

Oracle® Retail Process Orchestration and Monitoring Implementation Guide



Release 22.1.401.0

F71170-01

October 2022

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 2022, Oracle and/or its affiliates.

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, and MySQL 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	viii
Documentation Accessibility	viii
Customer Support	viii
Improved Process for Oracle Retail Documentation Corrections	ix
Oracle Retail Documentation on the Oracle Help Center (docs.oracle.com)	ix
Conventions	ix

1 Introduction

2 Batch Concepts

Batch Hierarchy	2-1
Cycle	2-1
Flow	2-2
Process	2-2
Job	2-2
Internal Dependencies	2-2
Inter-Schedule Dependencies	2-2
External Dependencies	2-2

3 Integration

Setting Up a New Batch Schedule in POM	3-1
Invoking Cycles in POM	3-1
Execution Request Creation	3-1
Execution Request Status Tracking	3-4
External Status Update	3-6
Schedule Configuration	3-6

Job Configuration	3-7
Payload Specification	3-9
External Dependency	3-11
Configuration	3-12
Releasing Dependency	3-13

4 Schedule Customization

5 Custom Schedules

Introduction	5-1
Set Up the Schedule Spreadsheet	5-1
Configure the New Schedule	5-1
Load the New Schedule	5-1
Schedule the Batch Tasks	5-2
Run Batch	5-2

6 Generic ReST Jobs

Introduction	6-1
Features of Generic ReST Jobs	6-1
Execution Sequence	6-2
Handling Restarts	6-3
Handling Kills	6-4
Defining Custom Job Types	6-4
Batch Schedule Spreadsheet	6-4
POM UI	6-4
Environment-specific Information	6-4
Internal Representation	6-4
Job Parameter Restrictions	6-5
Custom Job Type Endpoint Specifications	6-5
Authentication	6-5
Job Start API	6-5
Job Status API	6-6
Job Restart API	6-7
Job Kill API	6-7
Job Log API	6-8
Job System Check API	6-8

7 Export/Import Schedule Configuration

Overview	7-1
Schedule Info Tab	7-2
Schedule Configuration Tab	7-2
Throttling Tab	7-3
Nightly Jobs Configuration Tab	7-4
Recurring Jobs Configuration Tab	7-5
Flows Configuration Tab	7-6
Adhoc Jobs Configuration Tab	7-7
Job Dependencies Tab	7-8
Job External Associations Tab	7-9
Scheduling Flows Tab	7-10
Scheduling Adhoc Tab	7-12
Notification Tab	7-13

8 Notifications

9 User Roles and OAuth Scopes

10 Invoking POM Services Using OAuth

Prerequisite	10-1
Requesting the Access Token	10-1
Invoking the POM Service	10-2

A Batch Schedule Spreadsheet Explained

Overview	A-1
Guidelines for Updating the Batch Schedule Spreadsheet	A-1
Process Tab	A-1
Job Tab	A-2
Process Job Mapping	A-4
Dependency Tab	A-5
Recurring Flow Tab	A-6
Recurring Flow Process Tab	A-6
Obsolete Tab	A-7
System Option Tab	A-8
Application Tab	A-9
Schedule Tab	A-10

Throttling Configuration Tab	A-10
Inter Schedule Dependency Tab	A-11
Modules Tab	A-12
External Dependencies Tab	A-13
Batch Links Tab	A-13
Job Types Tab	A-14

Send Us Your Comments

Oracle Retail Process Orchestration and Monitoring Implementation Guide

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 Process Orchestration and Monitoring Implementation Guide* describes the requirements and procedures to install this Oracle Retail Product release.

Audience

This Implementation Guide is for the following audiences:

- System administrators and operations personnel
- Database administrators
- System analysts and programmers
- Integrators and implementation staff personnel

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.

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

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 Technology Network 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 Technology Network at the following URL:

<http://www.oracle.com/technetwork/documentation/oracle-retail-100266.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 also 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.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction

The Process Orchestration and Monitoring (POM) application provides comprehensive batch scheduling and monitoring capabilities for Oracle Retail SaaS Applications.

POM's key features are:

- Support for Cyclical (or Hourly)/Ad-hoc (or Standalone)/End of Day (or Nightly) scheduling modes
- Configurable Schedule Invocation - Time/Externally triggered
- Configurable Schedule - Configure schedule specific to customer needs
- Hooks to integrate with external (customer) systems - For dependency management and job status updates
- Configurable notifications

2

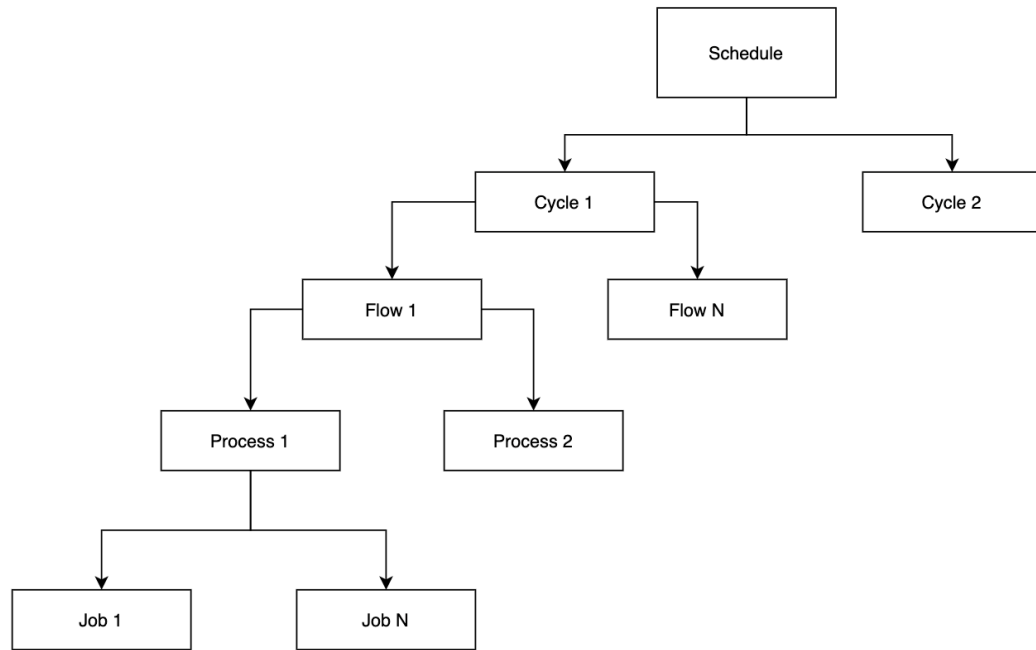
Batch Concepts

This chapter describes the Process Orchestration and Monitoring (POM) concepts that are key to configure and implement the product successfully.

Batch Hierarchy

POM logically groups batch jobs in a schedule into a hierarchy of Processes, Flows and Cycles based on functionality and expected execution frequency.

Figure 2-1 Batch Hierarchy Diagram



Cycle

A Batch Cycle is a logical collection of flows. Cycles are categorized into 3 groups,

1. Ad hoc/Standalone

The jobs that can run multiple times a day on an as-needed basis are classified as Ad hoc cycles. Such processes are allowed to run independent of other processes in a Flow/ Cycle

2. Recurring/Hourly

Recurring or Hourly jobs are a special group of jobs which run multiple times a day at scheduled intervals.

Example: The sales processing jobs in RMS support trickle processing by running every 30 minutes during the store trading hours. The schedule is pre-loaded with 24 cycles. Each cycle can have one or more flows where each flow is a set of jobs. The recurring cycles are time triggered from the Scheduler. Based on the client's business operations, the individual flows of each cycle will need to be scheduled. If any flow/cycle is not used then it has to be disabled as the nightly cycle will not start until the enabled cycles are all complete.

3. End of Day/Nightly

This contains the set of jobs that are executed at end of the business day. The nightly cycle is time triggered from Scheduler. The nightly cycle will start only when all the loaded recurring flows are complete.

Flow

A batch flow is a logical collection of batch processes that run together. The batch flow always starts and ends with a single process, but it can run parallel processes in the middle of the batch flow.

Process

A batch process is a logical collection of batch jobs that always run in sequential order.

Job

A batch job is smallest entity in POM that can be scheduled. A job represents an actual application script.

Internal Dependencies

Internal dependencies are the dependencies between the process/job of the same schedule.

Inter-Schedule Dependencies

Inter-Schedule dependencies are the dependencies between jobs of the different schedules running on the same POM instance.

External Dependencies

External dependencies are the dependencies between the jobs running on external systems, such as a customer's system and jobs running on the POM schedule.

3

Integration

This chapter describes the various scenarios which involve configuring & integrating the Process Orchestration and Monitoring (POM) application to connect with external systems.

Setting Up a New Batch Schedule in POM

When POM is first installed for a specific customer, it does not include any application batch schedules out of the box such as Merchandising or Retail Intelligence, and so on. An Oracle administrator or a system integrator need to first configure those schedules before they get loaded with the scheduling data. Configuring a new schedule entails setting up schedule properties such as the schedule name and description, and customer environment information for callbacks. It also entails setting up the location of different components and services with which different POM components need to interact to function properly.

Refer to the section "Configure New Schedule" in the "System Configuration" section of the *POM User Guide*.

Invoking Cycles in POM

Different SaaS customers operate in different models for running batch.

For some, the Oracle support team configures the time and frequency for running the batch cycles in POM on their behalf, based on their requirement.

Others may choose to use the Scheduler feature of the product. Refer to the *POM User Guide* for documentation on the Scheduler.

For others, the capability exists for them to control the time and frequency of batch executions by invoking the following ReST service.

Some may use a hybrid of all or some of the above methods.

Note:

All POM endpoints are protected by the OAuth authorization protocol. To invoke an endpoint in POM, a customer application must use an Access Token that was generated using the OAuth Client Credentials Grant. Refer to Invoking POM Services Using OAuth in this guide for further details.

