", "parameters": "ENTITY=ITEM_MASTER ITEM_SUPPLIER ITEM_SUPPLIER_CTRY" }</pre>	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_ITEM_INT_SUPCTRY_CFA ORCV_ITEM_INT_SUPCTRY_SVC ORCV_ITEM_INT_SUPCTRY ORCV_ITEM_INT_SUP_CTRYLOC ORCV_ITEM_INT_SUPCTRY_DIM ORCV_ITEM_INT_MFQUEUE ORCV_ITEM_INT_MASTER_SVC ORCV_ITEM_INT_MASTER ORCV_ITEM_INT_MASTER_CFA_SVC ORCV_ITEM_INT_MASTER_CFA ORCV_ITEM_INT_SUP_SVC ORCV_ITEM_INT_SUP ORCV_ITEM_INT_SUPPLIER_CFA_SVC ORCV_ITEM_INT_SUPPLIER_CFA ORCV_ITEM_INT_SUPCTRY_CFA_SVC

URL Path	Request Method	Request Body	Affected Tables
log	GET	URL query parameters: ? jobName=ORCV_ITEM_INT_MFCS_JOB& executionId=<EXEC_ID>& logDate=YYYY-DD-MM	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "<EXEC_ID>", "jobName": "ORCV_ITEM_INT_MFCS_JOB", "agentExecutionId": "", "parameters": "" }	N/A
status	GET	URL query parameters: ? jobName=ORCV_ITEM_INT_MFCS_JOB& executionId=<EXEC_ID>& logDate=YYYY-DD-MM	N/A

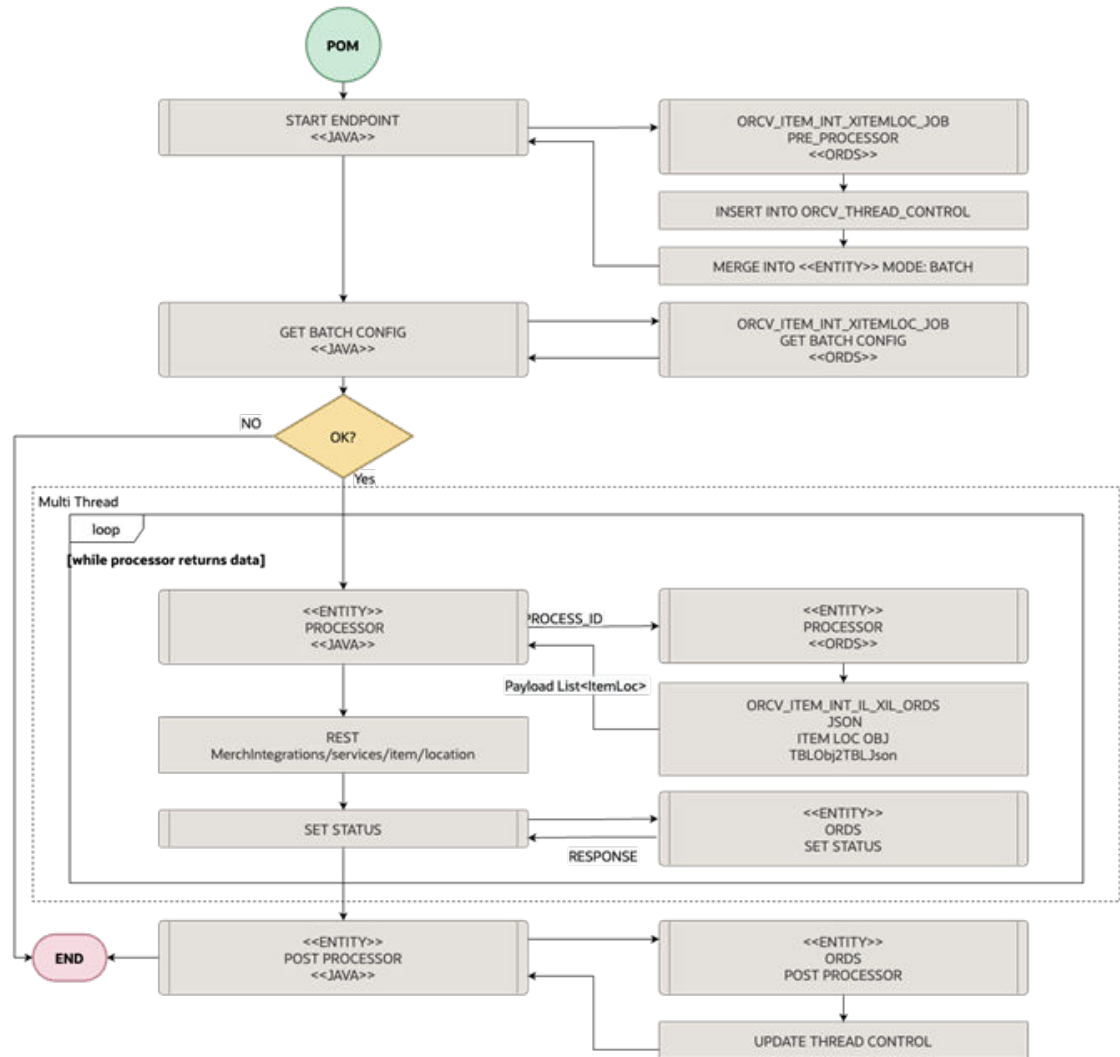
Item Location Integration to MFCS

This integration has a maximum chunk size configuration and has a thread aggregator per number of different hierarchy level records per item. The chunk size represents the maximum number of hierarchy level that each thread iteration will process per item and send to MFCS MerchIntegrations item location service that subscribes items and their locations.

Process Overview

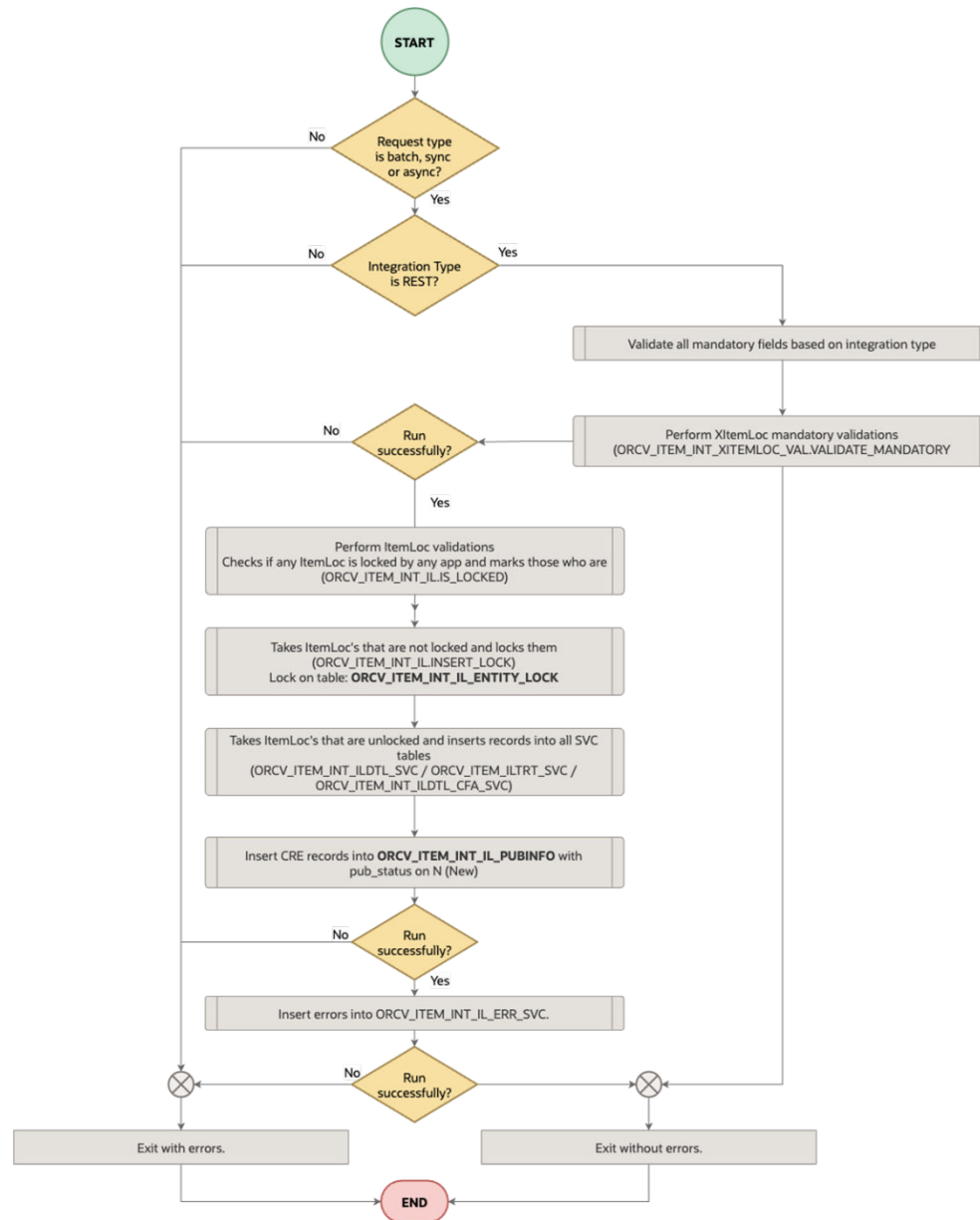
The Item Location integration service will support the integration of item locations into MFCS through the MerchIntegrations ReST endpoint.

Figure 27-80 Item Location Integration to MFCS Process Overview



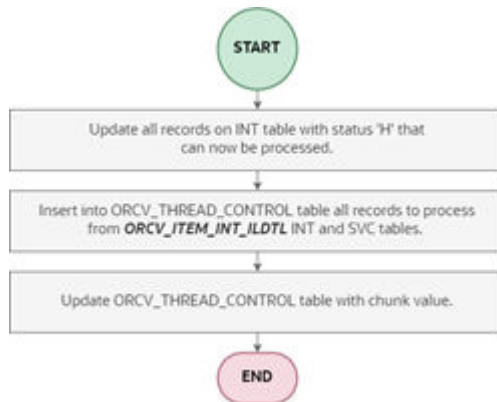
Process Item Loc

Figure 27-81 Item Location Process Overview



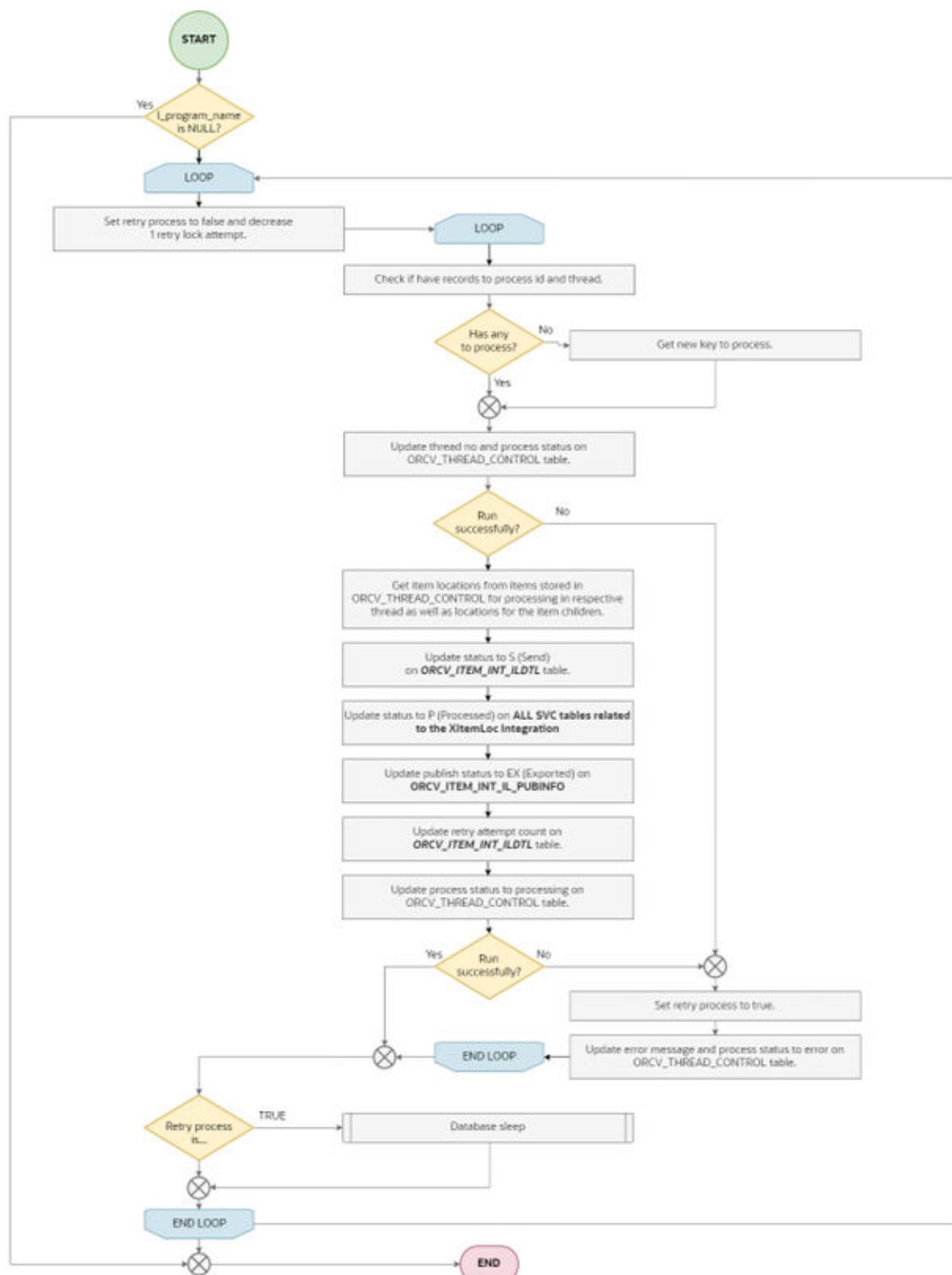
Pre-Processor

Figure 27-82 Pre-Processor Item Location MFCS Integration



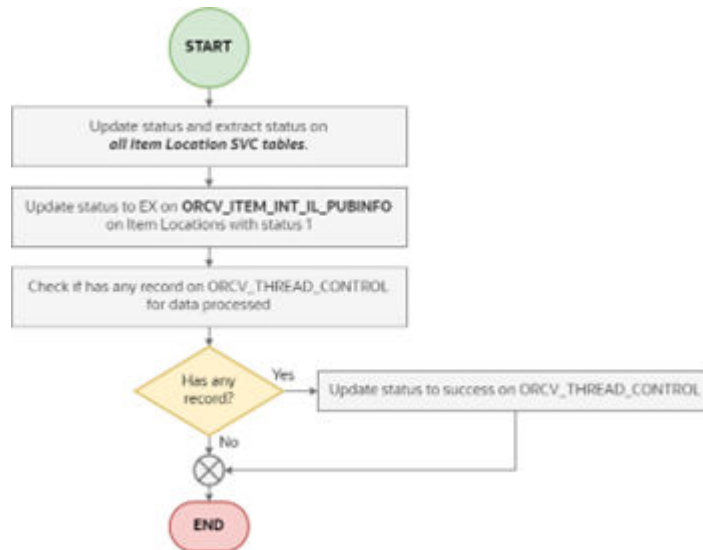
Processor

Figure 27-83 Processor Item Location MFCS Integration



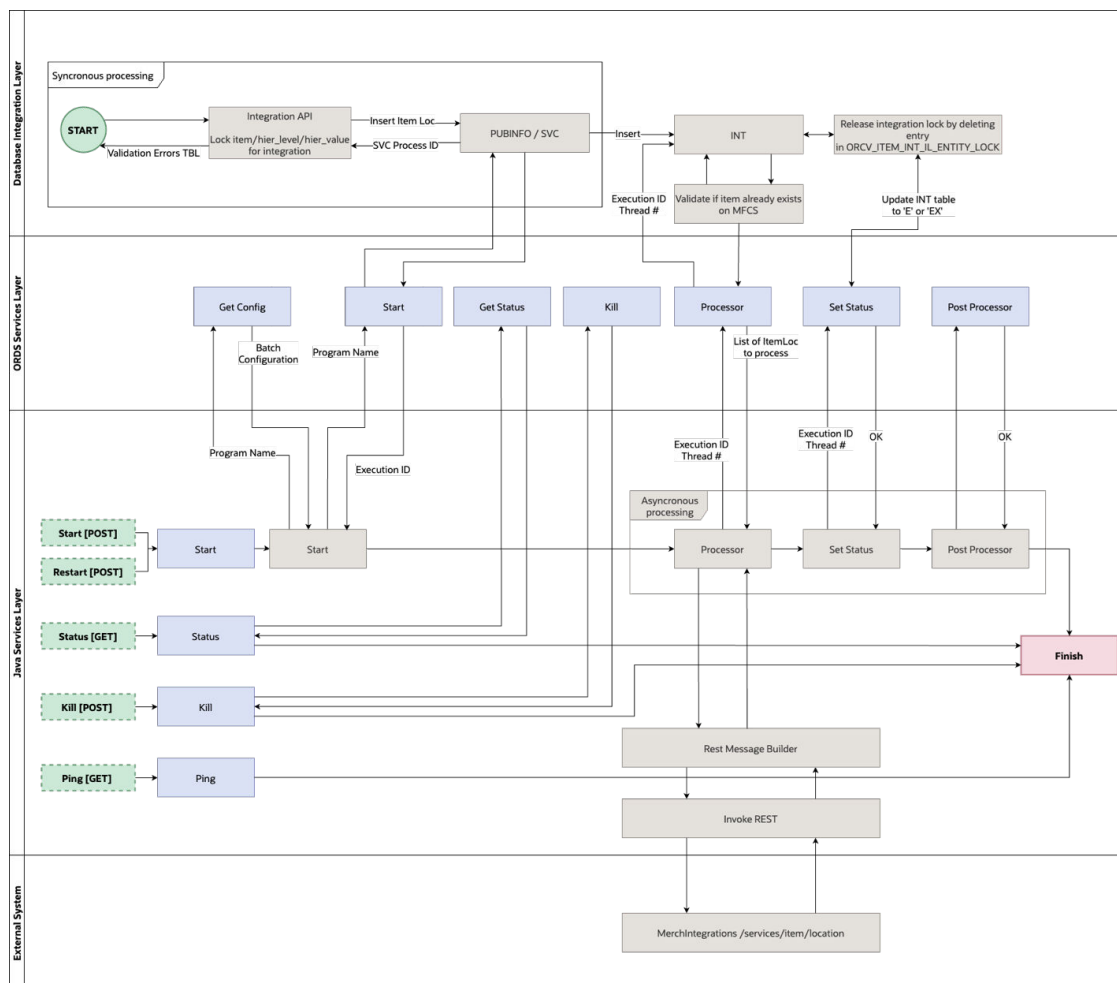
Set Integration Status

Figure 27-84 Set Integration Status Flow for Item Location MFCS Integration



Batch Flow

Figure 27-85 Item Location MFCS Integration Batch Flow



Configuration

This section describes configurations.

Initial Batch Configuration for ORCV_ITEM_INT_XITEMLOC_JOB

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_ITEM_INT_XITEMLOC_JOB
MAX_CHUNK_SIZE	10
MAX_CONCURRENT_THREADS	4
BULK_LIMIT	1000
RETRY_LOCK_ATTEMPTS	4

Column Name	Default Values
RETRY_WAIT_TIME	2
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Web Service

Frequency: Ad hoc

Authorization Method: OAuth 2.0

Base URL: <BASE_URL>/XItemLocJavaServicesWeb/services/orcv/merch/item/integration/
xitemloc /<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
batch/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_ITEM_INT_ XITEMLOC_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_ITEM_INT_IL_ERR_SVC ORCV_ITEM_INT_ILDTL_SVC ORCV_ITEM_INT_ILDTL ORCV_ITEM_INT_ILTRT_SVC ORCV_ITEM_INT_ILTRT ORCV_ITEM_INT_ILDTL_CFA_SVC ORCV_ITEM_INT_ILDTL_CFA ORCV_ITEM_INT_IL_PUBINFO
batch/restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "<EXEC_ID>", "jobName": "ORCV_ITEM_INT_ XITEMLOC_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_ITEM_INT_IL_ERR_SVC ORCV_ITEM_INT_ILDTL_SVC ORCV_ITEM_INT_ILDTL ORCV_ITEM_INT_ILTRT_SVC ORCV_ITEM_INT_ILTRT ORCV_ITEM_INT_ILDTL_CFA_SVC ORCV_ITEM_INT_ILDTL_CFA ORCV_ITEM_INT_IL_PUBINFO
batch/log	GET	URL query parameters: /log?jobName=ORCV_ITEM_INT_ XITEMLOC_JOB &executionId=<EXEC_ID>&date= YYYY-MM-DD	N/A

URL Path	Request Method	Request Body	Affected Tables
batch/kill	POST	{ "executionId": <EXEC_ID>, "jobName": "ORCV_ITEM_INT_ XITEMLOC_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS
batch/status	GET	URL query parameters: /status? jobName=ORCV_ITEM_INT_ XITEMLOC_JOB &executionId=<EXEC_ID>	N/A
/ping	GET	Empty	N/A

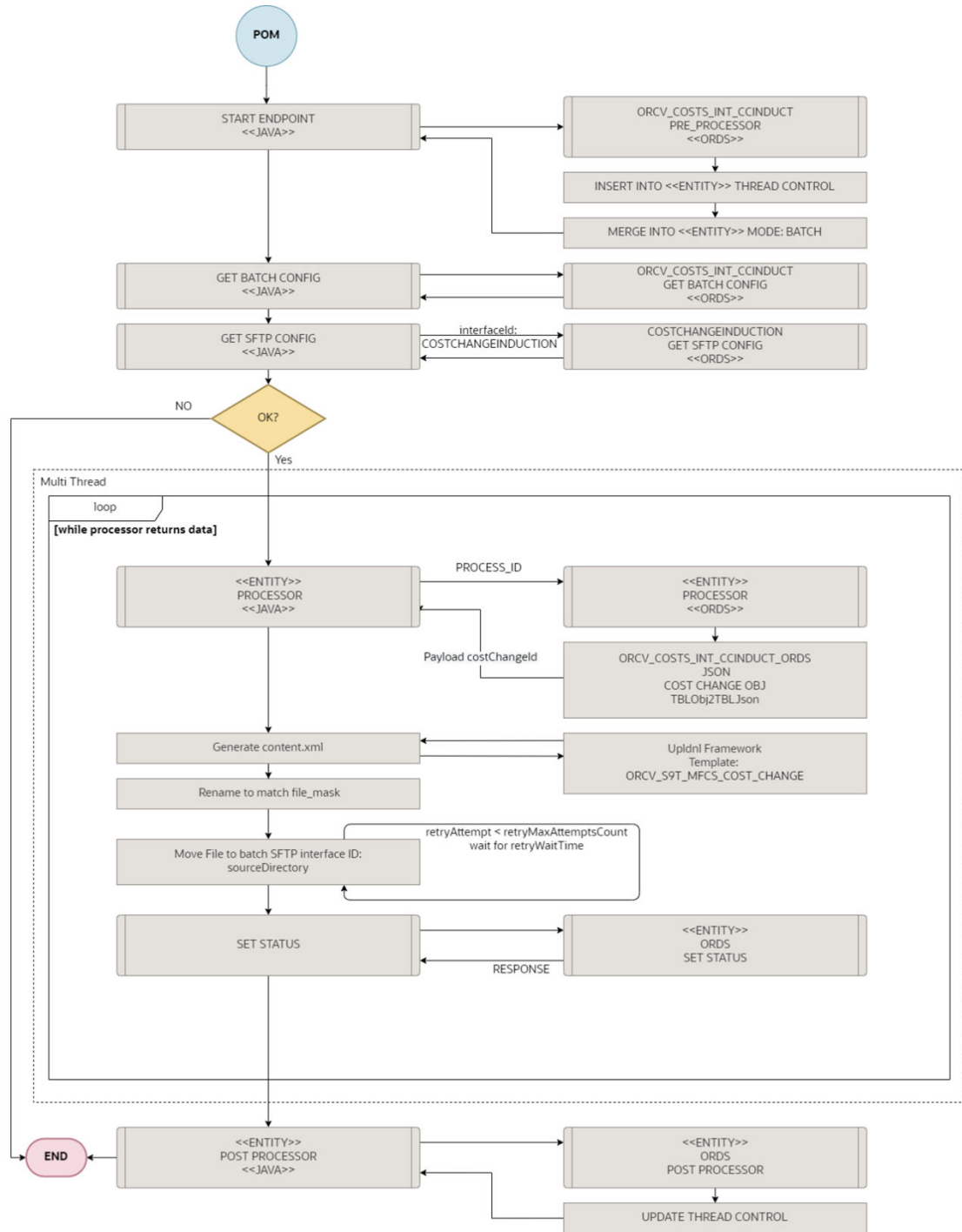
Cost Change Induction to MFCS

This section describes the Cost Change Induction to MFCS.

Process Overview

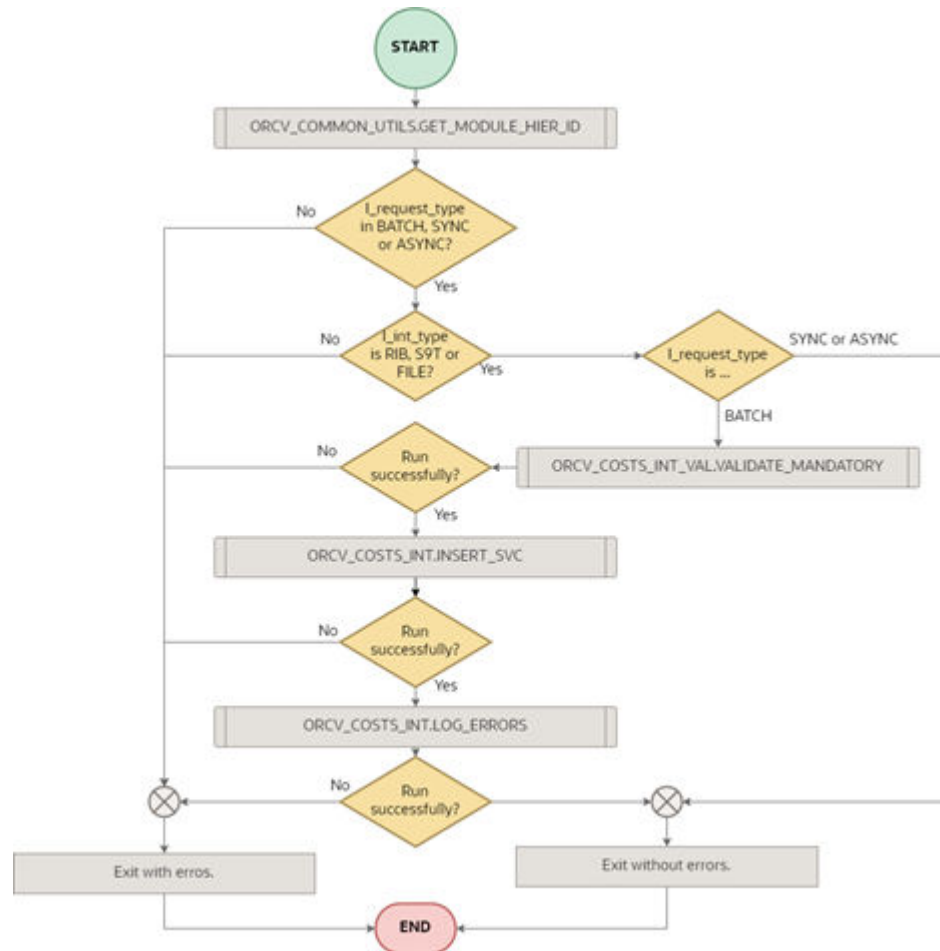
The Cost Change induction service will support the integration of Cost Changes into MFCS through the induction batch.