Execution Request Creation

The specification of the ReST service to start the POM cycle execution is shown below:

HTTP Method	POST
Path	<p>http://<pom-server-host>/ProcessServices/services/private/executionEngine/schedules/<Schedule_Name>/execution</p> <p>Note: <pom-server-host> and <Schedule_Name> variables need to be changed according to the target environment and schedule.</p>
HTTP Headers	<p>Content-Type = application/json</p> <p>Authorization = Bearer <<OAuth_Token_FROM_IDCS_or_OCI_IAM>></p>
Request Body	<pre>{ "cycleName": "<Cycle Name>", "flowName": "<Flow Name>", "processName": "<Process Name>", "requestParameters": "<Comma Separated Key Value Pairs>" }</pre>
	<p>Notes:</p> <ul style="list-style-type: none"> • processName attribute required only for Ad hoc cycles • requestParameters attribute is optional. This is useful if external systems would like to provide custom identifiers to POM and expect them to be returned on callbacks sent from POM.
Response Body	<pre>{ "value": "171" -Execution Request ID, "cycleName": "CycleName", "flowName": "FlowName", "processName": "ProcessName", "jobName": "JobName", "requestType": "requestType", "requestParameters": ":"<Comma Separated Key Value Pairs>" "executionEngineInfo": "STARTED", "hyperMediaContent": {} }</pre>

Invocation	Request Payload
Nightly Cycle invocation	<pre>{ "cycleName" : "Nightly", "flowName" : "Nightly", "requestParameters" : "callerId=XXX,correlationId=123" }</pre> <p>Note: The Nightly Cycle contains a single flow hence a single invocation will suffice to start the Nightly Flow.</p>

Invocation	Request Payload
Hourly Cycle invocation	<pre data-bbox="860 294 1315 525">{ "cycleName" : "Hourly_Cycle_<N>", "flowName" : "<FLOW_NAME>", "requestParameters" : "callerId=XXX,correlationId=456" }</pre> <p data-bbox="860 567 1315 693"><N> is the cycle number (1 to 24) <FLOW_NAME> is name of the flow (Process Group) to invoke. For example, SALESPROCESS_CYCLE.</p> <pre data-bbox="860 735 1364 1008">{ "cycleName" : "Hourly_Cycle_1", "flowName" : "SALESPROCESS_CYCLE", "requestParameters":"callerId=Sys_N ame,correlationId=456" }</pre> <p data-bbox="860 1050 1364 1155">Note: The Hourly Cycles comprise of many distinct flows and for each flow a separate invocation would be required.</p>

Invocation	Request Payload
<p>Ad hoc Cycle invocation</p> <p>In case of Adhoc processes, the parameters for Jobs can be overridden by parameters specified as part of the invocation request.</p>	<pre data-bbox="868 304 1323 556"> { "cycleName" : "Adhoc", "flowName" : "Adhoc", "processName" : "<ADHOC_PROCESS_NAME>", "requestParameters" : "callerId=xxx,correlationId=789" } </pre> <p>Example:</p> <pre data-bbox="868 672 1364 987"> { "cycleName": "Adhoc", "flowName": "Adhoc", "processName": "RPM_NEW_ITEM_LOCATION_PROCESS_ADHOC", "requestParameters": "callerId=Ext_System_Name, correlationId= 789" } </pre> <p>Example with overriding job parameters:</p> <pre data-bbox="868 1102 1364 1480"> { "cycleName": "Adhoc", "flowName": "Adhoc", "processName": "RPM_NEW_ITEM_LOCATION_PROCESS_ADHOC", "requestParameters": "callerId=Ext_System_Name, correlationId=789,jobParams.RPM_NEW_ITEM_JOB=param1 param2" } </pre> <p>Note: Ad hoc Cycles are composed of many discrete individual Processes. For each Process, a separate invocation is required.</p>

Execution Request Status Tracking

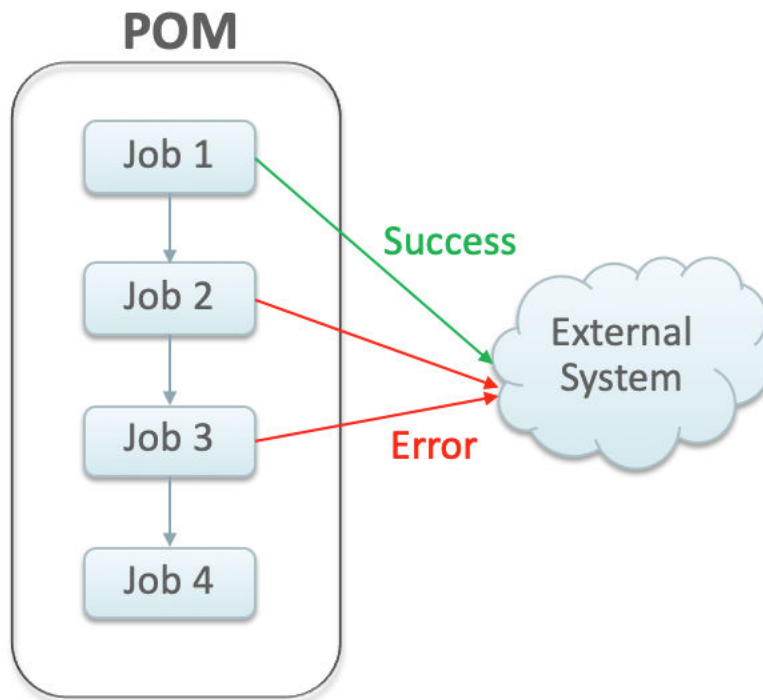
The following endpoints provide the ability to check the status of an execution request in POM:

HTTP Method	GET
Path	<p>http://<pom-server-host>/ProcessServices/services/private/executionEngine/schedules/<scheduleName>/requests/{executionId}</p> <p>Note: Replace <pom-server-host> according to the target environment.</p> <p><scheduleName>: Name of the schedule. For MOM apps, the schedule name is "MERCH". For RDE, the schedule name is "RDE".</p> <p><executionId>: ID of the execution request returned by POM when execution request was submitted.</p>
HTTP Headers	<p>Content-Type = application/json</p> <p>Authorization = Bearer <<OAuth_Token_FROM_IDCS_or_OCI_IAM>></p>
Response Body	<ul style="list-style-type: none"> • executionId - ID of the execution request • scheduleName - Name of the schedule for which this execution request was created. Sample values: MERCH, RDE, and so on. • cycleName - Name of the Cycle for which this execution request was created. Sample values: Nightly, Hourly, or Adhoc. • flowName - Name of the Flow for which the execution request was created Sample Values: For Ad hoc Cycle: Adhoc For Nightly Cycle: Nightly For Hourly Cycle: Name of the flow such as DEAL_PURGE_CYCLE, and so on. • processName - Name of the Process for which the execution request was created. Sample Values: For Ad hoc: Name of the process such as POINDBATCH_PROCESS_ADHOC, and so on. For Nightly/Hourly: Always set to "ALL". • requestParameters - Parameters associated with the execution request. • status - Status of the execution request. • Possible Values: • QUEUED: Request is queued up for execution. • RUNNING: Jobs from this request are being executed • ERROR: One of the job in this request has failed. Note that a failed job would be restarted by POM Admin; there is no need to re-submit the execution request. • COMPLETED: All jobs from this request were executed successfully.

External Status Update

The External Status Update feature provides the ability for external systems to register with POM to receive the Job status notifications as a callback to their ReST interface.

Figure 3-1 External Dependency



Schedule Configuration

This section details the steps to configure the External Status Updates feature at the schedule level:

1. Navigate to the System Configuration screen.
2. Click the Edit icon on the External Configuration Panel to open the External Configuration window.

Figure 3-2 External Configuration Window

External Configuration for RSP 19.0.0

External Status URL

External Status Update Mode

Credentials

3. Enter the configuration values:

- **External Status URL** - External system's URL that needs to be called for status updates.

 **Note:**

In addition to this configuration, you must work with Oracle support to get the External Status URL allowlist.

- **External Status Update Mode** - Choose one of the options below:
 - **ALL** - POM will send a status update to the external system for each job's execution in the schedule regardless of success or failure.
 - **FAILED** - POM will notify only for failed jobs.
 - **NONE** - No status updates will be sent by POM.

 **Note:**

The External Status Update Mode defined on this screen applies to all the jobs in a schedule. If status update is desired only for specific jobs then set the mode on the above screen to NONE and follow the steps defined in the Job Configuration section below.

- Click **Update Credentials** and provide the credentials for the external system.

Job Configuration

This section describes the steps to configure the External Status Update Mode at job level.

1. Navigate to the Batch Administration screen and select the schedule to update with the external status.

Figure 3-3 Batch Administration Screen

The screenshot shows the 'Batch Administration' screen with a summary of four schedules: RSP 18.0.002, MERCH 19.0.001, R1 18.0.002.6.1, and RDE 18.0.002.3. Below this, the 'RSP Standalone' section is active, displaying a table of jobs. The table has columns for Enabled, Job, Process Name, Application, Module, Parameters, Active Parameter, External Status Update, Skip On Error, and Thre Time. The first row is selected, showing job 'AC_PROD_ATTR_LOC_SHARE_PROCESS_A' with process name 'AC_PROD_ATTR_LOC_SHARE_ADHOC' and application 'RSP'.

Enabled	Job	Process Name	Application	Module	Parameters	Active Parameter	External Status Update	Skip On Error	Thre Time
<input checked="" type="radio"/>	AC_PROD_ATTR_LOC_SHARE_PROCESS_A	AC_PROD_ATTR_LOC_SHARE_ADHOC	RSP	CS_Attr_S...	null	null	None	<input checked="" type="checkbox"/>	0
<input type="radio"/>	AC_PROD_ATTR_LOC_SHARE_SETUP_ADH	AC_PROD_ATTR_LOC_SHARE_ADHOC	RSP	CS_Attr_S...	-YYYYMMDD -YYYYMMDD -1Y	-YYYYMM...	None	<input type="checkbox"/>	0
<input type="radio"/>	DT_LOC_RANGE_PROCESS_ADHOC_JOB	DT_LOC_RANGE_ADHOC	RSP	DT_Required	null	null	None	<input checked="" type="checkbox"/>	0
<input checked="" type="radio"/>	DT_LOC_RANGE_SETUP_ADHOC_JOB	DT_LOC_RANGE_ADHOC	RSP	DT_Required	-YYYYMMDD -YYYYMMDD -1Y	-YYYYMM...	None	<input checked="" type="checkbox"/>	0
<input checked="" type="radio"/>	DT_PROD_LOC_RANGE_PROCESS_ADHOC	DT_PROD_LOC_RANGE_ADHOC	RSP	DT_Required	null	null	None	<input type="checkbox"/>	0
<input checked="" type="radio"/>	DT_PROD_LOC_RANGE_SETUP_ADHOC_JC	DT_PROD_LOC_RANGE_ADHOC	RSP	DT_Required	-YYYYMMDD -YYYYMMDD -1Y	-YYYYMM...	None	<input type="checkbox"/>	0
<input checked="" type="radio"/>	RSE_BATCH_DISABLE_ADHOC_JOB	RSE_BATCH_DISABLE_ADHOC	RSP	MBA_Batc...	null	null	Failed	<input checked="" type="checkbox"/>	2
<input checked="" type="radio"/>	RSE_BATCH_ENABLE_ADHOC_JOB	RSE_BATCH_ENABLE_ADHOC	RSP	MBA_Batc...	null	null	Failed	<input type="checkbox"/>	0
<input checked="" type="radio"/>	RSE_CUST_ATTR_LOY_PROCESS_ADHOC	RSE_CUST_ATTR_LOY_ADHOC	RSP	CS_CustM...	null	null	None	<input type="checkbox"/>	0
<input checked="" type="radio"/>	RSE_CUST_ATTR_LOY_SETUP_ADHOC_JOI	RSE_CUST_ATTR_LOY_ADHOC	RSP	CS_CustM...	-YYYYMMDD -YYYYMMDD -1Y	-YYYYMM...	None	<input checked="" type="checkbox"/>	0
<input checked="" type="radio"/>	RSE_CUST_ENG_METRIC_BASE_PROCESS	RSE_CUST_ENG_METRIC_BASE_ADHOC	RSP	CS_CustM...	null	null	None	<input checked="" type="checkbox"/>	0
<input checked="" type="radio"/>	RSE_CUST_ENG_METRIC_BASE_SETUP_AC	RSE_CUST_ENG_METRIC_BASE_ADHOC	RSP	CS_CustM...	-YYYYMMDD -YYYYMMDD -1Y	-YYYYMM...	None	<input checked="" type="checkbox"/>	0
<input checked="" type="radio"/>	RSE_CUST_ENG_METRIC_CALC_PROCESS	RSE_CUST_ENG_METRIC_CALC_ADHOC	RSP	CS_CustM...	null	null	None	<input type="checkbox"/>	0
<input checked="" type="radio"/>	RSE_CUST_ENG_METRIC_CALC_SETUP_AC	RSE_CUST_ENG_METRIC_CALC_ADHOC	RSP	CS_CustM...	-YYYYMMDD -YYYYMMDD -1Y	-YYYYMM...	None	<input type="checkbox"/>	0
<input checked="" type="radio"/>	RSE_FAKE_CUST_PROCESS_ADHOC_JOB	RSE_FAKE_CUST_ADHOC	RSP	DT_TXN, C...	null	null	None	<input type="checkbox"/>	0

2. Select one of the Cycle - Nightly/Recurring/Standalone
3. Select the **Process/Job** combination and click Edit from table action menu to open the popup below.