Figure 27-86 Cost Change Induction to MFCS Process Overview



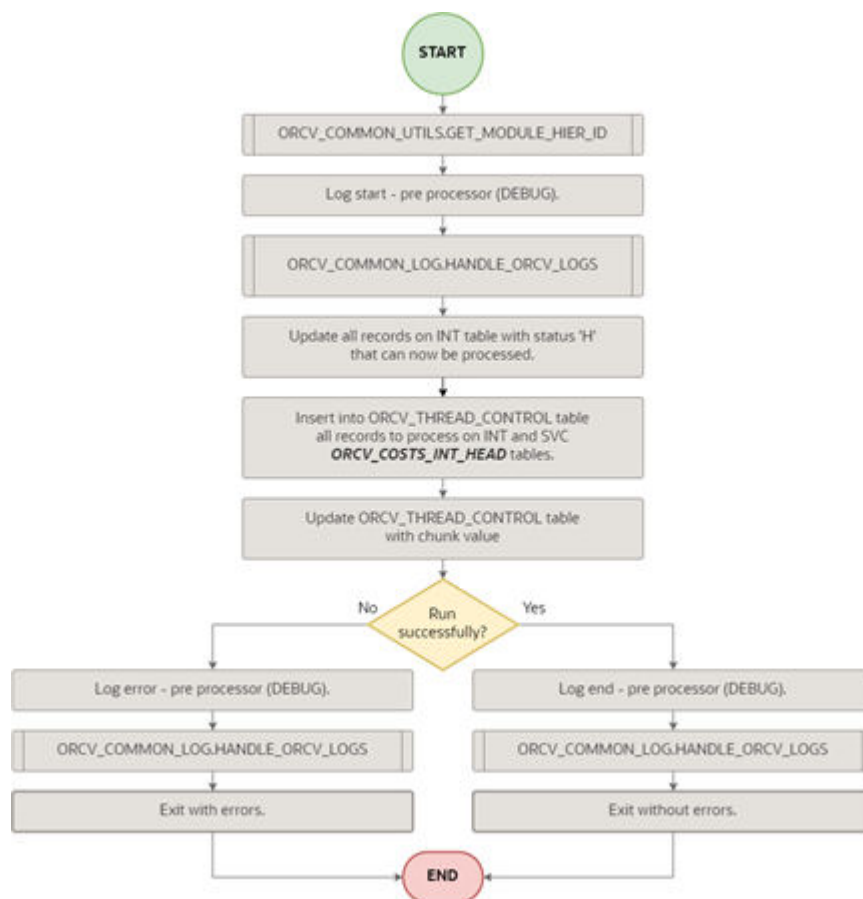
Process Cost Change

Figure 27-87 Cost Change Process Overview



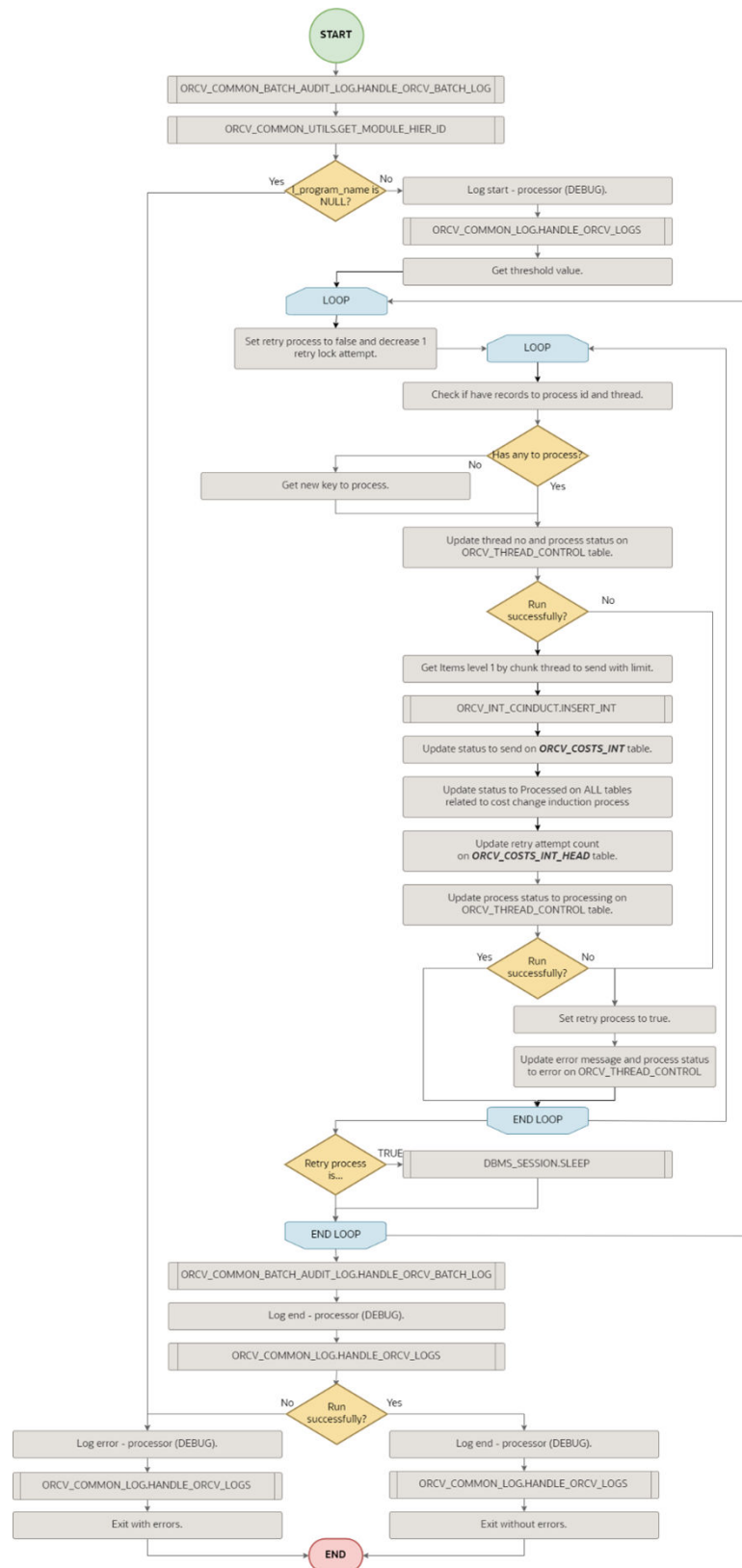
Pre-Processor

Figure 27-88 Pre-Processor Cost Change Induction to MFCS Integration



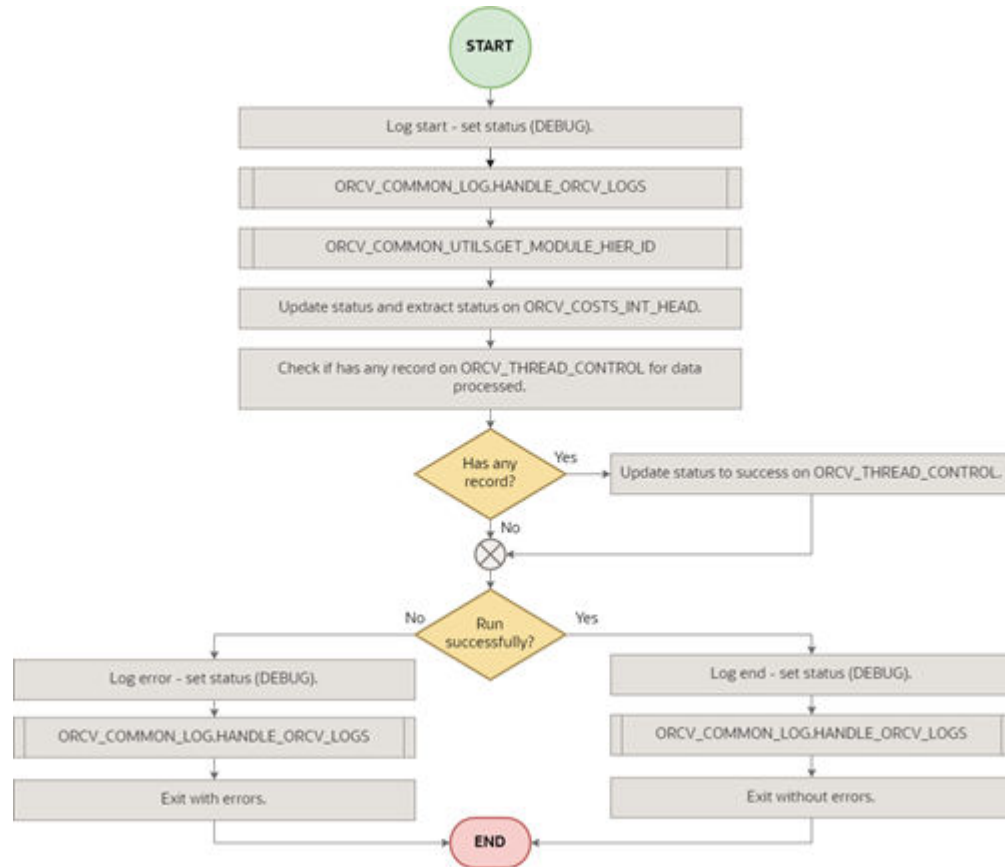
Processor

Figure 27-89 Processor Cost Change Induction to MFCS Integration



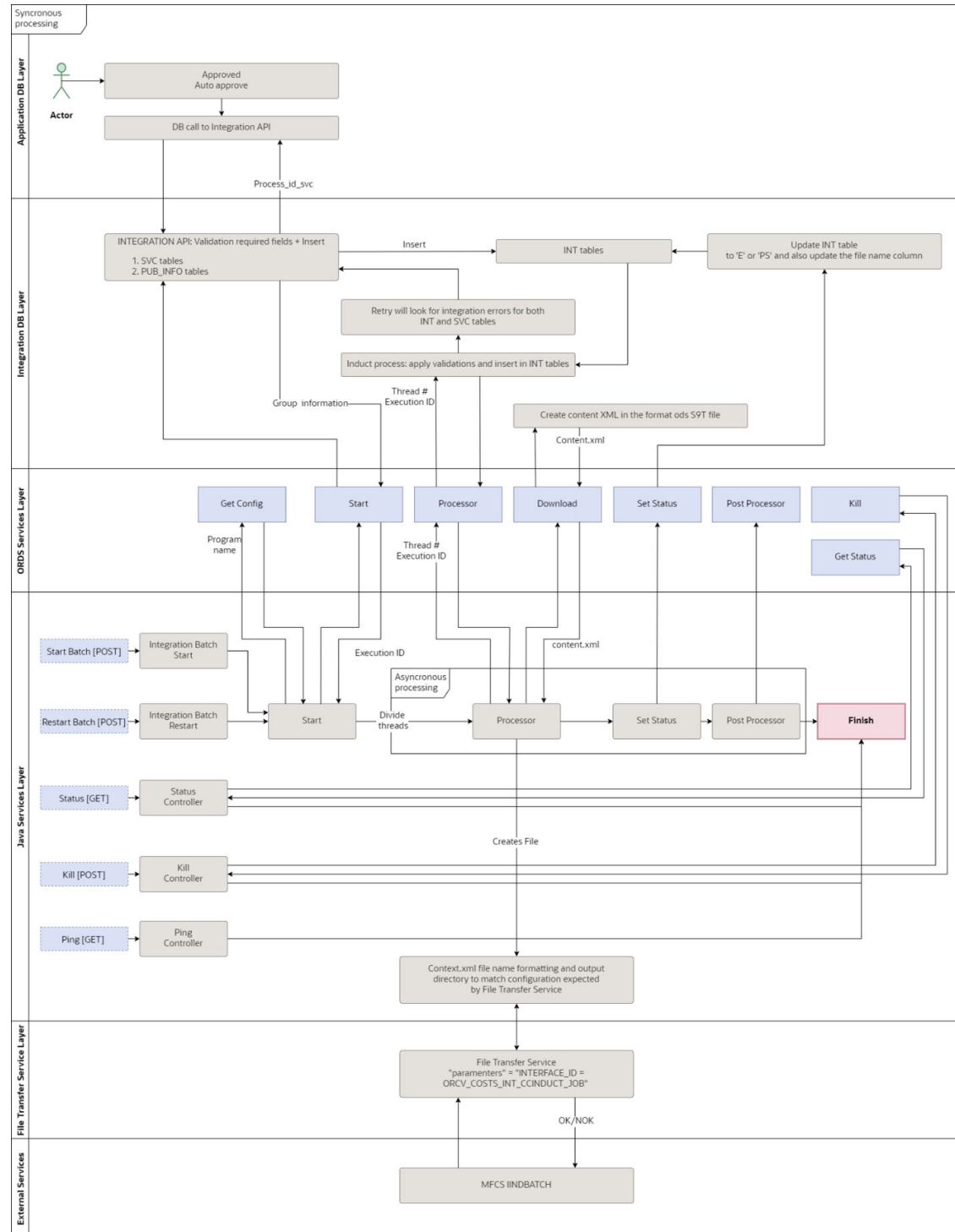
Set Integration Status

Figure 27-90 Set Integration Status Flow for Cost Change Induction to MFCS Integration



Batch Flow

Figure 27-91 Processor Cost Change Induction to MFCS Integration Batch Flow



Configuration

The Cost Change Induction integration service has a prerequisite of the interface with ID COSTCHANGEINDUCTION being configured in ORCV_BATCH_SFTP and ORCV_BATCH_SFTP_TARGET tables.



Note:

POM configuration for MFCS IINDBATCH_CC_JOB is set to run at a high frequency if required and the final target is Merchandising instead of staging tables. The cost change number should be aligned to avoid collision in Merchandising.

Initial Batch Configuration for ORCV_COSTS_INT_CCINDUCT_JOB

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_COSTS_INT_CCINDUCT_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
BULK_LIMIT	1000
RETRY_LOCK_ATTEMPTS	4
RETRY_WAIT_TIME	2
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

System Parameters

Listed below is the list of configurations at system level that are required by this integration.

The parameters are defined in table ORCV_SYSTEM_PARAMETERS.

Area	Parameter	Description	Value 1
INTEGRATION	INDUCTION_TEMPLATE	Template key.	ORCV_S9T_MFCS_COST_CHANGE

Web Service

Frequency: Ad hoc

Base URL: <BASE_URL>/CostChangeInductionJavaServicesWeb/services/orcv/merch/costs/integration/

costchangeinduct/<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
batch/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_COSTS_IN T_ CCINDUCT_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_COSTS_INT_HEAD_SVC ORCV_COSTS_INT_DETAIL_SVC ORCV_COSTS_INT_DETAIL_LOC_SVC ORCV_COSTS_INT_HEAD ORCV_COSTS_INT_DETAIL ORCV_COSTS_INT_DETAIL_LOC ORCV_COSTS_INT_CC_HEAD_PUBIN FO ORCV_COSTS_INT_CC_DTL_PUBINF O ORCV_COSTS_INT_CC_LOC_PUBINF O ORCV_COSTS_INT_ERR_SVC
batch/restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_COSTS_IN T_ CCINDUCT_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_COSTS_INT_HEAD_SVC ORCV_COSTS_INT_DETAIL_SVC ORCV_COSTS_INT_DETAIL_LOC_SVC ORCV_COSTS_INT_HEAD ORCV_COSTS_INT_DETAIL ORCV_COSTS_INT_DETAIL_LOC ORCV_COSTS_INT_CC_HEAD_PUBIN FO ORCV_COSTS_INT_CC_DTL_PUBINF O ORCV_COSTS_INT_CC_LOC_PUBINF O ORCV_COSTS_INT_ERR_SVC
batch/log	GET	URL query parameters: /log? jobName=ORCV_COSTS_INT_ CCINDUCT_JOB& executionId=<EXEC_ID>&date= YYYY-MM-DD	N/A
batch/kill	POST	{ "executionId": <EXEC_ID>, "jobName": "ORCV_COSTS_IN T_ CCINDUCT_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS

URL Path	Request Method	Request Body	Affected Tables
batch/status	GET	URL query parameters: /status? jobName=ORCV_COSTS_INT_ CCINDUCT_ JOB&executionId=<EXEC_ID >	N/A
ping	GET	Empty	N/A

Return to Vendor to MFCS

Return to Vendor integration to MFCS MerchIntegration extends the Database Integration Framework to Publish RTVs.