Figure 3-4 Edit Job Dialog

4. Set the **External Status Update Mode** one of the following values:
 - **ALL** - POM will send a status update to the external system for each job's execution regardless of success or failure.
 - **FAILED** - POM will notify only for when this job fails.
 - **NONE** - No status update will be sent by POM for this job.

Payload Specification

Attribute	Description
processName	Name of the root process in a given cycle/flow Note: Process names in the callback response are prefixed with the name of the schedule. For instance, a callback response sent for Process "P1" would have processName attribute as MERCH_P1 on return.

Attribute	Description
processExecutionId	Unique identifier generated by POM to track the process executions.
activityName	Name of the job for which the callback/status update is sent.
activityExecutionId	Unique identifier generated by POM to track the job run instance.
callerId	Identifier provided by the caller to POM when submitting the invocation/execution request. POM returns the same ID back to caller.
correlationId	Identifier provided by the caller to POM when submitting the invocation/execution request. POM returns the same ID back to caller
callBackServiceDataDetail.<KeyName>	Key value pairs supplied to POM when submitting the invocation/execution request. Those are returned back to the caller
failedActivity	In the case where the callback is for a failed job, this field is populated with the details of the failed job.
status	Status of the job execution - <ul style="list-style-type: none"> • COMPLETED • SKIPPED • ERROR • SKIPPED_ON_ERROR
activityStatus	Status of the job, and the derived activity state: <ul style="list-style-type: none"> • ERROR - ACTIVITY_FAILED • COMPLETED - ACTIVITY_COMPLETED • SKIPPED - ACTIVITY_COMPLETED • SKIPPED_ON_ERROR - ACTIVITY_COMPLETED

Payload Examples

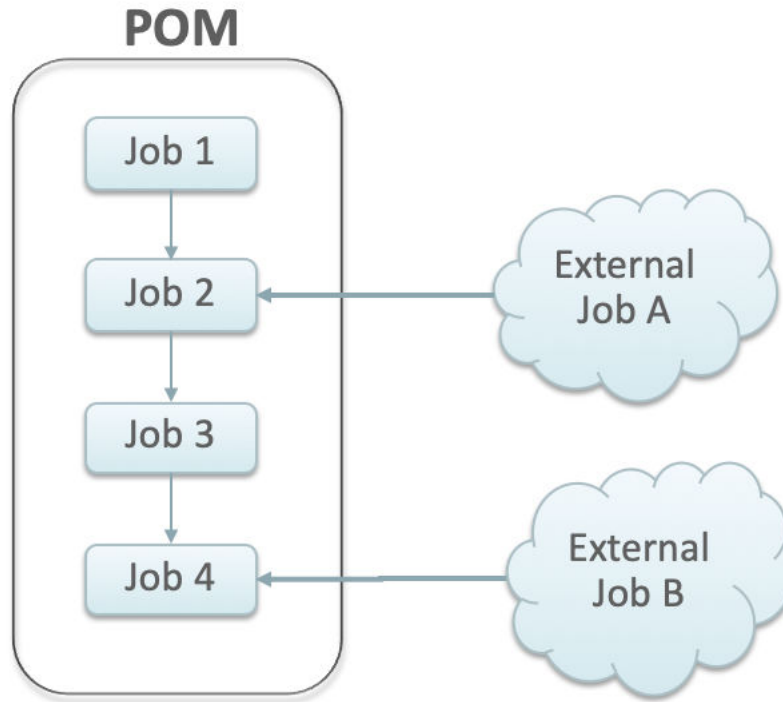
Below are sample external status update payloads for the MERCH schedule.

Description	Payload
Hourly Job Callback	<pre> { " callerId " : "XXX", " correlationId " : "37", " processName " : "MERCH_ BATCH_PROCESS_01", " processExecutionId " : "MERCH_ BATCH_PROCESS_01 ~asfasfdasdfas", " activityName " : " MERCH_ BATCH_JOB", " activityExecutionId " : "123456", " status " : "COMPLETED", " activityStatus " : "ACTIVITY_COMPLETED", " failedActivity " : null } </pre>
Nightly Job Callback	<pre> { " callerId " : "XXX", " correlationId " : "37", " processName " : "MERCH_ START_NIGHT_BATCH_PROCESS ", " processExecutionId " : "MERCH_ START_NIGHT_BATCH_PROCESS ~asfasfdasdfas", " activityName " : " MERCH_ START_NIGHT_BATCH_JOB", " activityExecutionId " : "123456", " status " : "COMPLETED", " activityStatus " : "ACTIVITY_COMPLETED", " failedActivity " : null } </pre>

External Dependency

This feature allows customers to control the execution of a schedule running in POM by defining custom pre-dependencies. POM pauses the schedule execution on encountering these external pre-dependences and resumes the execution once they are released by customer.

Figure 3-5 External Dependency



Configuration

This section details the steps involved in setting up the external dependency.

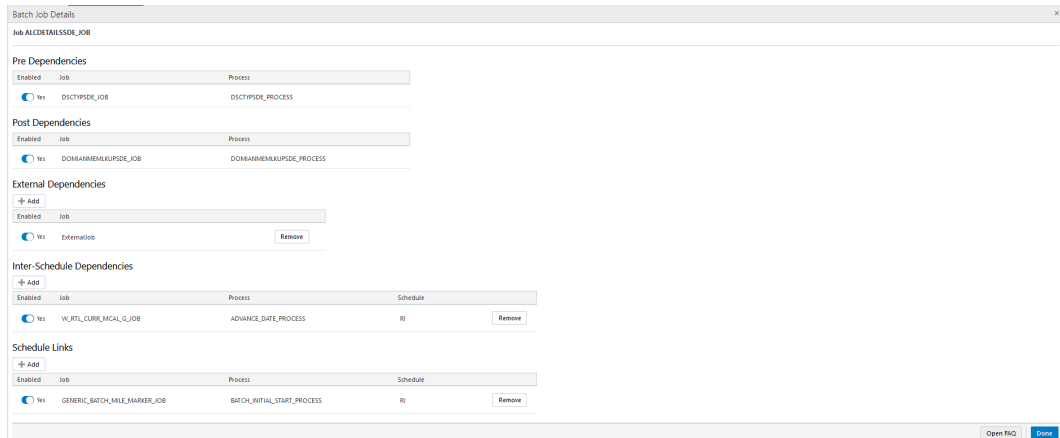
1. Navigate to the Batch Administration screen and select the schedule to which the external dependency will be added.

Figure 3-6 Batch Administration Screen

Enabled	Job	Process Name	Application	Module	Parameters	Active Parameter	External Status Update	Skip On Error	Thre Time
<input checked="" type="radio"/>	AC_PROD_ATTR_LOC_SHARE_PROCESS_A	AC_PROD_ATTR_LOC_SHARE_ADHOC	RSP	CS_Attr_S...	null	null	None	✓	0
<input type="radio"/>	AC_PROD_ATTR_LOC_SHARE_SETUP_ADH	AC_PROD_ATTR_LOC_SHARE_ADHOC	RSP	CS_Attr_S...	-YYYYMMDD-eYYYYMMDD-fY	-YYYYMM...	None	✓	0
<input type="radio"/>	DT_LOC_RANGE_PROCESS_ADHOC_JOB	DT_LOC_RANGE_ADHOC	RSP	DT_Required	null	null	None	✓	0
<input checked="" type="radio"/>	DT_LOC_RANGE_SETUP_ADHOC_JOB	DT_LOC_RANGE_ADHOC	RSP	DT_Required	-YYYYMMDD-eYYYYMMDD-fY	-YYYYMM...	None	✓	0
<input checked="" type="radio"/>	DT_PROD_LOC_RANGE_PROCESS_ADHOC	DT_PROD_LOC_RANGE_ADHOC	RSP	DT_Required	null	null	None	✓	0
<input checked="" type="radio"/>	DT_PROD_LOC_RANGE_SETUP_ADHOC_IC	DT_PROD_LOC_RANGE_ADHOC	RSP	DT_Required	-YYYYMMDD-eYYYYMMDD-fY	-YYYYMM...	None	✓	0
<input checked="" type="radio"/>	RSE_BATCH_DISABLE_ADHOC_JOB	RSE_BATCH_DISABLE_ADHOC	RSP	MBA_Batc...	null	null	Failed	✓	2
<input checked="" type="radio"/>	RSE_BATCH_ENABLE_ADHOC_JOB	RSE_BATCH_ENABLE_ADHOC	RSP	MBA_Batc...	null	null	Failed	✓	0
<input checked="" type="radio"/>	RSE_CUST_ATTR_LOY_PROCESS_ADHOC...	RSE_CUST_ATTR_LOY_ADHOC	RSP	CS_CustM...	null	null	None	✓	0
<input checked="" type="radio"/>	RSE_CUST_ATTR_LOY_SETUP_ADHOC_JOI	RSE_CUST_ATTR_LOY_ADHOC	RSP	CS_CustM...	-YYYYMMDD-eYYYYMMDD-fY	-YYYYMM...	None	✓	0
<input checked="" type="radio"/>	RSE_CUST_ENG_METRIC_BASE_PROCESS	RSE_CUST_ENG_METRIC_BASE_ADHOC	RSP	CS_CustM...	null	null	None	✓	0
<input checked="" type="radio"/>	RSE_CUST_ENG_METRIC_BASE_SETUP_AD	RSE_CUST_ENG_METRIC_BASE_ADHOC	RSP	CS_CustM...	-YYYYMMDD-eYYYYMMDD-fY	-YYYYMM...	None	✓	0
<input checked="" type="radio"/>	RSE_CUST_ENG_METRIC_CALC_PROCESS	RSE_CUST_ENG_METRIC_CALC_ADHOC	RSP	CS_CustM...	null	null	None	✓	0
<input checked="" type="radio"/>	RSE_CUST_ENG_METRIC_CALC_SETUP_AC	RSE_CUST_ENG_METRIC_CALC_ADHOC	RSP	CS_CustM...	-YYYYMMDD-eYYYYMMDD-fY	-YYYYMM...	None	✓	0
<input checked="" type="radio"/>	RSE_FAKE_CUST_PROCESS_ADHOC_JOB	RSE_FAKE_CUST_ADHOC	RSP	DT_TXN, C...	null	null	None	✓	0