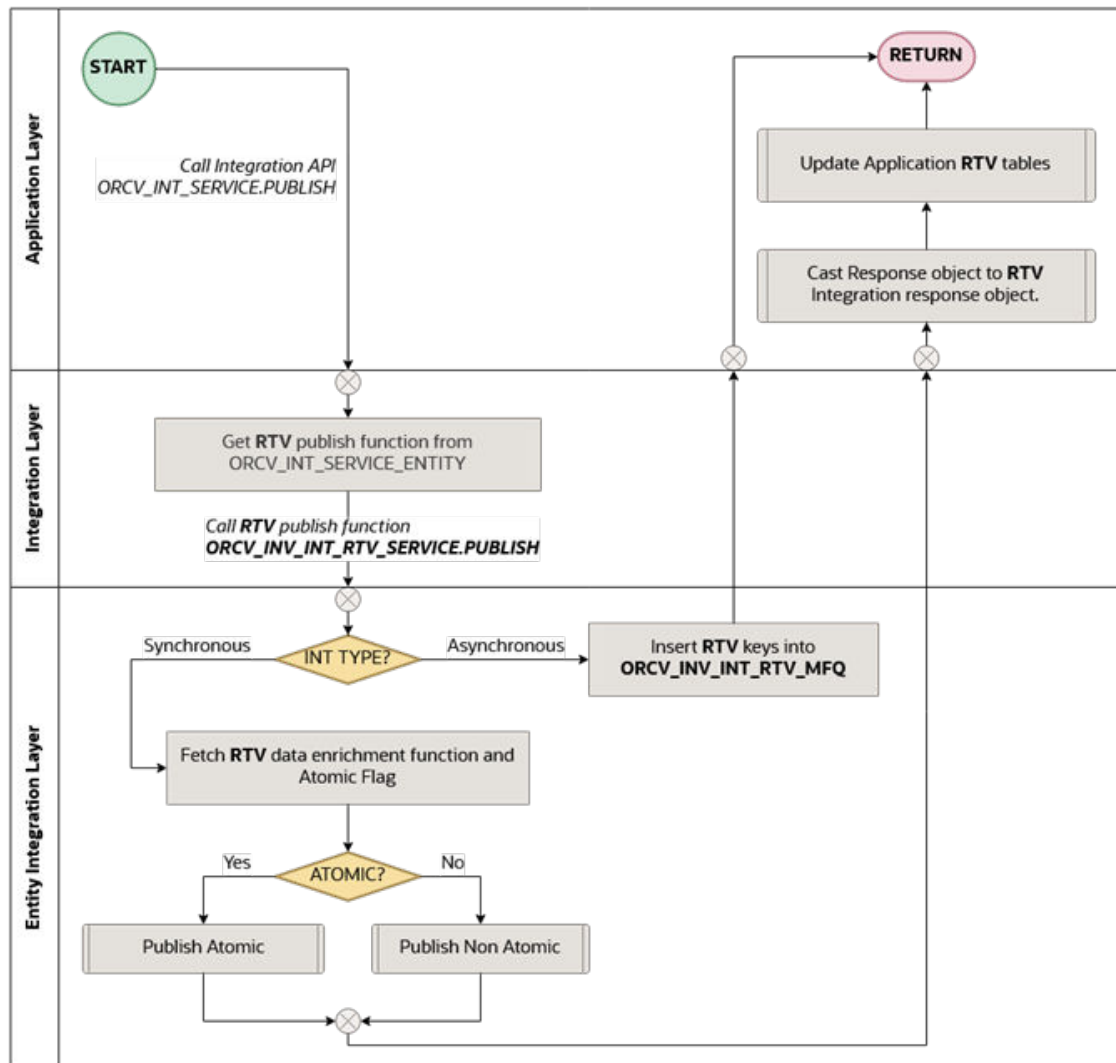
Process Overview

This section describes the process overview.

Publish Flow

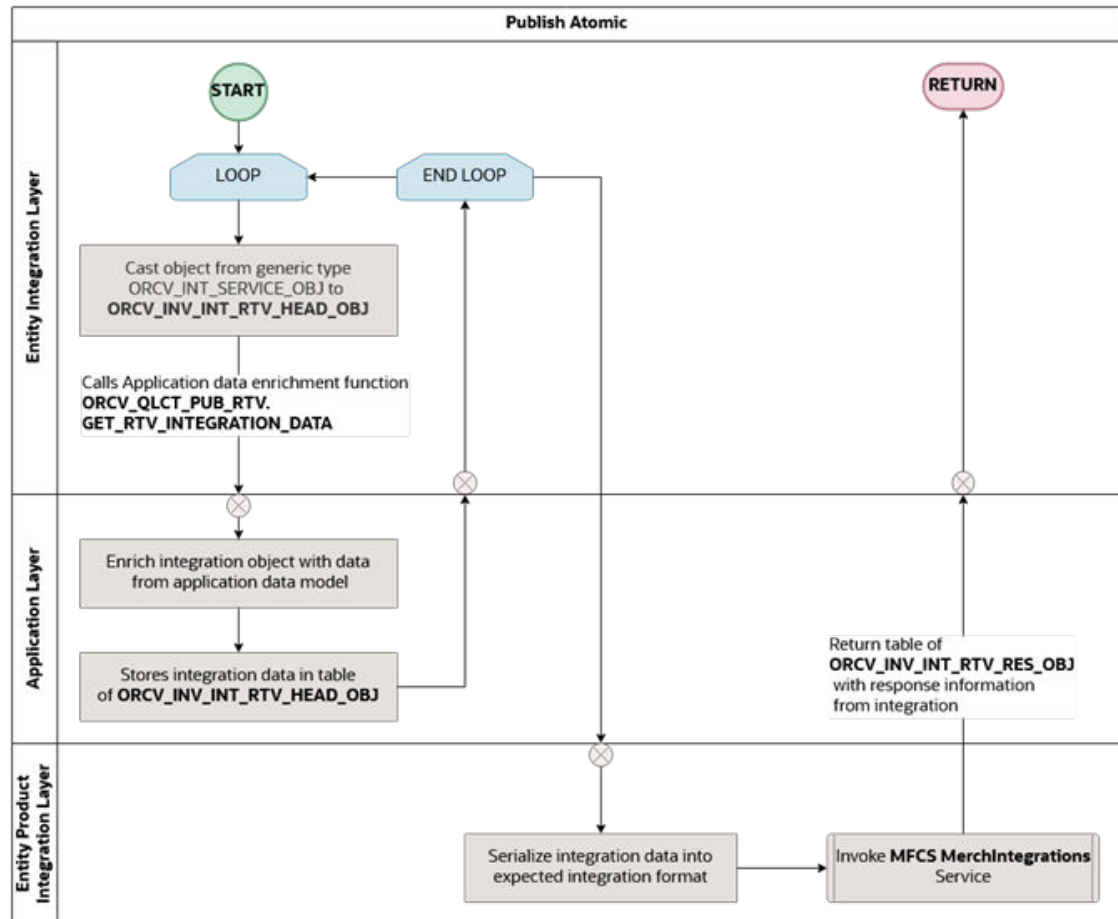
Publish allows the calling of integration endpoints in synchronous mode. Those calls can be either atomic or non-atomic. Atomic is in case all the entries need to be integrated together in bulk, while the non-atomic deals with each entry individually.

Figure 27-92 Return to Vendor to MFCS Publish Flow



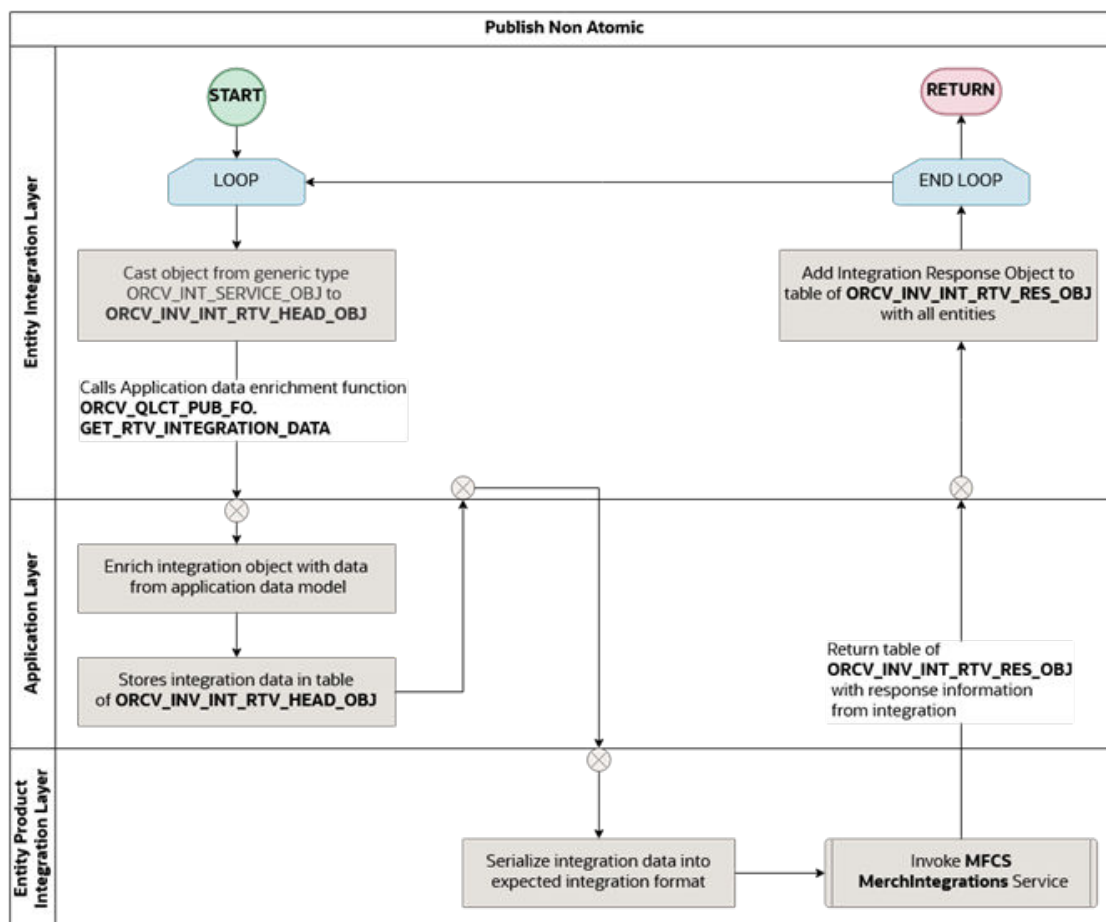
Publish Atomic

Figure 27-93 Return to Vendor to MFCS Publish Atomic



Publish Non Atomic

Figure 27-94 Return to Vendor to MFCS Publish Non Atomic



Configuration

Quality Control and Return to Vendor publish RTVs with the following configurations in Database Integration Framework.

Table ORCV_INT_SERVICE_ENTITY

Configuration	Value
Entity ID	RTV
Publication Function	ORCV_INV_INT_RTV_SERVICE.PUBLISH

Table ORCV_INT_SERVICE_MAPPING

Quality Control

Configuration	Value
Application Alias	QLCT
Application Entity Get Data Function	ORCV_QLCT_PUB_RTV.GET_RTV_INTEGRATION_DATA
Application Status Function	NA
Atomic Indicator	N

Return to Vendor

Configuration	Value
Application Alias	SPRV
Application Entity Get Data Function	ORCV_SPRV_PUB_RTV.GET_RTV_INTEGRATION_DATA
Application Status Function	NA
Atomic Indicator	N

Table ORCV_WS_ENDPOINT

Configuration	Value
Endpoint ID	MFCS_MERCH_RTV_MANAGE
Credential Static ID	OAuth_MFCS_INTEGRATION
URL	<MFCS_MerchIntegrations_URL>/services/inventory/returnsToVendor/manage
Action	PUT
Request Type	REST

Integration Object Type

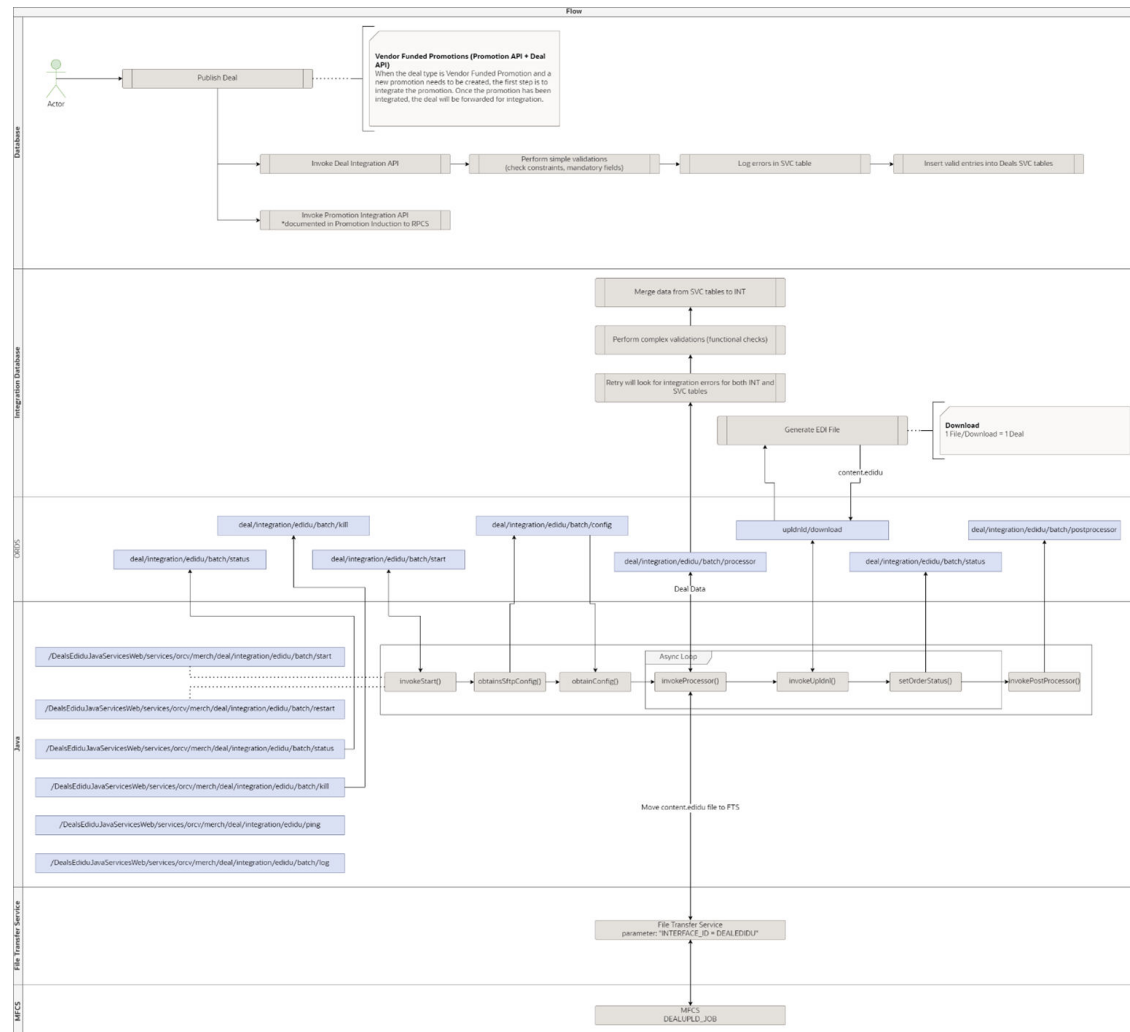
ORCV_INT_SERVICE_OBJ is a base object type for handling the reference keys sent for integration and Return to Vendor to MFCS implements ORCV_INV_INT_RTV_HEAD_OBJ under this object following the RTVs specific requirements.

Deals to MFCS

The Deal to MFCS batch service will support the integration of Deals into MFCS as an EDIDU file through the deal upload batch (DEALUPLD_JOB).

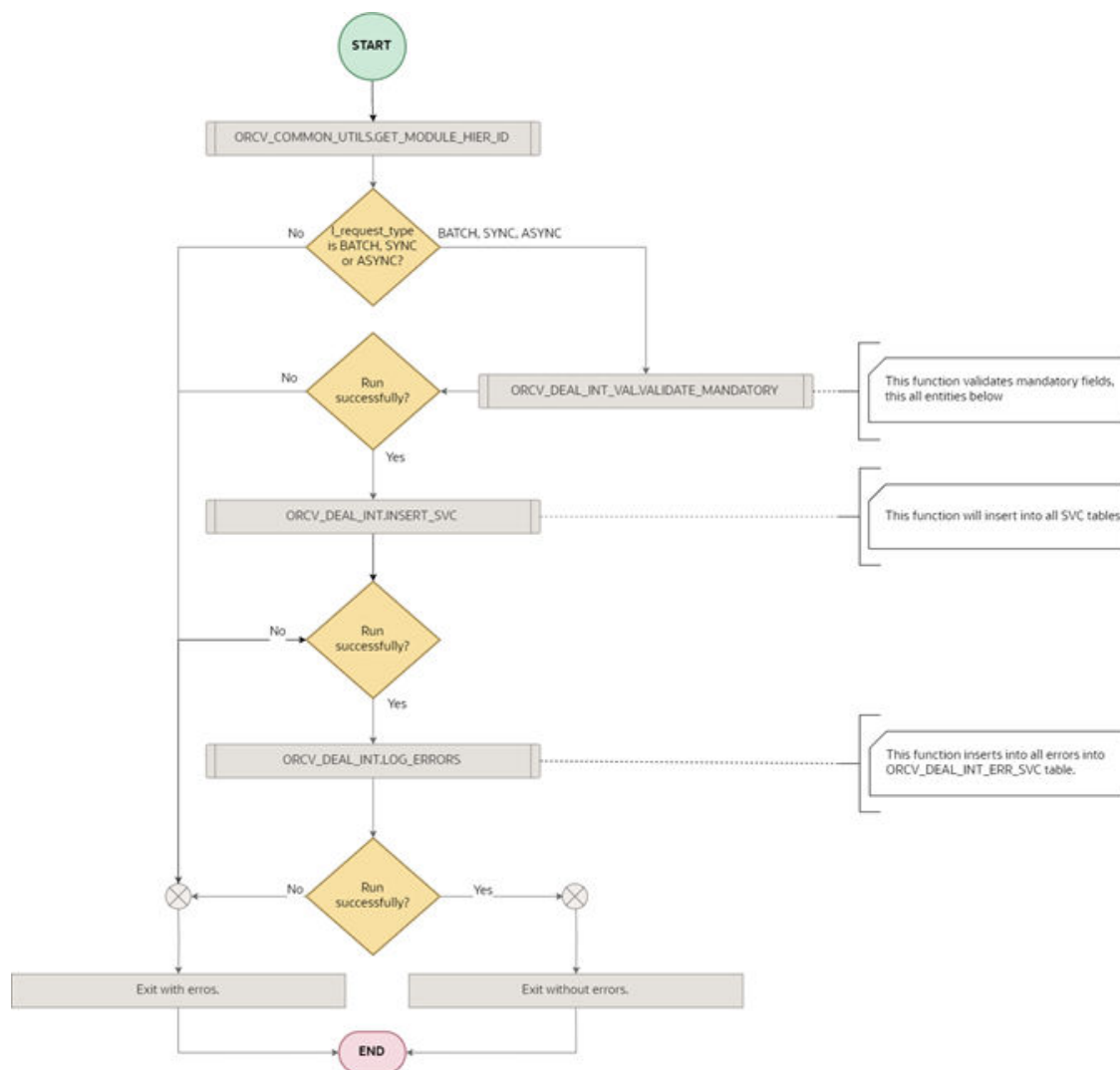
Process Overview

Figure 27-95 Deal Integration to MFCS Batch Process Overview



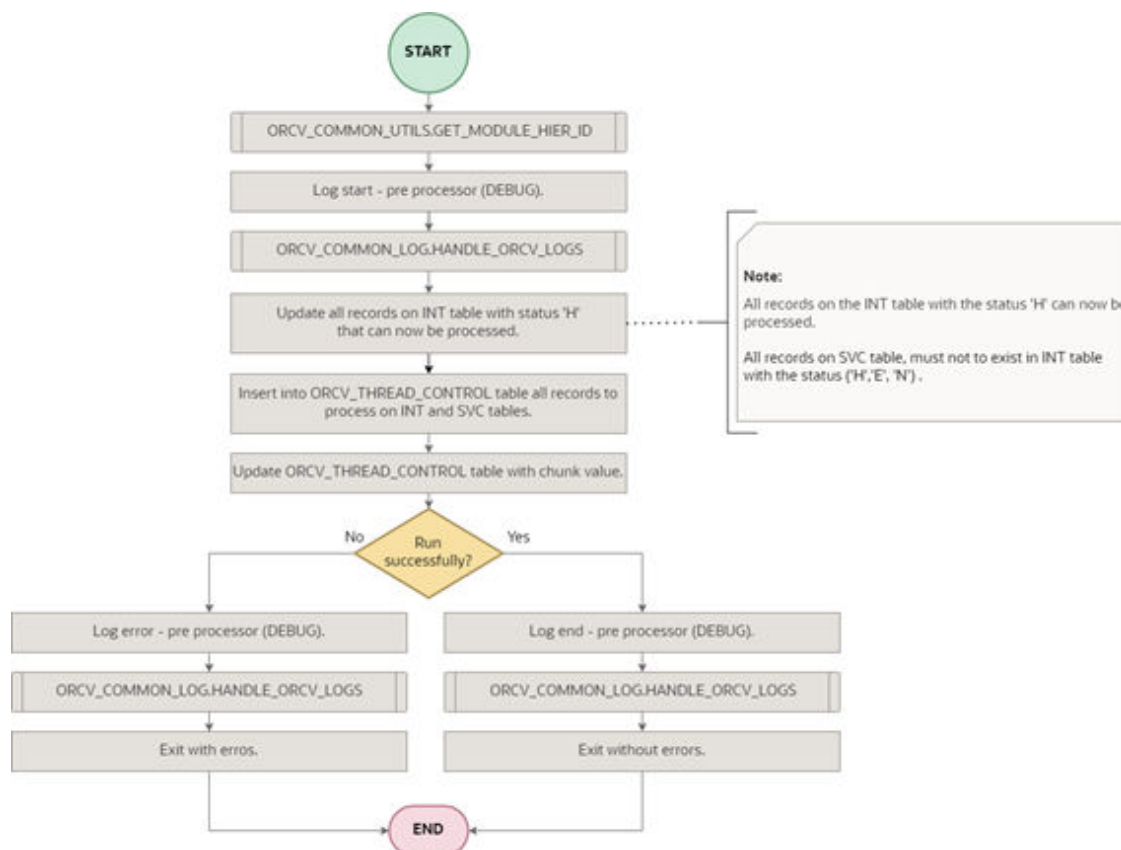
Process Deal

Figure 27-96 Deal Process Overview



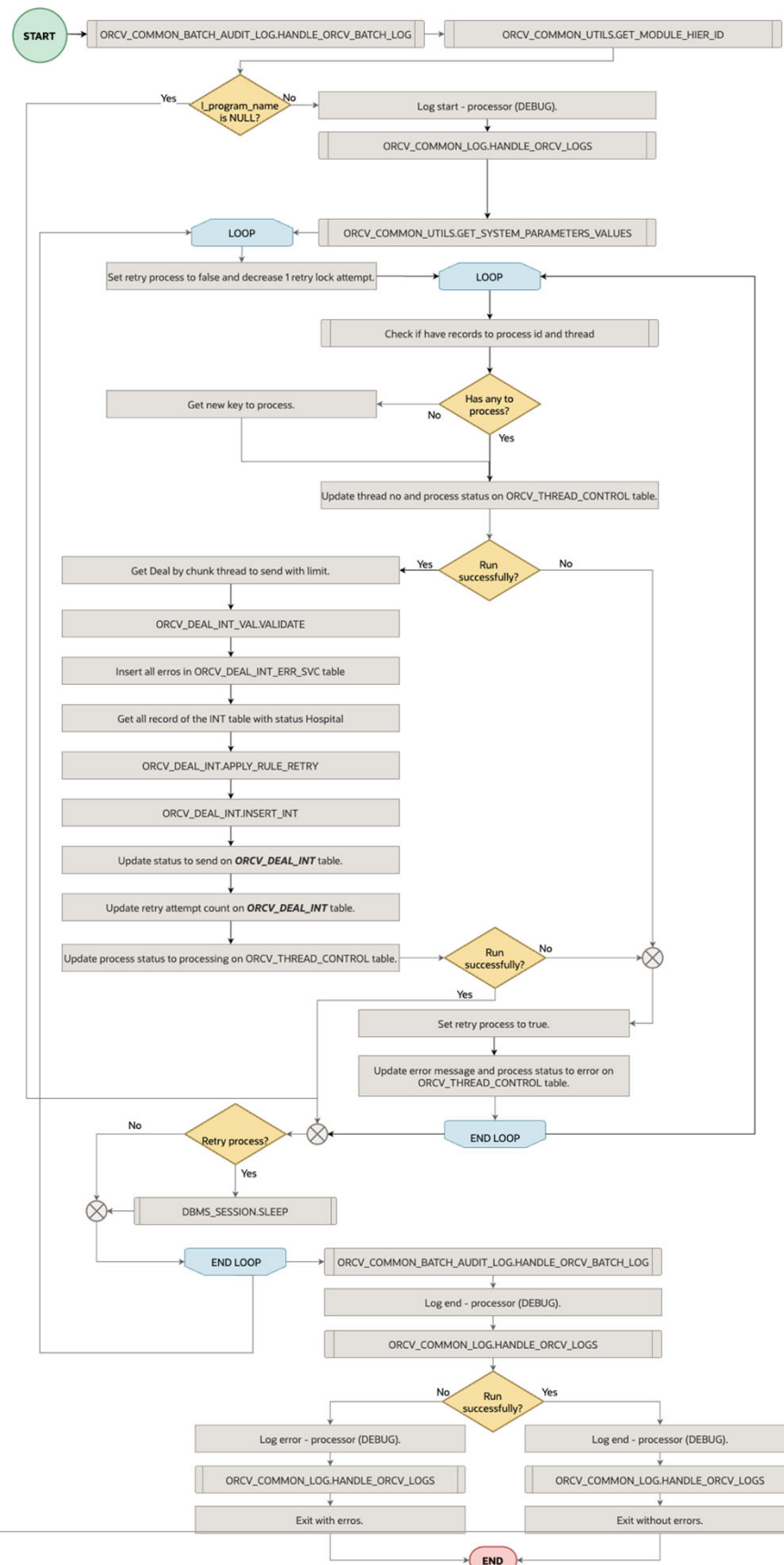
Pre-Processor

Figure 27-97 Pre-Processor Deal Integration to MFCS



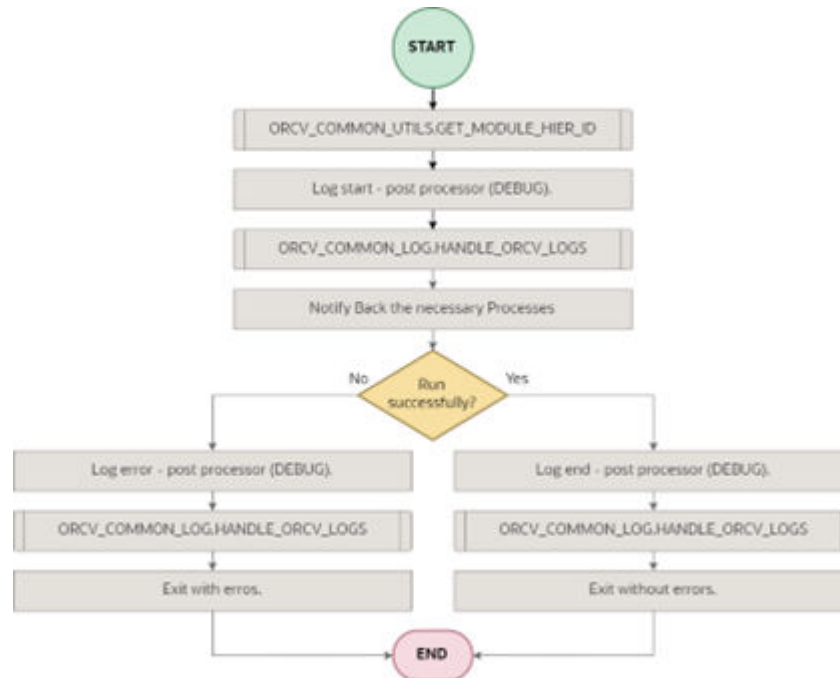
Processor

Figure 27-98 Processor Deal Integration to MFCS



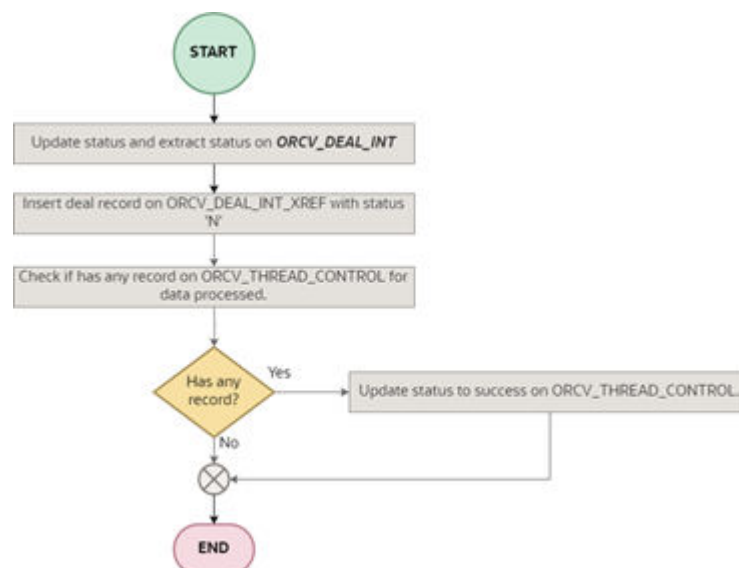
Post Processor

Figure 27-99 Post Processor Deal Integration to MFCS



Set Integration Status

Figure 27-100 Set Integration Status Flow for Deal Integration to MFCS



Configuration

The Deal to MFCS batch service has a prerequisite of the interface ID DEALEIDU being configured on ORCV_BATCH_SFTP and ORCV_BATCH_SFTP_TARGET tables.



Note:

POM configuration for ORCV_DEAL_INT_EDIDU_JOB Ad-hoc execution is advised to be set at a high frequency.

Initial Batch Configuration for ORCV_DEAL_INT_EDIDU_JOB

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_DEAL_INT_EDIDU_JOB
MAX_CHUNK_SIZE	1 (one deal per edidu file)
MAX_CONCURRENT_THREADS	4
BULK_LIMIT	1000
RETRY_LOCK_ATTEMPTS	4
RETRY_WAIT_TIME	2
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

System Parameters

Listed below is the list of configurations at system level that are required by this integration.