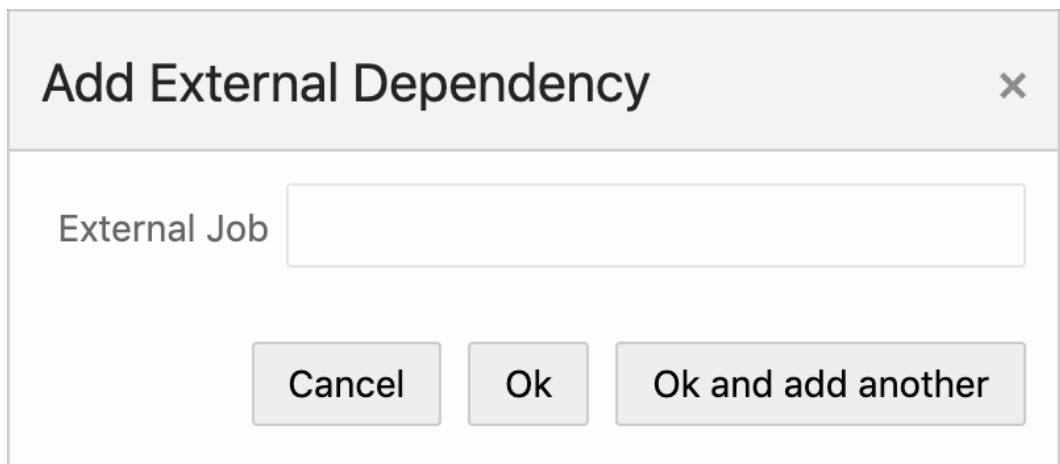
2. Select a Cycle - Nightly/Recurring/Standalone.
3. Find the **Process/Job** combination to which dependency needs to be added.
4. Click on the **Job** name to open the Batch Job Details panel.

Figure 3-7 Batch Job Details



5. On the External Dependency section of the screen, click the **Add** button to create the external dependency.
6. Provide the external job name.

Figure 3-8 Add External Dependency



7. Click **Ok** to save and exit or **Ok and add another** to create another dependency.

Releasing Dependency

External systems need to invoke the following POM ReST service to release/fulfill the external dependencies.

HTTP Method	POST
URL	http://<pom-server-host>/ProcessServices/ services/private/schedules/ <Schedule_Name>/external/jobs/ <Ext_Dependency_Name>/status/ COMPLETED Note: <pom-server-host>, <Schedule_Name> and <Ext_Dependency_Name> variables need to be changed according to the target environment and schedule.
HTTP Headers	Content-Type = application/json Authorization = Bearer <<OAuth_Token_FROM_IDCS_or_OCI_IAM>>
Request Body	None
Response Body	<pre>{ "value": "true", "links": [], "hyperMediaContent" : { "linkRDO": []} }</pre> <p>The true/false in the value attribute shown above indicates the success/failure of releasing the external dependency respectively.</p>

4

Schedule Customization

All retail applications that run batch cycles in POM maintain a standard batch schedule with full list of jobs, pre-defined dependencies, job parameters, and so on.

As a part of the implementation, customers can customize those settings based on their needs.

POM supports modifying the following attributes of a schedule:

- Enable/Disable a job based on the functional use-case
- Modify the job parameters
- Modify the job dependencies
- Enable/Disable the hourly cycles

Customizations can be set up in the Batch Administration screen or through the Export/Import feature of POM (see [Export/Import Schedule Configuration](#)). If the plan is to set up a new non-Oracle-provided schedule, see [Custom Schedules](#).

5

Custom Schedules

This chapter describes how a user sets up a custom batch schedule in POM.

Introduction

As described in [Schedule Customization](#), POM comes with a set of standard retail application batch schedules such as Merchandising and RI. Each schedule is defined in a spreadsheet with multiple tabs that each detail the components of that schedule, such as the Process tab, Job tab, ProcessJobMapping tab, Dependency tab, and so on. For each retail application a customer has subscribed to, a number of steps are performed by either a system integrator or the customer themselves. These steps are to configure and load those schedules, then customize them as described in [Schedule Customization](#).

POM also provides the ability for customers to set up their own custom schedule based on predefined retail application jobs and environment where these are run.

For a user to develop and deploy a custom schedule, the following activities must be completed:

- [Set Up the Schedule Spreadsheet](#)
- [Configure the New Schedule](#)
- [Load the New Schedule](#)
- [Schedule the Batch Tasks](#)
- [Run Batch](#)

The next sections detail these activities.

Set Up the Schedule Spreadsheet

The first step for creating a batch schedule in POM is to fill out the batch schedule spreadsheet using [the template provided](#) with the documentation set and specifications detailed in [Batch Schedule Spreadsheet Explained](#).

Configure the New Schedule

The second step for setting up a batch schedule in POM is to configure the schedule. Refer the “Configure New Schedule” subsection of the “System Configuration” section of the *POM User Guide* for details.

Load the New Schedule

The third step for setting up a batch schedule is to load the schedule into POM. This is accomplished on the Schedule Maintenance screen by selecting this new schedule's tile at the top of the screen, then clicking the **Import Latest Schedule** button. Refer to the “Schedule Maintenance” section of the *POM User Guide* for details.

Schedule the Batch Tasks

The next step for setting up a batch schedule in POM is to schedule the execution of the different cycles. This is accomplished on the Scheduler Administration screen. Refer to the “Scheduler Administration” section of the *POM User Guide* for details.

Run Batch

The final step for setting up a batch schedule in POM is to actually run the batches. For that, open the Batch Monitoring screen, select the schedule tile at the top of the screen, chose the correct date right above the tile section and click the **Create Schedule** button right below the tile section to the right of the screen. Refer to the “Batch Monitoring” section of the *POM User Guide* for details.

6

Generic ReST Jobs

This chapter describes how a user can set up ReST Jobs that can be invoked and tracked by POM.

Introduction

As a Job Scheduler, POM provides the ability for applications to invoke and track different types of Jobs. The table below lists the types of Jobs that are supported in POM by default.

Job Type	Description
EXEC	Refers to shell-script based Jobs. POM is expected to start a shell script and track its execution.
BDI	Indicative of BDI (bulk data) Jobs. POM will invoke a BDI Process and track its execution for these types of Jobs.
RI	Refers to Data Intelligence Services (DIS) Jobs for the RI Schedule. POM invokes custom DIS wrapper endpoints and tracks their execution.
RASE	Refers to Data Intelligence Services (DIS) Jobs for the RSP Schedule. The wrapper endpoints invoked here are the same as used by the RI job type.
RPAS	Indicative of RPAS-WebService Jobs for the RPAS schedule. POM invokes RPAS endpoints to start and track RPAS batches.

 **Note:**

This is the default type of Job. If a Job type has not been specified on the batch schedule spreadsheet, it is assumed to be of this type.

While this list of types meets the needs of Retail Applications that already have their Batch Schedules on POM, it is fixed and lacks extensibility.

To remedy this, POM provides generic interfaces to define and run ReST-based Jobs on external systems that meet certain requirements.

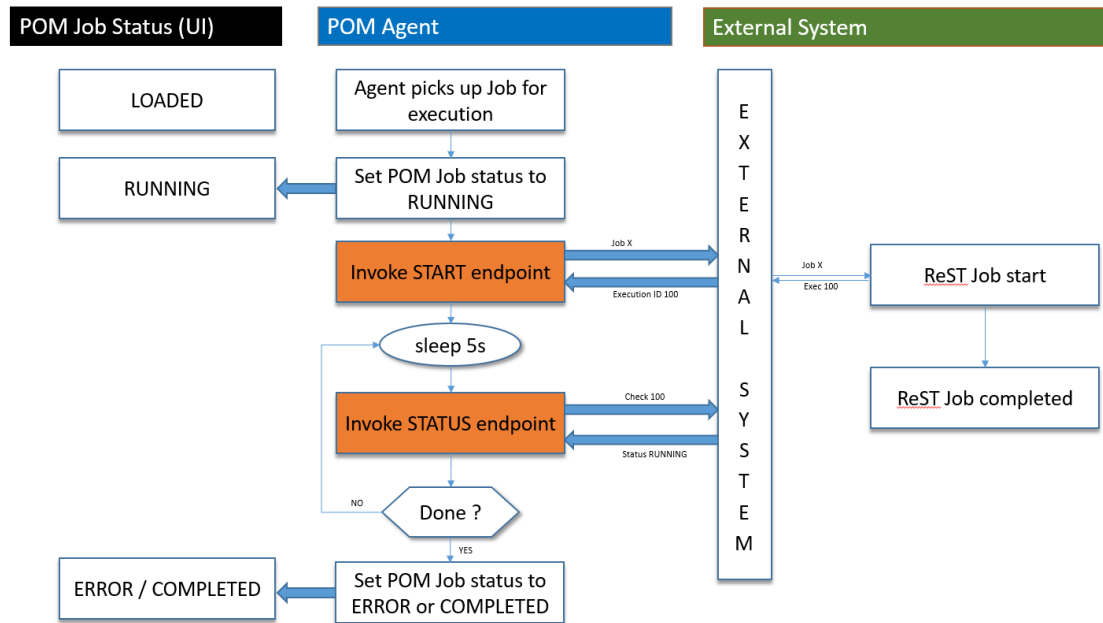
Features of Generic ReST Jobs

To have POM integrate with an external system and to invoke and track Jobs run on that external system, the external system must provide the following capabilities.

#	Ability / ReST endpoints	Description	Mandatory / Optional
1	Start requested Job	Ability to start a Job that was requested and return a unique ID to track its execution. If the Job can't be started or does not exist, throw an exception	Mandatory
2	Check status of Job	Ability to accept a unique execution ID and return a status. In case the execution ID is unknown, throw an exception	Mandatory
3	Restart Job	Ability to restart a Job, given an execution ID of a failed job. Once restarted, a unique execution ID to track the new execution must be returned.	Optional. In cases where the Restart API is not defined, POM uses the Start API (#1)
4	Kill Job	Ability to stop/abort/kill a running Job. This is not meant to be a graceful stop. It should be a hard-stop and not just a request to the Job to stop when possible.	Optional. If a KILL is requested when it's not defined, then POM shows an exception popup.
5	Job Log	Ability to provide runtime logs for the given execution ID.	Optional.
6	Check system status	Ability for POM to check whether the system is up and available. This will show up on the POM Health-check screen.	Optional. In case this is not defined, this will show up in the Health check popup with an error.

Execution Sequence

At runtime, POM invokes generic ReST jobs as depicted below.



- Initially the status of the Job in POM is `LOADED`.
- The Execution Engine sends the request to the Job Agent to run the Job. The Agent Poller picks the Job for execution.
- The Agent sets the status of the Job to `RUNNING`.
- The Start endpoint is invoked on the External System, with the details of the Job to run. The endpoint starts the Job and returns a unique ID to track the execution (100 returned, as shown above).
- The Agent then goes to sleep for 5 seconds (configurable interval).
- It then invokes the Status endpoint on the External System with the execution ID returned previously (in this case 100)
- If a status of `ERROR` or `COMPLETED` is returned, then the Job is assumed to have finished and the status is set on the POM job accordingly.
- In case the status is still `RUNNING`, the Agent then goes back to sleep for 5 seconds and thereafter polls the status again. It does so, until a terminal status of `ERROR` or `COMPLETED` is received.