The parameters are defined in table ORCV_SYSTEM_PARAMETERS.

Area	Parameter	Description	Value 1
PURGE	PURGE_DAYS	Default purge Period.	30
INTEGRATION	INDUCTION_TEMPLATE	Deals Integration template key	MFCS_EDIDU_DATA_V23
INTEGRATION	THRESHOLD_LOWER_LIMIT	Lower limit of the deal component. This is the minimum value will apply, when multiple_thresholds_ind = 'N'	0
INTEGRATION	THRESHOLD_UPPER_LIMIT	'Upper limit of the deal component. This is the maximum value will apply, when multiple_thresholds_ind = 'N'	99999999

Area	Parameter	Description	Value 1
CONFIG	GENERATE_DEAL_COMPONENT	Apply deal component rule and insert integration tables.	ORCV_DEAL_INT_EDIDU. INSERT_INT

Web Service

Frequency: Ad hoc

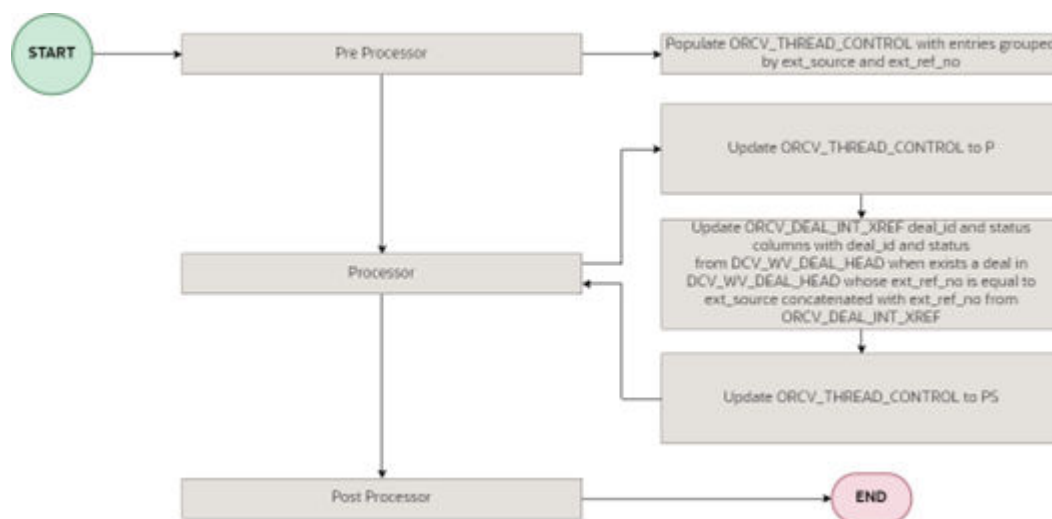
Base URL: <BASE_URL>/DealsEdiduJavaServicesWeb/services/orcv/merch/deal/integration/
edidu/< URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
batch/start	POST	{ "cycleName":""," "flowName":""," "processName":""," "async":""," "executionId":""," "jobName":"ORCV_DEAL_INT_ EDIDU_JOB", "agentExecutionId":""," "parameters":""," }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_ENGINE_THREADS ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG ORCV_DEAL_INT_HEAD_SVC ORCV_DEAL_INT_DETAIL_SVC ORCV_DEAL_INT_COMP_PRM_SV C ORCV_DEAL_INT_IL_ITEM_SVC ORCV_DEAL_INT_POP_TERMS_SV C ORCV_DEAL_INT_THRESHOLD_S VC ORCV_DEAL_INT_ERR_SVC ORCV_DEAL_INT_HEAD ORCV_DEAL_INT_DETAIL ORCV_DEAL_INT_COMP_PRM ORCV_DEAL_INT_IL_ITEM ORCV_DEAL_INT_POP_TERMS ORCV_DEAL_INT_THRESHOLD ORCV_DEAL_INT_XREF ORCV_DEAL_INT_HEAD_PUBINF O

URL Path	Request Method	Request Body	Affected Tables
batch/ restart	POST	{ "cycleName":""," "flowName":""," "processName":""," "async":""," "executionId":<EXEC_ID>, "jobName":"ORCV_DEAL_INT_ EDIDU_JOB", "agentExecutionId":""," "parameters":""," }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_ENGINE_THREADS ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG ORCV_DEAL_INT_HEAD_SVC ORCV_DEAL_INT_DETAIL_SVC ORCV_DEAL_INT_COMP_PRM_SV C ORCV_DEAL_INT_IL_ITEM_SVC ORCV_DEAL_INT_POP_TERMS_SV C ORCV_DEAL_INT_THRESHOLD_S VC ORCV_DEAL_INT_ERR_SVC ORCV_DEAL_INT_HEAD ORCV_DEAL_INT_DETAIL ORCV_DEAL_INT_COMP_PRM ORCV_DEAL_INT_IL_ITEM ORCV_DEAL_INT_POP_TERMS ORCV_DEAL_INT_THRESHOLD ORCV_DEAL_INT_XREF ORCV_DEAL_INT_HEAD_PUBINF O
batch/log	GET	URL query parameters: /log?jobName=ORCV_DEAL_INT_ EDIDU_JOB& executionId=<EXEC_ID>&date= YYYY-MM-DD	N/A
batch/kill	POST	{ "executionId": <EXEC_ID>, "jobName":"ORCV_DEAL_INT_ EDIDU_JOB" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG
batch/ status	GET	URL query parameters: /status? jobName=ORCV_DEAL_INT_ EDIDU_JOB&executionId=<EXEC_ ID>	N/A
ping	GET	Empty	N/A

Deal XREF Batch Service

The Deal cross reference batch service allow synchronization of MFCS deal ID to the external reference integrated through the Deal Integration Batch Service.

Figure 27-101 Deal XREF Batch Service Overview

Initial Batch Configuration for ORCV_DEAL_INT_EDIDU_XREF_JOB

Batch default configuration (ORCV_BATCH_CONFIG):

Column Name	Default Values
PROGRAM_NAME	ORCV_DEAL_INT_EDIDU_XREF_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_LOCK_ATTEMPTS	4
RETRY_WAIT_TIME	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_IND	N

Web Service

Frequency: Ad-hoc

Base URL: <BASE_URL>/ords/mfcs/orcv/merch/deal/integration/xrefupdate/< URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
batch/start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_DEAL_INT_ EDIDU_XREF_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTR OL ORCV_THREAD_CONTROL ORCV_BATCH_ENGINE_THRE ADS ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG ORCV_DEAL_INT_XREF
batch/ restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": <EXEC_ID>, "jobName": "ORCV_DEAL_INT_ EDIDU_XREF_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTR OL ORCV_THREAD_CONTROL ORCV_BATCH_ENGINE_THRE ADS ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG ORCV_DEAL_INT_XREF
batch/log	GET	URL query parameters: /log?jobName=ORCV_DEAL_INT_ EDIDU_XREF_JOB& executionId=<EXEC_ID>&logDate= YYYY-MM-DD	N/A
batch/kill	POST	{ "executionId": <EXEC_ID>, "jobName": "ORCV_DEAL_INT_ EDIDU_XREF_JOB" }	ORCV_BATCH_STATUS_CONTR OL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_BATCH_AUDIT_LOG
batch/ status	GET	URL query parameters: /status?jobName=ORCV_DEAL_INT_ EDIDU_XREF_JOB&executionId=<EXE C_ID>	N/A
ping	GET	Empty	N/A

Fulfillment Order to MFCS

Fulfillment Order to MFCS extends the Database Integration Framework to publish Fulfillment Orders to MFCS Merch Integrations.

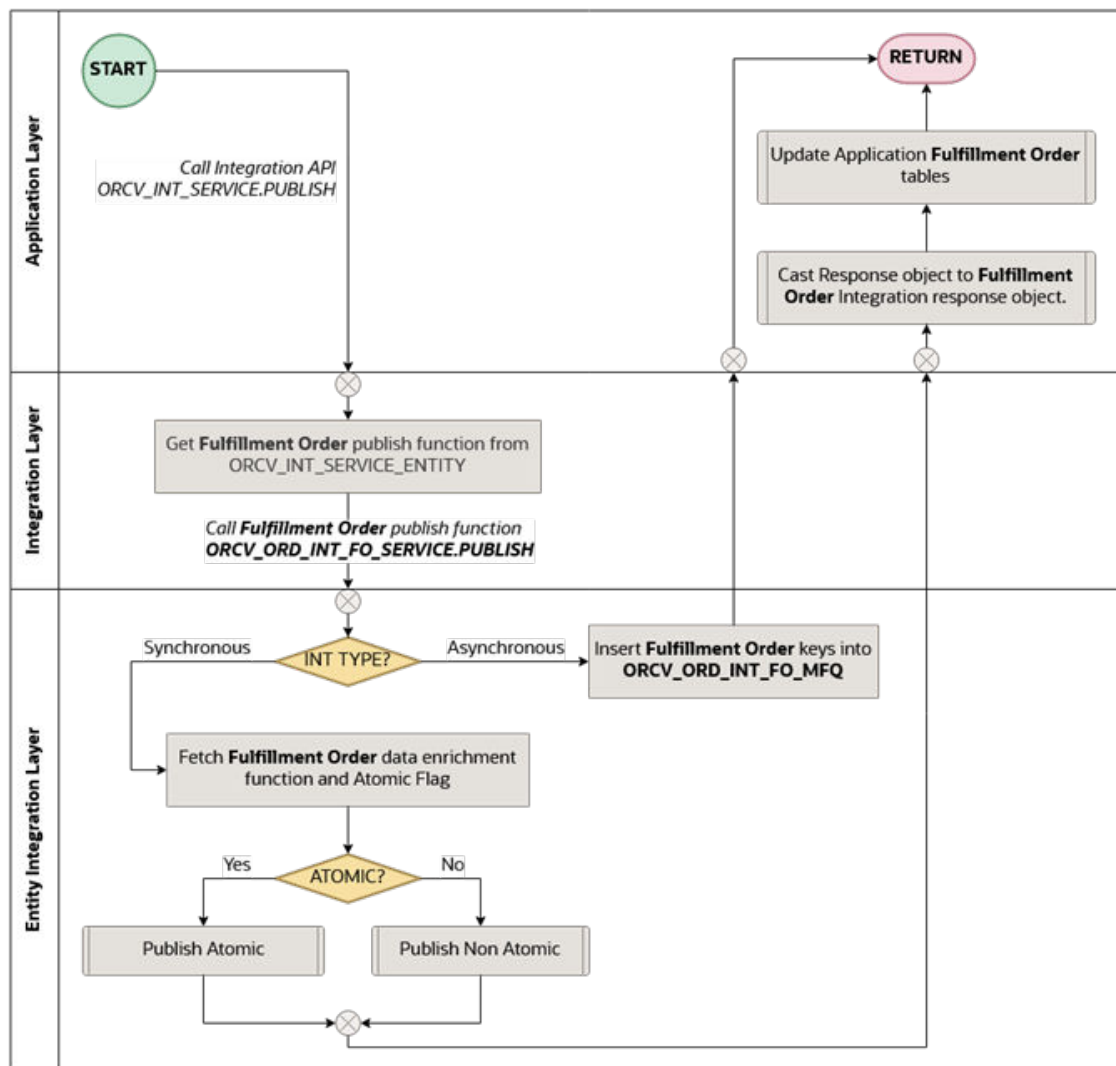
Process Overview

This section describes the process overview.

Publish Flow

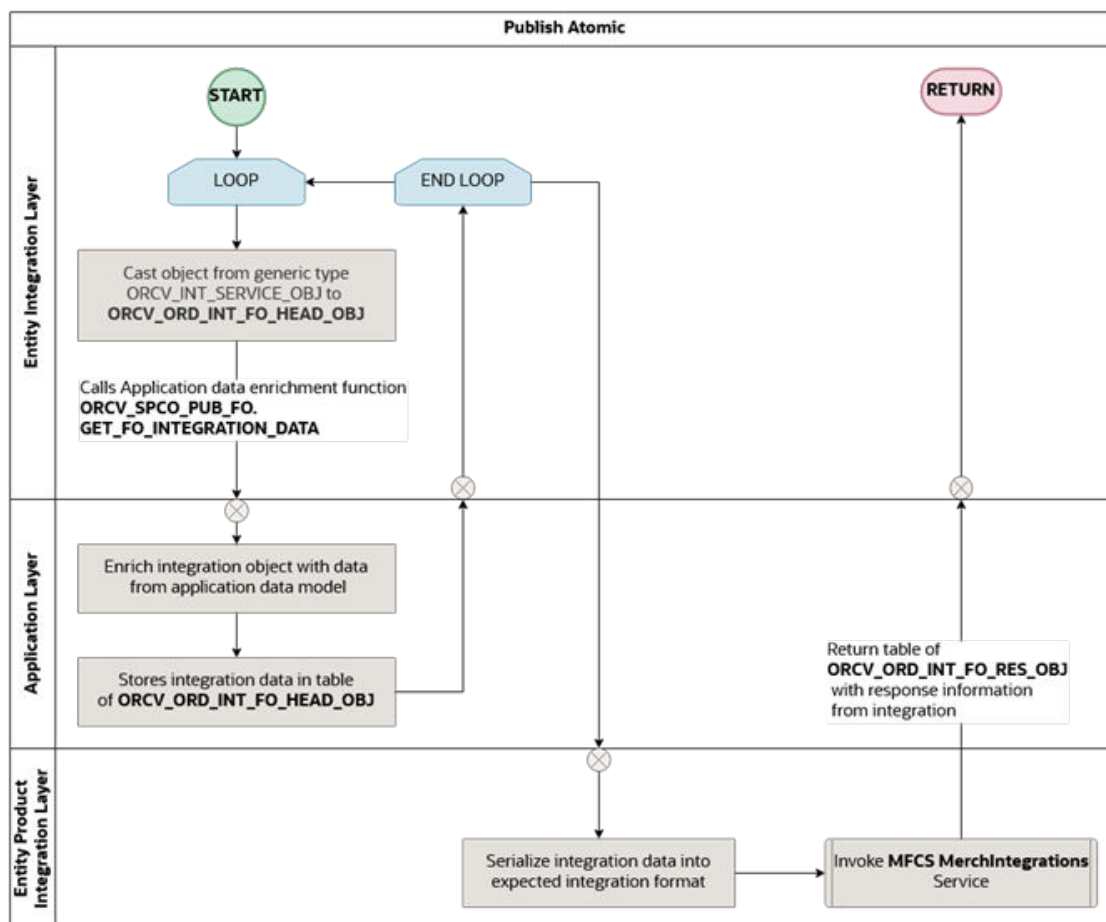
Publish allows the call of integration endpoints in synchronous and asynchronous mode. Those calls can be either atomic or non-atomic. Atomic is in case all the entries need to be integrated together in bulk while the non-atomic deals with each entry individually.

Figure 27-102 Fulfillment Order to MFCS Publish Flow



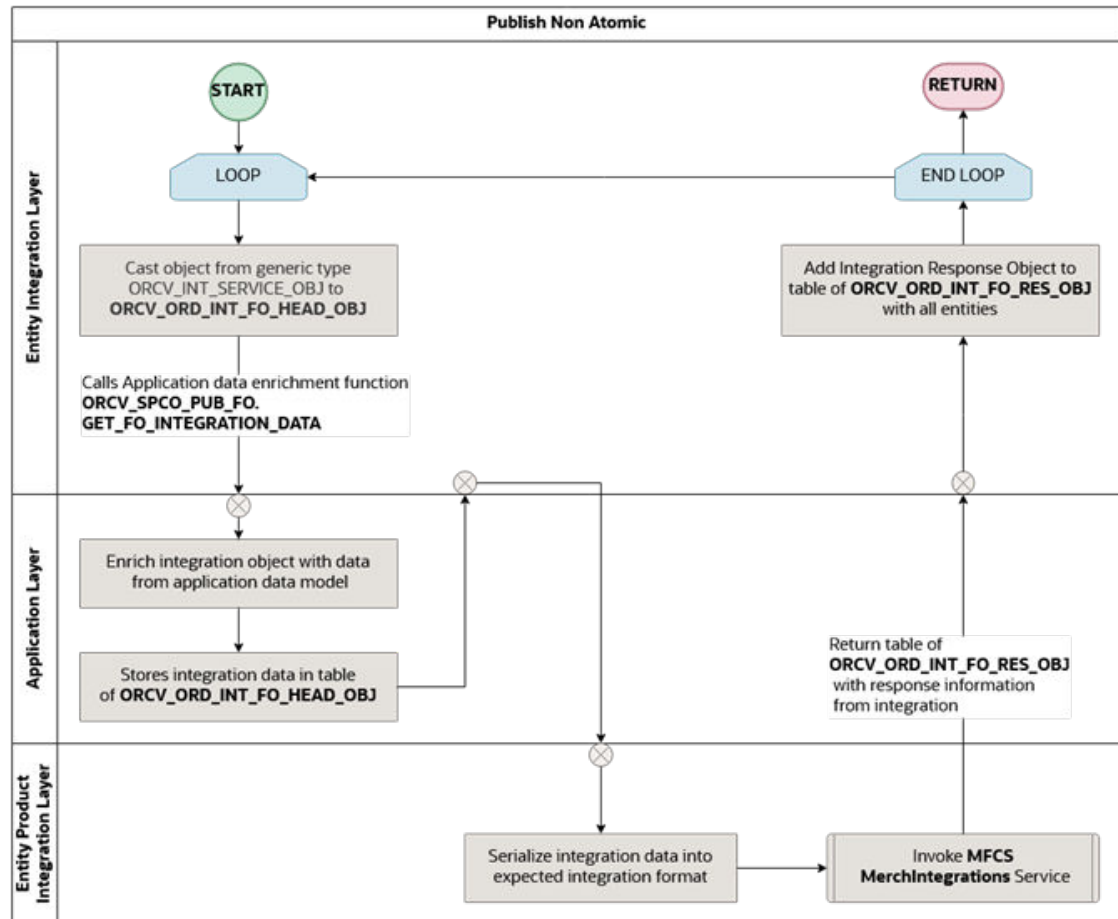
Publish Atomic

Figure 27-103 Fulfillment Order to MFCS Publish Atomic



Publish Non Atomic

Figure 27-104 Fulfillment Order to MFCS Publish Non-Atomic



Publish Asynchronous Batch

Fulfillment Order Publish to MFCS

The Fulfillment Order Publish to MFCS Batch checks the queue table `ORCV_ORD_INT_FO_MFQ` for Fulfillment Orders to be published in the table.

- Job: `ORCV_ORD_INT_FO_MFCS_JOB`
- Frequency: Ad-hoc
- Dependency: None
- Inter Scheduler Dependency: None

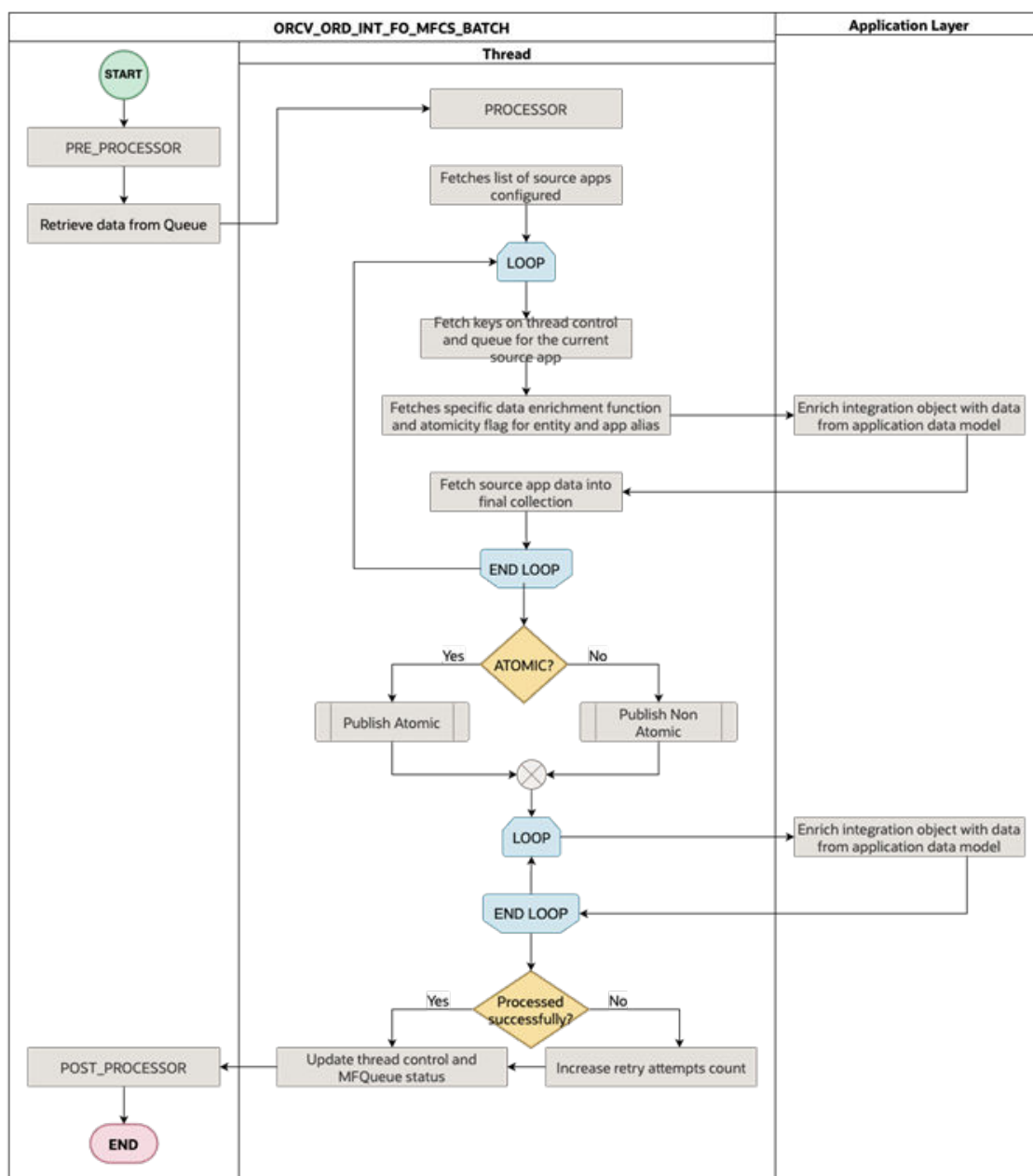
Batch default configuration (`ORCV_BATCH_CONFIG`):

Column Name	Default Values
Program Name	<code>ORCV_ORD_INT_FO_MFCS_JOB</code>

Column Name	Default Values
Maximum Chunk Size	10000
Maximum Concurrent Threads	4
Retry Lock Attempts	4
Retry Wait Time	2
Retry Maximum Attempts	3
Retry Ordered	N
Retry Indicator	N

Process Overview

Figure 27-105 Fulfillment Order to MFCS Publish Asynchronous Batch Overview



Configuration

Customer Order Fulfillment publishes Fulfillment Orders with the following configurations in Database Integration Framework:

Table ORCV_INT_SERVICE_ENTITY

Configuration	Value
Entity ID	FO_MFCS
Publication Function	ORCV_ORD_INT_FO_SERVICE.PUBLISH

Table ORCV_INT_SERVICE_MAPPING

Configuration	Value
Application Alias	SPCO
Application Entity Get Data Function	ORCV_SPCO_PUB_FO.GET_FO_INTEGRATION_DATA
Application Status Function	ORCV_SPCO_PUB_FO.SET_FO_INTEGRATION_STATUS
Atomic Indicator	N

Table ORCV_WS_ENDPOINT

Configuration	Value
Endpoint ID	MFCS_MERCH_FO_MANAGE
Credential Static ID	OAUTH_MFCS_INTEGRATION
URL	<MFCS_MerchIntegrations_URL>/services/customerOrders/fulfillmentOrder/create
Action	PUT
Request Type	REST

Integration Object Type

ORCV_INT_SERVICE_OBJ is a base object type for handling the reference keys sent for integration and Fulfillment Order to MFCS implements ORCV_ORD_INT_FO_HEAD_OBJ under this object following the Fulfillment Orders specific requirements.

Security

Authentication and Single Sign-On

Authentication in Retail Supply Chain Hub is managed through Oracle Cloud Infrastructure Identity and Access Management (OCI IAM). The IDCS or OCI IAM tenant is the same tenant that is used by the rest of the Oracle Retail suite of applications, enabling SSO or federated with another instance.

ReST Services Authentication

ReST Service services endpoints support IDCS Authentication OAuth 2.0 with grant types of Client Credentials. It is applicable to all ReST endpoints in ORDS. The Java Services all support OAuth2.0, but only with grant type Client Credentials.

To ensure a proper security configuration the following must be done:

- Client Credentials for batch executions
- Client Credentials for Integrations:
 - Create an Application in IDCS with a specific scope for the client.
 - Create a user in IDCS with the uppercased name matching the IDCS Application name.
 - Grant IDCS Application Role RETAILER_ADMINISTRATOR for RSCH.
 - Run IDCS Synchronization Batch.
 - Map the client scopes with the appropriate roles of the endpoint in table ORCV_SEC_ROLES_SCOPES.
 - Set appropriate Data permissions for the actions to be performed in Profile.

Database

Data Filter Policy

Retail Supply Chain Hub provides an optional layer of data filtering in the application user interface, which limits the data end users see by levels in the merchandise, supplier site and location. Whether or not this is used in your environment is controlled by association to a supplier, department, location, or a specific role that determines if the user is a retailer or a supplier and they are an administrator of the system.

With data filtering enabled for merchandise hierarchy, users will only be able to work with entities that are part of the merchandise hierarchy to which they have been given data filtering access. Otherwise, by default, any retailer user will have access to all departments. This configuration will be persisted in the table ORCV_SEC_USER_DEPT and the user information will need to be set up on the ORCV_SEC_USER table.

GDPR

Retail Supply Chain Hub complies with GDPR requirements that mandates that organizations protect the personal data of individuals and provides individuals with rights regarding their data, such as the right to access, rectification, erasure, and data portability. In essence, the available web-services manage the process of identifying, modifying, and/or deleting personal data across various tables in the ORCV database to comply with GDPR regulations.

Web Services

This section describes the web services.

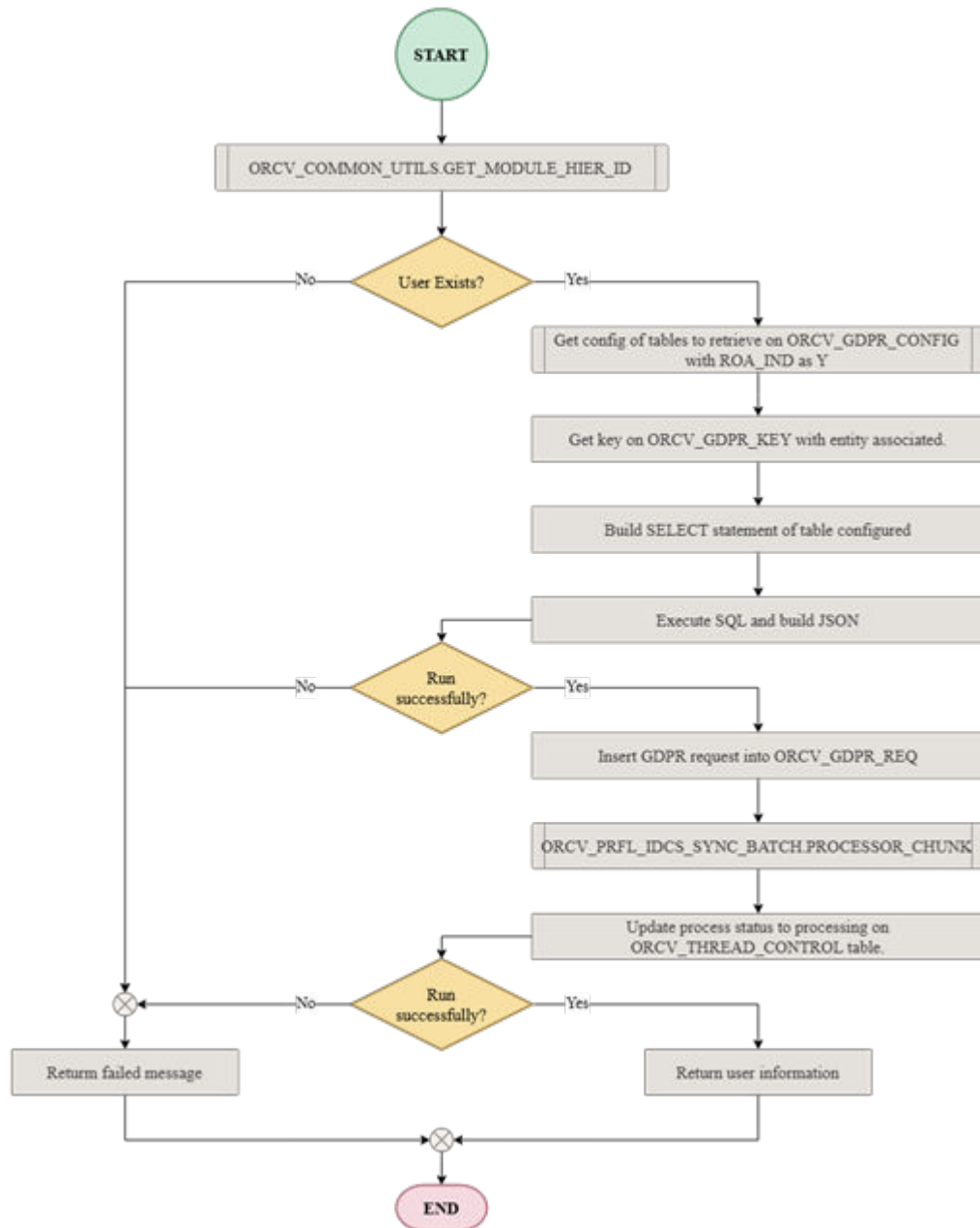
Right of Access

This service retrieves and returns user information based on the provided user ID.