Handling Restarts

A restart is handled in almost the same sequence as shown above. Except, instead of invoking the Start endpoint, the Restart endpoint will be used with the previously failed execution ID.

In case the Restart endpoint has not been defined, then the Start endpoint will be used.

Handling Kills

In case of killing/aborting a `RUNNING` Job, the Kill endpoint is invoked. A successful response from this endpoint will let POM perform its cleanup and also mark the Job with a status of `ERROR`.

In case the Kill endpoint has not been defined for a Custom Job Type, then invoking the Kill from the POM UI for a running Job of that type will simply show a popup stating the Kill is not defined for the selected Job type.

Defining Custom Job Types

Generic ReST Jobs are defined in POM as Custom Job Types. A class of Generic ReST Jobs is represented in POM as a single Custom Job Type.

All Jobs in POM today have a Job Type attribute. For Generic ReST Jobs, this Job Type will be a Custom Job Type that will encapsulate all the necessary information needed about the destination system.

Batch Schedule Spreadsheet

Custom Job Types can be defined through the **Job Types** tab of the Batch Schedule spreadsheet. Custom Job Types that are specified on the spreadsheet get seeded into POM when the Batch Schedule is loaded. See [Batch Schedule Spreadsheet Explained](#) for more information.

POM UI

The Administrator can also maintain Custom Job Types in the **Job Types** section of the **System Configuration** screen. Refer to the “Edit Job Type” subsection of the “System Configuration” section of the *POM User Guide*.

New Custom Job Types can be added there as well.

Environment-specific Information

The only aspects of a Custom Job Type that will change based on its deployment are its URL and its OAuth Scope.

Today teams use an endpoint to configure a new Schedule in POM. This endpoint has been enhanced to accept the URL and the OAuth scope needed for a Custom Job Type. Refer to the “Configure New Schedule” subsection of the “System Configuration” section in the *POM User Guide* for further information.

Internal Representation

A JSON representation of a Custom Job Type stored within POM looks like the following:

```

{
  "type" : "MOCKTYPE",
  "scheduleName" : "MERCH",
  "url" : "http://pomagentservice:8080/POMAgent/services/private/jobTypes/mockType",
  "actions" : [
    {
      "action" : "JOB_START",
      "resourcePath" : "/start"
    },
    {
      "action" : "JOB_RESTART",
      "resourcePath" : "/restart"
    },
    {
      "action" : "JOB_STATUS",
      "resourcePath" : "/status"
    },
    {
      "action" : "JOB_KILL",
      "resourcePath" : "/kill"
    },
    {
      "action" : "JOB_LOG",
      "resourcePath" : "/log"
    },
    {
      "action" : "VALIDATION",
      "resourcePath" : "/validation"
    }
  ],
  "scopes" : "rgbu:pom:agent-administrator-STG1000"
}

```

In this case the Custom Job Type is named `MOCKTYPE`. All Jobs of this type would invoke the endpoints defined above.

Job Parameter Restrictions

The definition of parameters to Generic ReST Jobs in POM is restricted to a double-pipe-delimited string of key value pairs.

For example: `key=value||key1=value1||key2=value2`

Custom Job Type Endpoint Specifications

Authentication

POM only supports the OAuth System Credential Grant mode of authentication. No other types of authentication, such as Basic Auth, are supported.

The OAuth scope stored for each Custom Job Type informs POM about which scopes are to be used when invoking the Custom Job Type APIs.

Job Start API

HTTP Method	POST
--------------------	------

Request Payload (JSON)

Attribute	Description
cycleName	POM Batch Cycle
flowName	POM Batch Flow
processName	POM Batch process name
jobName	POM Batch job name
parameters	Job Parameters (Double-pipe delimited key-value pairs) as defined in POM
agentExecutionId	A unique ID assigned by POM for every job run

Authentication

OAuth System Credentials Grant

Response Payload (JSON)

In case of successful invocations, the response will contain the following:

Attribute	Description
executionId	Unique ID returned by the target app to POM for status tracking
status	Any status
executionInfo (optional)	Any additional info the target app would like to share with POM

In case of any issues while submitting the Job, a 500 exception response must be thrown.

Job Status API

HTTP Method

GET

Query Parameters (JSON)

Attribute	Description
processName	POM Batch process name
jobName	POM Batch job name
executionId	Unique Id returned by the target app to POM for status tracking

Authentication

OAuth System Credentials Grant

Response Payload (JSON)

In case of a successful invocation, the response will contain the following:

Attribute	Description
executionId	Unique Id returned by the target app to POM for status tracking
status	RUNNING / ERROR / COMPLETED.
executionInfo (optional)	Any additional info the target app would like to share with POM.

In case of any issues while submitting the Job, a 500 exception response must be thrown.

Job Restart API

HTTP Method	POST																
Request Payload (JSON)	<table border="1"> <thead> <tr> <th>Attribute</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>cycleName</td> <td>POM Batch Cycle</td> </tr> <tr> <td>flowName</td> <td>POM Batch Flow.</td> </tr> <tr> <td>processName</td> <td>POM Batch process name</td> </tr> <tr> <td>jobName</td> <td>POM Batch job name</td> </tr> <tr> <td>parameters</td> <td>Job Parameters (Double-pipe delimited key-value pairs) as defined in POM</td> </tr> <tr> <td>agentExecutionId</td> <td>A unique ID assigned by POM for every job run</td> </tr> <tr> <td>executionId</td> <td>Unique Id of the previously failed execution returned by the target app to POM for status tracking</td> </tr> </tbody> </table>	Attribute	Description	cycleName	POM Batch Cycle	flowName	POM Batch Flow.	processName	POM Batch process name	jobName	POM Batch job name	parameters	Job Parameters (Double-pipe delimited key-value pairs) as defined in POM	agentExecutionId	A unique ID assigned by POM for every job run	executionId	Unique Id of the previously failed execution returned by the target app to POM for status tracking
Attribute	Description																
cycleName	POM Batch Cycle																
flowName	POM Batch Flow.																
processName	POM Batch process name																
jobName	POM Batch job name																
parameters	Job Parameters (Double-pipe delimited key-value pairs) as defined in POM																
agentExecutionId	A unique ID assigned by POM for every job run																
executionId	Unique Id of the previously failed execution returned by the target app to POM for status tracking																
Authentication	OAuth System Credentials Grant																
Response Payload (JSON)	In case of a successful invocation, the response will contain the following: <table border="1"> <thead> <tr> <th>Attribute</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>executionId</td> <td>Unique Id returned by the target app to POM for status tracking</td> </tr> <tr> <td>status</td> <td>Any status</td> </tr> <tr> <td>executionInfo (optional)</td> <td>Any additional info the target app would like to share with POM.</td> </tr> </tbody> </table> <p>In case of any issues while restarting the Job, a 500 exception response must be thrown.</p>	Attribute	Description	executionId	Unique Id returned by the target app to POM for status tracking	status	Any status	executionInfo (optional)	Any additional info the target app would like to share with POM.								
Attribute	Description																
executionId	Unique Id returned by the target app to POM for status tracking																
status	Any status																
executionInfo (optional)	Any additional info the target app would like to share with POM.																

Job Kill API

HTTP Method	POST								
Request Payload (JSON)	<table border="1"> <thead> <tr> <th>Attribute</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>processName</td> <td>POM Batch process name</td> </tr> <tr> <td>jobName</td> <td>POM Batch job name</td> </tr> <tr> <td>executionId</td> <td>Unique ID of the previously failed execution returned by the target app to POM for status tracking</td> </tr> </tbody> </table>	Attribute	Description	processName	POM Batch process name	jobName	POM Batch job name	executionId	Unique ID of the previously failed execution returned by the target app to POM for status tracking
Attribute	Description								
processName	POM Batch process name								
jobName	POM Batch job name								
executionId	Unique ID of the previously failed execution returned by the target app to POM for status tracking								
Authentication	OAuth System Credentials Grant								

Response Payload (JSON) In case of successful invocations, the response would contain the following

Attribute	Description
executionId (optional)	Unique ID returned by the target app to POM for status tracking
status (optional)	Any status
executionInfo (optional)	Any additional info the target app would like to share with POM.

In case of Job cannot be killed, a 500 exception response must be thrown.

Job Log API

HTTP Method	GET								
Query Parameters (JSON)	<table border="1"> <thead> <tr> <th>Attribute</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>processName</td> <td>POM Batch process name</td> </tr> <tr> <td>jobName</td> <td>POM Batch job name</td> </tr> <tr> <td>executionId</td> <td>Unique ID returned by the target app to POM for status tracking</td> </tr> </tbody> </table>	Attribute	Description	processName	POM Batch process name	jobName	POM Batch job name	executionId	Unique ID returned by the target app to POM for status tracking
Attribute	Description								
processName	POM Batch process name								
jobName	POM Batch job name								
executionId	Unique ID returned by the target app to POM for status tracking								
Authentication	OAuth System Credentials Grant								
Response Payload (JSON)	<p>In case of a successful invocation, the response will contain the log data for the execution provided. This log will be viewable in the log downloaded from the POM UI.</p> <p>In case of any issues while fetching the Job log, a 500 exception response must be thrown.</p>								

Job System Check API

HTTP Method	GET
Authentication	OAuth System Credentials Grant
Response Payload (JSON)	<p>A success response of 200 or 204 indicates that the system is up and running.</p> <p>Any exception while calling this endpoint, should be interpreted as the system being down.</p>

7

Export/Import Schedule Configuration

This chapter explains the export/import schedule configuration feature of POM.

Overview

POM provides a feature, in the user interface, to export and import schedule configuration data such as data retention limits, throttle limits, enable/disable flags, schedule times, notifications configuration, and so on.

This feature can be used in two ways:

1. Data exported on a specific environment can be imported back on a different environment without any modifications. This is a typical use case where the batch schedule on a certain environment, such as Stage, is set up and fine-tuned according to a customer's requirements. This includes specifying which jobs should or shouldn't run. It also includes specifying the times certain cycles or flows will start. It also includes specifying the e-mail addresses that will receive certain notifications and how long to keep those notification in the system before purging them.

Once the schedule is configured and fine-tuned (on Stage, for example) the export/import feature can be used to export the configurations from Stage and import them back into production.

Note:

The user has the option to export to a spreadsheet or a JSON file. In this first use case, it is recommended the user exports to a JSON file then imports it back into the other environment.

2. Data can be altered in the exported spreadsheet then imported back into the same environment. This is useful at provisioning time where an environment is first set up with the default batch schedule configuration. A user would then export the default configuration, then modify that configuration on the spreadsheet to conform to the customer's needs. The spreadsheet is then imported back into the same environment, applying the desired configurations.

Note:

When importing the spreadsheet on the Batch Administration screen, the user will have the option of importing job-related configuration or scheduler-related configuration (run times) or both. Refer to the "Batch Administration" section in the *POM User Guide* for more details.

This chapter describes each tab in the exported spreadsheet, along with the data on each sheet. It also indicates which fields can be modified.

Schedule Info Tab

This is an informational only tab. No fields are modifiable on it.