Endpoint	/ords/mfcs/orcv/common/gdpr/access
Authorization Method	OAuth 2.0
Request Method	POST
Request Body	The request body should be a JSON object containing the user_id. Example: { "user_id": "MY_USER" }
Response Body	A JSON object containing the user's information. Example: { "entity": "User with id RETAILER_ADMIN information.", "list": [{ "USER_ID": "MY_USER", "USER_NAME": "My User", "FIRST_NAME": "My", "LAST_NAME": "User", "USER_STATUS": "A", "JOB_TITLE": null, "EMAIL": "my.user@example.com", "USER_TYPE": "RA" }] }

This endpoint will verify the user exists in the system. If the user is confirmed to exist, the function proceeds to gather all the relevant information associated with that user and returns it in a JSON format.

Figure 27-106 GDPR Right of Access Rest Service



Right to be Forgotten

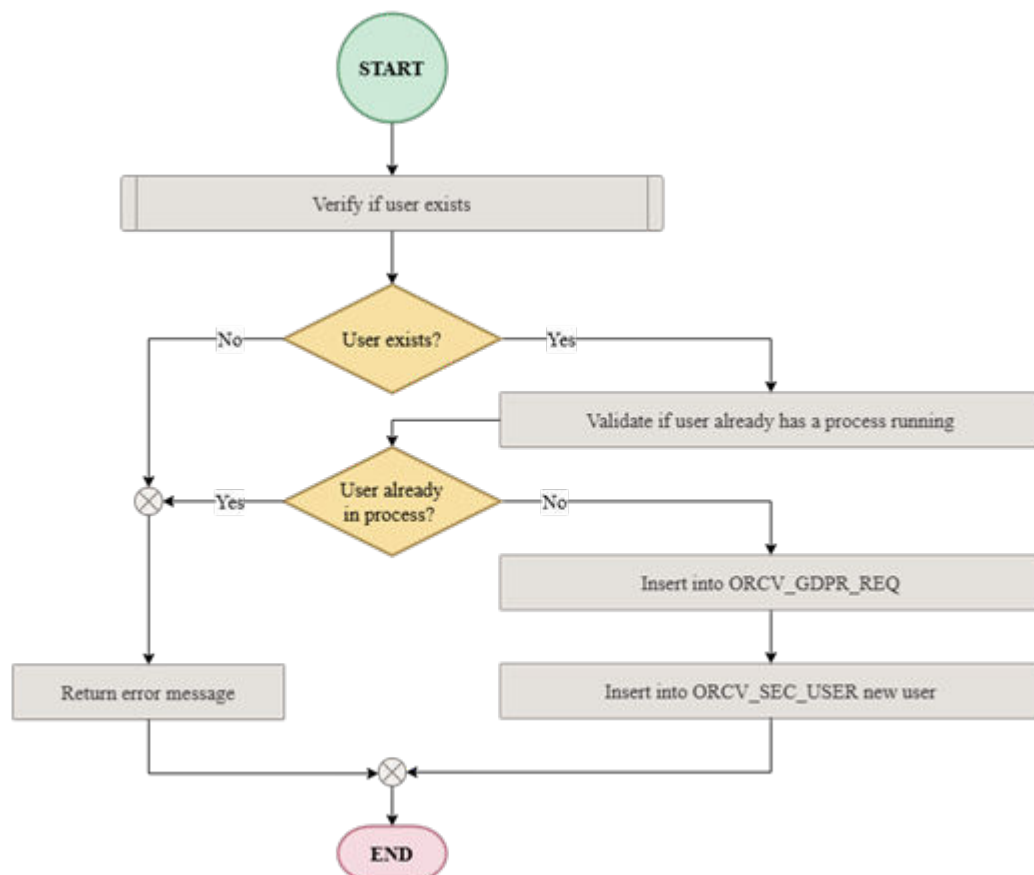
This service initiates the process of forgetting a user's data in accordance with GDPR regulations.

URL	<code>/ords/mfcs/orcv/common/gdpr/forget</code>
Authorization Method	OAuth 2.0
Request Method	POST

URL	<code>/ords/mfcs/orcv/common/gdpr/forget</code>
Request Body	The request body should be a JSON object containing the user_id. Example: <pre>{ "user_id": "RETAILER_ADMIN" }</pre>
Response Body	A JSON object containing the process id of the batch executing in the background that will obfuscate the user id requested. Example: <pre>{ "process_id": "1234" }</pre>

This endpoint will verify the user exists in the system. If the user is confirmed to exist, the function proceeds to adding the user to be processed in the batch.

Figure 27-107 GDPR Right to be Forgotten Rest Service



Forget Batch

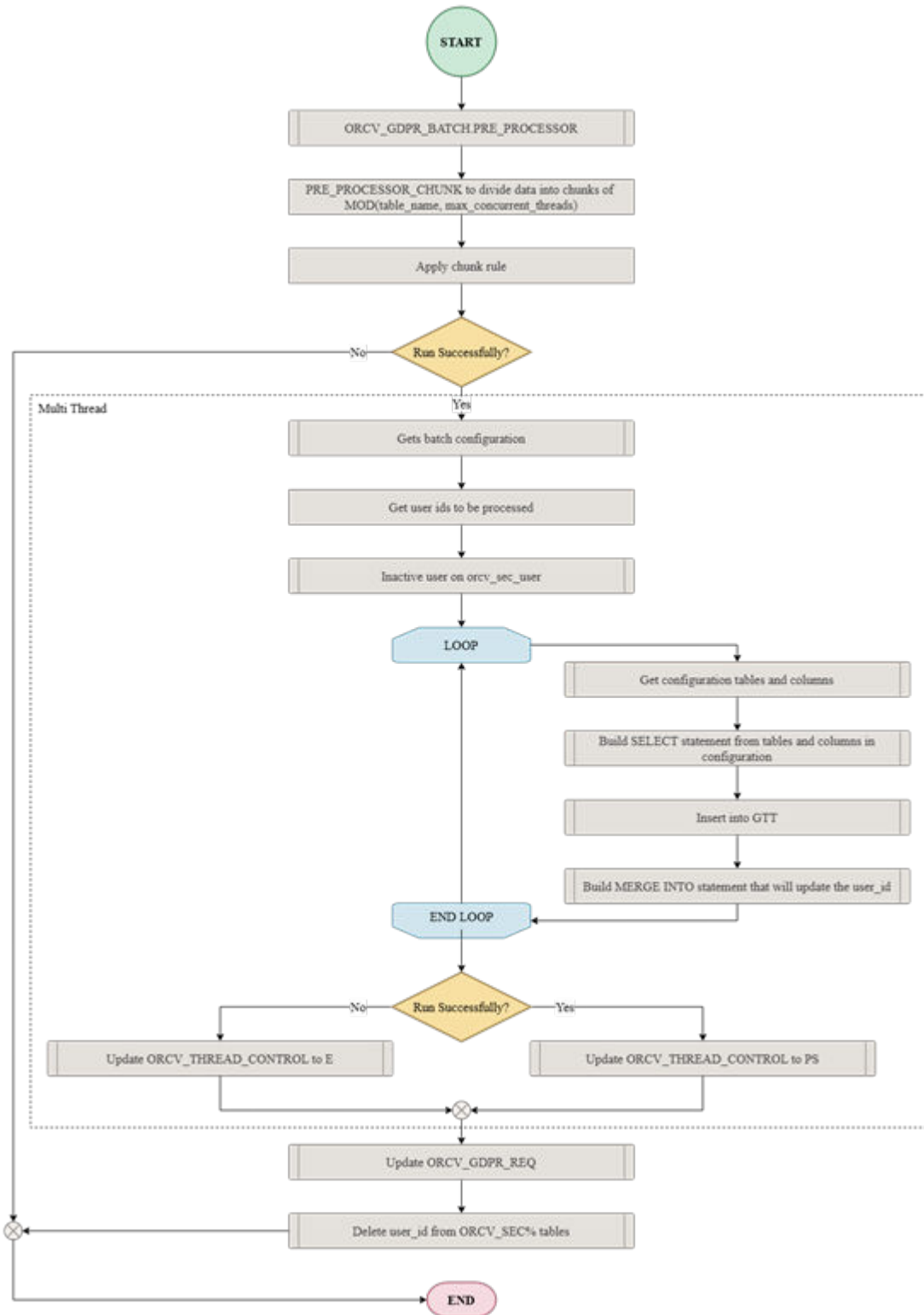
This process executes the 'forget' functionality for all tables configured in the ORCV_GDPR_CONFIG table where ROA_IND is 'N'. It updates the columns specified in the GDPR_COLUMN to anonymize user data.

Batch configuration table: ORCV_BATCH_CONFIG

Column Name	Default Values
PROGRAM_NAME	ORCV_GDPR_JOB
MAX_CHUNK_SIZE	1000
MAX_CONCURRENT_THREADS	4
RETRY_IND	N
RETRY_LOCK_ATTEMPTS	3
RETRY_MAX_ATTEMPTS_COUNT	3
RETRY_ORDERED	N
RETRY_WAIT_TIME	3

Process Overview

Figure 27-108 Forget Batch Process Overview



Pre-Processor: This function's primary responsibility is to divide the workload of the processing into smaller chunks that can be processed concurrently by different threads. This is achieved by collecting all the configurations in ORCV_GDPR_CONFIG table and defining the threads based on MOD of (TABLE_NAME, MAX_CONCURRENT_THREADS) to ORCV_THREAD_CONTROL.

Processor: The processor function is responsible for the bulk of the execution. It begins by inactivating the user identified for removal to prevent further actions associated with that user. The function updates the ORCV_GDPR_REQ table to mark the process as 'P'. After the handling of the thread number, it will update the ORCV_THREAD_CONTROL table. After this, the function will run through every TABLE_NAME configured on the ORCV_GDPR_CONFIG table and insert all records associated with the user in the configured column into a GTT table (ORCV_GDPR_USER_ACTIV_GTT). It will then finally update all columns with the user to the new created user inserted on the beginning of the process. The function concludes by updating the ORCV_THREAD_CONTROL table to 'PS' for success or 'E' for error."

Post Processor: The post-processor begins by checking the status of the ORCV_THREAD_CONTROL table for the given PROCESS_ID. If all threads completed successfully, the process status in ORCV_GDPR_REQ is set to 'C' (Completed); otherwise, it is set to 'E' (Error). If the status in ORCV_GDPR_REQ is 'C', the user is deleted from the ORCV_SEC_USER_APP, ORCV_SEC_USER_SUPP, ORCV_SEC_USER_DEPT, ORCV_SEC_USER_LOC, ORCV_SEC_TOKEN_HEAD, and ORCV_SEC_USER tables.

Operations

Authentication Method: OAuth2

Base URL: /ords/mfcs/orcv/merch/common/batch/<URL_PATH>

URL Path	Request Method	Request Body	Affected Tables
start	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "", "jobName": "ORCV_GDPR_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_GDPR_REQ ORCV_SEC_USER ORCV_SEC_USER_APP ORCV_SEC_USER_SUPP ORCV_SEC_USER_DEPT ORCV_SEC_USER_LOC ORCV_SEC_TOKEN_HEAD and tables listed in ORCV_GDPR_CONFIG

URL Path	Request Method	Request Body	Affected Tables
restart	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "<EXEC_ID>", "jobName": "ORCV_GDPR_JOB", "agentExecutionId": "", "parameters": "" }	ORCV_BATCH_STATUS_CONTROL ORCV_THREAD_CONTROL ORCV_BATCH_AUDIT_STATUS ORCV_GDPR_REQ ORCV_SEC_USER ORCV_SEC_USER_APP ORCV_SEC_USER_SUPP ORCV_SEC_USER_DEPT ORCV_SEC_USER_LOC ORCV_SEC_TOKEN_HEAD and tables listed in ORCV_GDPR_CONFIG
log	GET	URL query parameters: ?jobName=ORCV_GDPR_JOB& executionId=<EXEC_ID>& logDate=YYYY-DD-MM	N/A
kill	POST	{ "cycleName": "", "flowName": "", "processName": "", "async": "", "executionId": "<EXEC_ID>", "jobName": "ORCV_GDPR_JOB", "agentExecutionId": "", "parameters": "" }	N/A
status	GET	URL query parameters: ?jobName=ORCV_GDPR_JOB& executionId=<EXEC_ID>& logDate=YYYY-DD-MM	N/A

Right to be Forgotten Service

This service retrieves the status of a "forget user" request, identified by its process ID.

URL	/ords/mfcs/orcv/common/gdpr/forget
Authorization Method	OAuth 2.0
Request Method	GET
Request Parameters	The process_id received on the forget user web service. Example: /ords/mfcs/orcv/common/gdpr/forget?process_id=1234

URL	/ords/mfcs/orcv/common/gdpr/forget
Response Body	A JSON object containing the status of the forget user process. Example: { "status": "New" }

Glossary

API

Application Programming Interface, a type of web service. It provides a way of communicating with a particular computer program or internet service, such as the interfaces between Supplier Evaluation and other systems, by exchanging data as XML files.

DELETE

A web service call used to delete an existing resource (not generally used in the APIs exposed in Supplier Evaluation).

GET

A web service call used to retrieve data without modifying it, for example, to request details of specifications.

HEAD

Identical to GET except that the server does not return a message-body in the response. The header information retrieved is the same as for a GET request and obtains meta-information without transferring the entity-body itself.

HTTP

Hyper Text Transfer Protocol, an application protocol used as the foundation for data communication on the World Wide Web.

HTTPS

Communication over Hyper Text Transfer Protocol which is protected with an additional security layer. Used to authenticate the site visited and protect the exchanged data.

LCA

Life Cycle Assessment, a methodology for assessing environmental impacts associated with the stages of the life cycle of a product, process, or service. LCA partners may be used in the process of assessing products or supplier for the retailer (as a third-party accredited body).

oneTimePad

A secure encryption technique which uses a string of randomly generated numbers to produce a key for encoding and decoding a message. Both the sender and receiver of the message have a copy of the key and each key (or section of the key) is used only once.

POST

A web service call used to create new resources, for example, to create a new supplier or new user record.

PUT

A web service call used to update an existing resource, for example, to update/amend an existing supplier or user record.

REST / RESTful

Representational State Transfer, an architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements, within a distributed system.

In Supplier Evaluation, most new services or new versions of services will use RESTful architecture.

SOAP

Simple Object Access Protocol, an XML-based protocol for exchanging data over HTTP. In Supplier Evaluation, SOAP services are developed for specific, definite requirements or tend to be associated with historical services.

URI

Uniform Resource Identifier, a string of characters used to identify the name of a resource and enable it to be accessed over a network such as the World Wide Web. A common example of a URI is the Uniform Resource Locator (URL), or web address.

WADL

Web Application Description Language, an XML-based interface description language for describing the functionality offered by a RESTful web service.

WSDL

Web Services Description Language, an XML-based interface description language for describing the functionality offered by a SOAP web service.

XML

Extensible Markup Language, a language used to encode a document in a way which is both human readable and machine readable.

Markup symbols are used to define the content of the document and to indicate what that content is associated to in the sending or receiving system.

XSD

XML Schema Definition, defines the structure and data type of the elements in an XML document. XSD is used to validate the content of an XML document.