Table 7-1 Schedule Info

Field	Description	Modifiable?
Name	Schedule or application name.	No
Version	Schedule version.	No
Installation Date	Date the schedule was uploaded into POM.	No
Customer Name	This is used on notifications to identify which customer a notification is for.	No
Environment Name	This is used on notifications to identify which customer's environment a notification is for.	No

Schedule Configuration Tab

This tab contains schedule level settings.

Table 7-2 Schedule Configuration

Field	Description	Modifiable?
First Run Business Date	Date when batch was run for the first time. See subsection "Business Date Explained" in the "System Configuration" section of the <i>POM User Guide</i> .	No
Data Retention Days	Number of days historic data is to be retained in POM.	Yes - Recommended value is 30 days.
Long Run Average Multiplier	Number which is multiplied by a job's average run time or Threshold Runtime (if provided) to determine the threshold which, when exceeded, the job is deemed long running.	Yes - Usually a number between one and three.
External Dependencies	Indicates whether POM will respect external dependencies or not. They are usually dependencies on customers' internal processes.	Yes - Valid values are Y or N.
Inter-schedule Dependencies	Indicates whether POM will respect external dependencies or not. Those are dependencies on other schedules.	Yes - Valid values are Y or N.

Table 7-2 (Cont.) Schedule Configuration

Field	Description	Modifiable?
Callback mode	This is also known as External Status Update Mode. Depending on the value, the customer's system is notified of success or failure of any job in the entire schedule. The value set here is overridden by this same setting for any individual job defined on the other job-specific tabs.	Yes - Valid values are ALL, FAILED or NONE. When value is NONE, no callback is made. When value is ALL, callback is made on either success or failure. When value is FAILED, callback is made only on failure

Throttling Tab

At this time, this tab should only be used to set throttle limits at the application level. Throttling is a technique used to limit the number of jobs that can run concurrently for a specific application. Throttling limits are set so a server's resources are not overwhelmed by too many concurrently running jobs.



Note:

Previously, throttle limits could be set at the application level and/or at the module level. Now they can only be set at the application level.

Also note that this tab was intended to provide the capability to enable/disable whole applications or modules. This is not functioning at this time. In order to achieve this objective, filter on application on the Nightly Jobs Configuration, Recurring Jobs Configuration or Adhoc Jobs Configuration, then change all enabled flags to 'Y' or 'N' as desired. For enabling/disabling whole modules, this has to be done on the Batch Administration screen at this time.

(Continued)

Table 7-3 Throttling Configuration

Field	Description	Modifiable? - Possible values
Application	Application code.	No
Module	Module name.	No
Job Admin	Job Admin is the component usually running on the same server as the application and which executes that application's batch jobs.	No
Throttle Limit	The max number of jobs that can run concurrently for the specified application.	Yes - A number that is greater than zero.

Table 7-3 (Cont.) Throttling Configuration

Field	Description	Modifiable? - Possible values
Enabled	This flag currently does not enable or disable jobs belonging to an application/module. See the second note at the top of this section for more information.	Yes - Valid values are Y or N.

Nightly Jobs Configuration Tab

This tab contains all Nightly cycle's jobs along with their configuration.

Table 7-4 Nightly Jobs Configuration

Field	Description	Modifiable? - Possible values
Job	Job name.	No
Process	Process name to which the job belongs; a process can contain multiple jobs.	No
Cycle	This is a fixed value for this tab of 'Nightly'.	No
Application	Application name to which this process-job belongs.	No
Initial Parameter	The initial parameter this job was set up with when the base schedule was first loaded into POM.	Yes - Can contain blank or any space-separated list of parameters.
Parameter Change	Flag indicating whether the parameter can be changed after the initial load. If it can, then the new value must be loaded in the next field: Active Parameter.	Yes - Valid values are Y or N.
Active Parameter	Parameter which overrides the Initial Parameter when Parameter Change is 'Y'.	Yes - Can contain blank or any space-separated list of parameters.
Skip on error	Flag indicating whether this job can be skipped when it fails.	Yes - Valid values are Y or N.
Callback mode	This is also known as External Status Update Mode. Depending on the value, the customer's system is notified of success or failure of this specific job by calling a predefined customer endpoint. The value set here overrides that of this same setting on the Schedule Configuration tab.	Yes - Valid values are ALL, FAILED or NONE. When value is NONE, no callback is made. When value is ALL, callback is made on either success or failure. When value is FAILED, callback is made only on failure

Table 7-4 (Cont.) Nightly Jobs Configuration

Field	Description	Modifiable? - Possible values
Day of week	Contains the day(s) of the week on which this specific Job will run. POM automatically skips these Jobs on remaining days. If this field is left blank, the job will run on every day of the week.	Yes - Valid values are blank or any number of comma separated days of the week (for example: SUNDAY,MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY)
Threshold Runtime	Estimated runtime in seconds for this specific job. This is an optional field which, when entered, will be used to multiply by the Long Run Average Multiplier defined on the Schedule Configuration tab. When the resulting number is exceeded, the job is deemed long running. If this field is blank, then the job's average run time is used instead to multiply by the Long Run Average Multiplier.	Yes - Valid values are blank or an integer number of seconds.
Enabled	Flag indicating if this job is enabled or disabled	Yes - Valid values are Y or N.
Notify Job Start	Flag indicating if a notification is to be sent at start of this job.	Yes - Valid values are Y or N.
Notify Job Completion	Flag indicating if a notification is to be sent at successful completion of this job.	Yes - Valid values are Y or N.

Recurring Jobs Configuration Tab

This tab contains the list of Recurring cycle/Process/Job along with their configuration.

Table 7-5 Recurring Jobs Configuration

Field	Description	Modifiable? - Possible values
Job	Job Name	No
Process	Process name to which the job belongs; a process can contain multiple jobs.	No
Cycle	Name of the hourly/recurring cycle this job belongs to.	No
Application	Application name to which this process-job belongs.	No
Initial Parameter	The initial parameter this job was set up with when the base schedule was first loaded into POM.	Yes - Can contain blank or any space-separated list of parameters.

Table 7-5 (Cont.) Recurring Jobs Configuration

Field	Description	Modifiable? - Possible values
Parameter Change	Flag indicating whether the parameter can be changed after the initial load. If it can, then the new value must be loaded in the next field: Active Parameter.	Yes - Valid values are Y or N.
Active Parameter	Parameter which overrides the Initial Parameter when Parameter Change is 'Y'.	Yes - Can contain blank or any space separated list of parameters.
Skip on error	Flag indicating whether this job can be skipped when it fails.	Yes - Valid values are Y or N.
Callback mode	This is also known as External Status Update Mode. Depending on the value, customer's system is notified of success or failure of this specific job by calling a predefined customer endpoint. The value set here overrides that of this same setting on the Schedule Configuration tab.	Yes - Valid values are ALL, FAILED or NONE. When value is NONE, no callback is made. When value is ALL, callback is made on either success or failure. When value is FAILED, callback is made only on failure
Day of week	Contains the day(s) of the week on which this specific Job will run. POM automatically skips these Jobs on the remaining days. If this field is left blank, the job will run on every day of the week	Yes - Valid values are blank or any number of comma-separated days of the week (for example: SUNDAY,MONDAY,TUESDAY, WEDNESDAY,THURSDAY, FRIDAY,SATURDAY)
Threshold Runtime	Estimated runtime in seconds for this specific job. This is an optional field which, when entered, will be used to multiply by the Long Run Average Multiplier defined on the Schedule Configuration tab. When the resulting number is exceeded, the job is deemed long running. If this field is blank, then the job's calculated average run time is used instead to multiply by the Long Run Average Multiplier.	Yes - Valid values are blank or an integer number of seconds.
Enabled	Flag indicating whether this job is enabled or disabled.	Yes - Valid values are Y or N.
Notify Job Start	Flag indicating if a notification is to be sent at start of this job.	Yes - Valid values are Y or N.
Notify Job Completion	Flag indicating if a notification is to be sent at successful completion of this job.	Yes - Valid values are Y or N.

Flows Configuration Tab

This tab contains a flow/cycle matrix. It contains a row for each flow and all recurring cycles as columns. An intersection of 'Y' means the given flow is defined to run as part of the given recurring cycle. It is recommended that all recurring cycles are enabled

but not necessarily scheduled for running. See the Scheduling Flows Tab for further clarification.

Table 7-6 Flows Configuration

Field	Description	Modifiable?
Flow	Flow name.	No
Recurring cycle 1	A 'Y' or 'N' value indicating whether the given flow is run as part of recurring cycle 1. A value of N does not disable jobs belonging to the flow/cycle intersection. Those jobs need to be manually disabled on the Batch Administration screen.	Yes - Valid values are Y or N.
Recurring cycle 2	A 'Y' or 'N' value indicating whether the given flow is run as part of recurring cycle 2. A value of N does not disable jobs belonging to the flow/cycle intersection. Those jobs need to be manually disabled on the Batch Administration screen.	Yes - Valid values are Y or N.
Recurring cycle n	A 'Y' or 'N' value indicating whether the given flow is run as part of recurring cycle n. A value of N does not disable jobs belonging to the flow/cycle intersection. Those jobs need to be manually disabled on the Batch Administration screen.	Yes - Valid values are Y or N.

Adhoc Jobs Configuration Tab

This tab contains the list of ad hoc jobs (also known as standalone) along with their configuration.

Table 7-7 Adhoc Jobs Configuration

Field	Description	Modifiable? - Possible values
Job	Job name.	No
Process	Process name to which the job belongs; a process can contain multiple jobs.	No
Cycle	This is a fixed value for this tab of 'Adhoc'.	No
Application	Application name to which this process-job belongs.	No
Initial Parameter	The initial parameter this job was set up with when the base schedule was first loaded into POM.	Yes - Can contain blank or any space-separated list of parameters.

Table 7-7 (Cont.) Adhoc Jobs Configuration

Field	Description	Modifiable? - Possible values
Parameter Change	Flag indicating whether the parameter can be changed after the initial load. If it can, then the new value must be loaded in the next field: Active Parameter.	Yes - Valid values are Y or N.
Active Parameter	Parameter which overrides the Initial Parameter when Parameter Change is 'Y'.	Yes - Can contain blank or any space separated list of parameters.
Skip on error	Flag indicating whether this job can be skipped when it fails.	Yes - Valid values are Y or N.
Callback mode	This is also known as External Status Update Mode. Depending on the value, the customer's system is notified of the success or failure of this specific job by calling a predefined customer endpoint. The value set here overrides that of this same setting on the Schedule Configuration tab.	Yes - Valid values are ALL, FAILED or NONE. When value is NONE, no callback is made. When value is ALL, callback is made on either success or failure. When value is FAILED, callback is made only on failure
Day of week	Contains the day(s) of the week on which this specific Job will run. POM automatically skips these Jobs on remaining days. If this field is left blank, the job will run on every day of the week	Yes - Valid values are blank or any number of comma separated days of the week (for example: SUNDAY,MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY)
Threshold Runtime	Estimated runtime in seconds for this specific job. This is an optional field which, when entered, will be used to multiply by the Long Run Average Multiplier defined on the Schedule Configuration tab. When the resulting number is exceeded, the job is deemed long running. If this field is blank, then the job's calculated average run time is used instead to multiply by the Long Run Average Multiplier.	Yes - Valid values are blank or an integer number of seconds.
Enabled	Flag indicating whether this job is enabled or disabled.	Yes - Valid values are Y or N.
Notify Job Start	Flag indicating if a notification is to be sent at start of this job.	Yes - Valid values are Y or N.
Notify Job Completion	Flag indicating if a notification is to be sent at successful completion of this job.	Yes - Valid values are Y or N.

Job Dependencies Tab

This tab contains the definitions of all intra-schedule dependencies (or job dependencies within the same schedule) for all cycles (Nightly, Recurring and Adhoc).

External, inter-schedule dependencies and execution links are defined on the Job External Associations tab.

Only the Enabled flag can be changed on this tab. Caution should be exercised when disabling a dependency, as in certain situations this can cause a job to run before data is processed by the predecessor job. This can therefore cause data corruption.

Table 7-8 Adhoc Jobs Configuration

Field	Description	Modifiable? - Possible values
Job	Job name.	No
Process	Process name to which the job belongs.	No
Cycle	Nightly, Adhoc or specific recurring/hourly cycle.	No
Predecessor Process	Process name to which the predecessor job belongs.	No
Predecessor Job	Job which must complete before the job defined on the current row can run.	No
Enabled	Flag indicating whether this dependency is enabled or disabled.	Yes - Valid values are Y or N.

Job External Associations Tab

This tab contains the definitions of External dependencies, inter-schedule dependencies and execution links, so all dependencies other than the intra-schedule dependencies described in the previous section.



Note:

New external associations can be added in this tab.

External dependencies are usually those defined for customer processes. These are associated with a POM endpoint that the customer calls to satisfy the dependency.

Inter-schedule dependencies are those associated with another schedule. For instance, a Retail Insight job can be made to wait for a Merchandising job to complete.

An Execution link is a special dependency that sets up an application's schedule to be invoked based on a completion of a job in another application's schedule. For instance, the Retail Insight schedule can be set up to be invoked when job A completes in the Merchandising schedule.

Table 7-9 Job External Associations

Field	Description	Modifiable? - Possible values
Job	Job name.	Yes - Needs to be a valid job name already defined on the nightly schedule.
Process	Process name to which the job belongs.	Yes - Needs to be a valid process name already defined in the nightly schedule.
Cycle	Cycle name.	Yes - This can only be Nightly.
Association Type	Type of dependency.	Yes - Valid values are Internal for Inter-Schedule, External and EXEC_LINK
External Schedule	Name of the schedule containing the inter-schedule dependency or execution link.	Yes - This must be an existing valid schedule defined in the same instance of POM. This is only required for inter-schedule dependencies and execution links.
External Job	Name of external job.	Yes - In the case of an external dependency, this is the name that will be included in the payload of an endpoint called by the external system, such as the customer's. In the case of inter-schedule dependencies and execution links, this is a valid job name defined in the dependent schedule.
External Process	Process name to which the external job belongs.	Yes - This is required for inter-schedule dependencies and execution links. It's the process name associated with the external job.
Enabled	Flag indicating whether this dependency is enabled or disabled.	Yes - Valid values are Y or N.

Scheduling Flows Tab

This tab and the next contain configurations for scheduling flows or processes to run at specified times.

This Scheduling Flows tab contains the definitions of scheduled run times for the Nightly and recurring flows.

It simply contains one row for the Nightly flow stating the time when the Nightly processes will start every day.

The rest of the tab contains rows for each recurring flow, with columns for each of the hourly cycles making up the flow. For each cell at the intersection of Flow and hourly cycle, the time is entered for when the cycle is to start running. At present, there is a

maximum of 24 hourly cycles that can be defined which, when spaced equally, would run one hour apart. Times are optional, so a blank cell indicates that the cycle will not be scheduled to run.

 **Note:**

Although there is no validation that cross-references the presence of a time in a cell with the enabling of the cycle on the Flows Configuration tab, ideally these would match up. This means that, if you enter a time for running an hourly cycle, the same intersection on the Flows Configuration tab should be 'Y'. However, it is conceivable to enable the cycle on the Flows Configuration tab but not schedule it, leaving the option open to schedule it as needed. In fact, this is the recommended practice: Enable all hourly cycles and only schedule a few as needed, but have the option to run more to catch up when necessary.

The timezone region ID entered in cell B is used for all times entered for all subsequent cycles on a given row. It is possible to specify a different timezone for select time cells (for example, 5:00 America/Chicago).

 **Note:**

If the timezone region ID is null in the database, UTC is exported as the default into cell B of this tab.

Table 7-10 Scheduling Flows

Field	Description	Modifiable? - Possible values
Flow	Flow name.	No
Timezone Region ID	Timezone corresponding to the time entered in the subsequent cell(s). Timezone region ID is required, rather than a UTC offset (such as "UTC-06:00"). This is to accommodate Daylight Saving Time.	Yes - Valid timezone region IDs such as US/Eastern can be found as TZ database name at https://en.wikipedia.org/wiki/List_of_tz_database_time_zones
Nightly	Only fill this cell if Flow is Nightly. Enter the time for starting the Nightly cycle.	Yes - Enter a valid military time format such as 6:00 (6am) or 22:00 (10pm).
Recurring cycle 1	Enter a time in this cell to schedule recurring cycle 1 to run at that time. Leave blank to forego scheduling recurring cycle 1. It is possible to not schedule the cycle by default but elect to run it manually if needed.	Yes - Enter a valid military time format such as 6:00 (6am) or 22:00 (10pm). Conflict with the Nightly batch flow should be avoided as POM will skip running a recurring cycle if its start time is elapsed while Nightly is running.

Table 7-10 (Cont.) Scheduling Flows

Field	Description	Modifiable? - Possible values
Recurring cycle 2	Enter a time in this cell to schedule recurring cycle 2 for running at that time. Leave blank to forego scheduling recurring cycle 2. It is possible to not schedule the cycle by default but elect to run it manually if needed.	Yes - Enter a valid military time format such as 6:00 (6am) or 22:00 (10pm). Conflict with the Nightly batch flow should be avoided as POM will skip running a recurring cycle if its start time is elapsed while Nightly is running.
Recurring cycle n	Enter a time in this cell to schedule recurring cycle 1n for running at that time. Leave blank to forego scheduling recurring cycle n. It is possible to not schedule the cycle by default but elect to run it manually if needed.	Yes - Enter a valid military time format such as 6:00 (6am) or 22:00 (10pm). Conflict with the Nightly batch flow should be avoided as POM will skip running a recurring cycle if its start time is elapsed while Nightly is running.

Scheduling Adhoc Tab

This tab contains the definitions of scheduled run times for the ad hoc processes.

Table 7-11 Scheduling Adhoc

Field	Description	Modifiable? - Possible values
Process	Adhoc process name.	No
Description	Description of reason for running this Adhoc process at the specified time.	Yes - Optionally describe the purpose for scheduling the Adhoc process at the specified time
Frequency	Frequency in minutes at which this Adhoc process is to be run.	Yes - Valid values are Daily meaning run this process once at the specified time, or EVERY:x, where x is number of minutes. For example EVERY:120 means run this process every two hours.
Limit Occurrences	Relevant only when frequency of EVERY:x is used. If a Limit Occurrences value is entered, the process will run a maximum of times equal to the specified limit.	Yes - A positive integer. For example: Frequency of EVERY:120 with a Limit Occurrences of 4 means the process will run every two hours a maximum of four times
Prevent start during nightly	A flag which, when set to Y, indicates that this process is not to be started when the Nightly process is running. If the Scheduler tries to start a run while Nightly is running, the execution's status is set to Error with an Info message of "Nightly started running so can't run the process."	Yes - Valid values are Y or N.

Table 7-11 (Cont.) Scheduling Adhoc

Field	Description	Modifiable? - Possible values
Schedule Time	Entering a time in this field causes the process to be scheduled at that time in case of a DAILY frequency or to start the first run at that time in case of an EVERY:x frequency. Leaving this field blank causes the process to run immediately or the first run to start immediately when the scheduler day starts. If a specific time is entered in this field but the scheduler day starts after that time, the process will be scheduled for the next day at that time.	Yes - Enter a valid military time format such as 6:00 (6am) or 22:00 (10pm).
Enabled	Flag indicating if this process is to be scheduled.	Yes - Valid values are Y or N.
Timezone Region ID	Timezone corresponding to the time entered in the subsequent cell(s). Timezone region ID is required rather than a UTC offset such as UTC-06:00. This is to accommodate Daylight Saving Time.	Yes - Valid timezone region IDs such as US/Eastern can be found as TZ database name at https://en.wikipedia.org/wiki/List_of_tz_database_time_zones

Notification Tab

This tab contains all POM-defined notification types and associated e-mail addresses and retention periods. For a list of notification types, refer to [Notifications](#).

Table 7-12 Notifications Configuration

Field	Description	Modifiable?
Notification Type	Type of notification. There are several events in POM for which notifications are generated. An example of a notification type is: NightlyStart.	No
Email Subscription	Email address to which notifications of this type will be sent.	Yes - Valid values are blank or correctly formed email addresses.
Retention Period	Period in days notifications generated for this type are to be retained in the system before purging..	Yes - Valid values are blank or an integer number of days.

8

Notifications

This chapter provides the list of notifications sent by POM to alert users about events that occur throughout the batch execution.

By default, all notifications are shown on the POM application. It is an option to configure notifications to also send e-mails. This is accomplished through the Notifications Administration function of Retail Home. Refer to the "Notifications Administration" chapter of the *Retail Home Administration Guide* for more information.

1. **Error:** This notification is sent whenever a running batch fails for some reason.
2. **Warning type:** This notification is sent when application/user is trying to run an already running job.
3. **InformationNotification:** This notification is for System information. This notification is for Oracle internal users.
4. **SystemErrorNotification:** This notification is to notify of all unexpected system exceptions. This notification is for Oracle internal users.
5. **New scheduler day failure:** This notification is sent when there is an issue while setting up the new scheduler day, e.g. previous scheduler day has not yet been completed.
6. **Intraday cycle completed:** This is a low priority notification that is sent whenever an intraday cycle is completed.
7. **Intraday Cycle Skipped:** This is a low priority notification that is sent whenever an intraday cycle is skipped.
8. **NightlyStart:** This notification is sent when the Nightly cycle is kicked off.
9. **Nightly cycle completed:** This notification is sent when the nightly batch completes for the scheduler day.
10. **Long running job:** This notification is sent whenever a job is taking more than the configured threshold time for getting completed.
11. **NightlySummaryReport:** This notification sends out a batch summary report to the configured mail addresses when the nightly cycle is completed. This notification is for Oracle internal users.
12. **IntradayCycleSummaryReport:** This notification sends out a batch summary report to the configured mail addresses when an intraday cycle is completed for each flow.
13. **JosSyncFailed:** This notification is sent out when there is an issue in publishing disabled jobs from POM to JOS.

 **Note:**

Disabled jobs are published to JOS as a part of the New Scheduler Day process. If the previous day's nightly cycle is still running at the time of New Scheduler Day creation, then publishing of the disabled jobs to JOS is done when the last job of the nightly cycle completes.

14. **ScheduleChangesSummaryReport:** This notification sends out the Delta Summary Changes Report to the configured mail address. Delta changes include the Jobs added to the current Schedule or Jobs removed or Jobs Status changed from the previous schedule. The External Dependency changes are also shown on the report. If there are no changes between the previous and current schedules, the notification is not sent out.
15. **ExternalDepPending:** This notification is sent when a Job is waiting for and External Dependency.
16. **InterSchedDepPending:** This notification is sent when a Job is waiting for an Inter-Schedule Dependency.
17. **InterSchedDepIssue:** This notification is sent when:
 - The business date between the interdependent Schedules vary by more than a day.
 - The External Schedule is day ahead of the current Schedule and previous day data is not available.
 - The inter-schedule dependencies are not valid and disabled.
 - A schedule includes inter-schedule dependencies or execution links that are not valid.
18. **ApplicationModuleDisabled:** This notification is sent when applications or modules are disabled during MDF synchronization.
19. **ErrorNotificationExternal:** This notification is sent whenever a running batch failed for some reason. This notification is for external customers.
20. **NightlySummaryReportExternal:** This notification sends out a batch summary report to the configured mail addresses when nightly cycle is completed. This notification is for external customers.
21. **ExternalDepComplete:** This notification is sent when an external dependency is completed.
22. **ExecutionLinkIssue:** This notification is sent if there is a failure/warning in the Execution Link invocation.
23. **ApplicationModuleEnabled:** This notification is sent when applications or modules are enabled during MDF synchronization.
24. **ExecutionEngineIssue:** This notification is sent when there are important/critical events/failures with the Execution Engine. For example:
 - When the Job Admin invocation fails or
 - When the request is stuck in SUBMITTING state and not able to auto recover or
 - When the request is stuck in SUBMITTING state with auto reconcile or

- When the Job is stuck in STARTING state in Job Admin
25. **BatchScheduleImport:** This notification is sent as an update on progress/errors encountered during the schedule import process.
 26. **BatchScheduleConfigImport:** This notification is sent when the Import Configuration function is used on the Batch Administration screen. When this notification is associated with an e-mail address, the e-mail notification is sent with an attached report. This report lists all mismatches between the imported configuration and the target batch schedule. For example, this report will list a job present in the import configuration file but is not there in the target schedule.
 27. **JobStarted:** This notification is sent when a job starts and the 'Notification At Start of Job' is checked on the Batch Administration screen for that job.
 28. **JobCompleted:** This notification is sent when a job completes successfully and the 'Notification At Completion of Job' is checked on the Batch Administration screen for that job.
 29. **JobCompletedWithWarning:** This notification is sent when a job completes with a warning. This indicates that this job's shell script exited with a code that was defined on the System Options as a Completion with Warning code. For more information on how these codes are setup, see the System Options tab section of the Appendix: Batch Schedule Spreadsheet Explained.

9

User Roles and OAuth Scopes

This chapter lists the pre-loaded user roles/scopes in POM.

Table 9-1 User Roles and OAuth Scopes

Roles	Scopes	Description
BATCH_MONITORING_JOB	rgbu:pom:services-monitor	One of the classic user interface roles. Users within this role are typically retailer administrators responsible for monitoring and executing batch. They can perform select activities on the Batch Monitor screen to move the schedule along.
BATCH_BUSINESS_JOB	rgbu:pom:services-customer	Another one of the classic user interface roles. Users within this role are typically retailer business users responsible for just monitoring batch and configuring POM to enable callbacks into the Company's systems.
BATCH_ADMINISTRATOR_JOB	rgbu:pom:services-administrator	Users within this role are retailer administrators with full access to all POM actions. They monitor, maintain and configure the batch schedules. They may also maintain POM application configurations for efficient operations. They troubleshoot batch issues and work with Oracle support personnel to address those issues. Finally, they may apply batch schedule patches and upgrades. Additionally, users assigned this role are given access to the Oracle AMS Utilities screen.
BATCH_VIEWER_JOB	rgbu:pom:services-viewer	Users within this role are retailer business users responsible for just monitoring batch. They have view access to all POM screens except AMS Utilities.
BATCH_SCHEDULE_CONFIGURATION_MANAGER_JOB	rgbu:pom:services-customer-manager	Users within this role are typically retailer administrators responsible for just monitoring batch and configuring external dependencies and callbacks into the Company's systems. They have view access to all POM screens except AMS Utilities.
BATCH_SCHEDULE_ADMINISTRATOR_JOB	rgbu:pom:services-customer-administrator	Users within this role are typically retailer administrators responsible for maintaining monitoring and executing batch. They have view access to all POM screens except AMS Utilities. They can perform select activities on the Batch Monitor screen to move the schedule along. They also have update access to the Batch Administration screen. They can also configure some application properties and can configure a new schedule

Table 9-1 (Cont.) User Roles and OAuth Scopes

Roles	Scopes	Description
BATCH_ORACLE_AMS_ADMINISTRATOR_JOB	rgbu:pom:services-ams-administrator	Users within this role are typically Oracle AMS administrators who monitor, maintain and configure the batch schedules. They also maintain POM application configurations for efficient operations. They troubleshoot batch issues and work with other Oracle development and support personnel to address those issues. Finally they apply POM and batch schedule patches and upgrades.

 **Note:**

The first two roles mentioned in the above table are associated with POM's classic user interface. They are being deprecated along with the classic user interface. Customers need to migrate to the other four non-Oracle roles before those classic roles are removed.

These roles have been given similar access in the new user interface as the access they had in the classic user interface.

For more information regarding functional access of each POM role, refer to the *Oracle® Retail Process Orchestration and Monitoring Cloud Services Security Guide*.

10

Invoking POM Services Using OAuth

This chapter details the steps to invoke POM ReST services using the OAuth protocol. Using the OAuth protocol is a two-step process:

- Request an access token from an authentication provider: IDCS or OCI IAM.
- Provide the access token as an authorization header when invoking a service.

Prerequisite

As a one-time setup, Customers are required to create an OAuth client using the Retail Home Create IDCS OAuth 2.0 Client function. The OAuth client must be created against the "POM" app with the scope

```
rgbu:pom:services-customer-administrator-<ENV_ID>
```

where <ENV_ID> represents the unique environment identifier such as PRD1, STG1, DEV1 and so on.

For example, the DEV1 scope would be:

```
rgbu:pom:services-customer-administrator-DEV1
```

For more information about creating the OAuth client, refer to the "Creating IDCS OAuth 2.0 Client Apps" chapter in the *Retail Home Administration Guide*.

Requesting the Access Token

To generate a token from IDCS, the following information is needed:

- IDCS URL
- Client Id and Client Secret
- OAuth Scope

The curl command below invokes an IDCS service to generate an access token:

```
curl -I -H 'Authorization: Basic <base64Encoded OAuth_Clientid:Secret>' -H 'Content-Type: application/x-www-form-urlencoded;charset=UTF-8' --request POST <IDCS_URL>/oauth2/v1/token -d 'grant_type=client_credentials&scope=rgbu:pom:services-customer-administrator-<ENV_ID>'
```

This is a standard ReST call, with the following specifics:

- <IDCS URL> is the IDCS URL of this instance.

- <base64Encoded OAuth_Clientid:Secret> is the Base64-encoded OAuth Client Id and Client Secret provided as a Basic Authentication header.
- Specify the body as:

```
grant_type=client_credentials&scope=
rgbu:pom:services-customer-administrator-<ENV_ID>
```

The response to this call will be in this format:

```
{
  "access_token": "<TOKEN>",
  "token_type": "Bearer",
  "expires_in": 3600
}
```

Invoking the POM Service

To invoke the POM ReST service, you must add an authorization header as Bearer <token>, that is:

- The word Bearer
- A space
- A valid token obtained as described in Requesting the Access Token

For example, the POM nightly cycle start request would look something like the following:

```
curl -i -H 'Authorization: Bearer <OAuth Token>' -H 'Content-Type:
application/json' --request POST 'http://<pom-server-host>/
ProcessServices/services/private/executionEngine/schedules/
<Schedule_Name>/execution?skipVersion' -d '{ "cycleName" : "Nightly",
"flowName" : "Nightly"}'
```

A

Batch Schedule Spreadsheet Explained

Overview

This appendix explains all tabs and fields of the batch schedule spreadsheet template. The template was originally developed for internal Oracle use and has since been made available to system integrators and customers for customizing retail application provided schedules or for developing custom schedules. For the latter use, custom batch schedules may be less complex and may not require some tabs and fields.

Guidelines for Updating the Batch Schedule Spreadsheet

Process Tab

This tab allows you to define all processes for the schedule at hand. A process is a group of jobs that are to be run sequentially from start to end. The following is a description of all fields on this tab:

- **ProcessName**

- Uniquely identifies a process. The Process Name should be in upper case only with no spaces. Use an underscore if needed. It should end with `XXX_PROCESS`.
- The process name is the batch job name appended with `_PROCESS`.

For example: `DEALUPLD_PROCESS`

`SA_TRANSACTION_LOADING_PROCESS`

- For non-Merchandising processes, the process name should start with the Application name.

For example: `REIM_POSTING_PROCESS`, `ALC_DAILY_CLEANUP_PROCESS`

- If a process is part of the nightly batch cycle but is also an ad hoc job, or if it's also part of a recurring flow, three separate processes should be defined. The ad hoc process should end with `XXX_PROCESS_ADHOC` and the recurring process should end with `XXX_CYCLE_PROCESS`.

For example:

- * `SA_TRANSACTION_LOADING_PROCESS` – for the sales audit transaction loading jobs that will run in the nightly batch cycle.
- * `SA_TRANSACTION_LOADING_PROCESS_ADHOC` – for the sales audit transaction loading jobs that will run ad hoc.
- * `SA_TRANSACTION_LOADING_CYCLE_PROCESS` – for the sales audit transaction loading jobs that will run multiple times a day.

- **Description**

- Short description of the process. There should be no special characters. If the process contains just one job, the description can be the same as the batch job name.
- **DependencyType**
 - Valid values are `TIME`, `JOB` and `BOTH`.
 - If the first job in this process is triggered on completion of another job, mark as `JOB`.
 - If the process is scheduled to run at a specific time (for example, the first job of the nightly batch cycle, or part of a recurring flow process), mark as `TIME`.
 - If the process needs to run at a specific time and at the same time is dependent on another process, mark as `BOTH`.
 - If the process is ad hoc and is not part of the nightly batch cycle, leave as `NULL`.
- **ApplicationName**
 - This holds the application name which the batch process belongs to.
 - Valid values are `RMS`, `RPM`, `REIM`, `RESA`, `ALLOC`, `RDE`, `MFP`, `AIPFSL`, `RDF`, and so on.
- **AdhocInd**
 - Valid values are `Y` and `N`.
 - If the job(s) in the process are ad hoc jobs (can be run any time and not part of the nightly batch cycle), mark as `Y`. Otherwise, mark as `N`.
 - A process should not have a mix of ad hoc and scheduled jobs in it.

Job Tab

The Job tab contains all the individual jobs to be executed as part of the schedule at hand.

- **JobName**
 - Uniquely identifies a job. The Job Name should be in upper case only with no spaces. Use an underscore if needed. It should end with `XXX_JOB`. If the same job is part of multiple processes (for example, different parameters are passed), define separate jobs for it.

Examples:

 - * `EXPORT_DIFFS_JOB` (under `EXPORT_DIFFS_PROCESS`) – for the `export_diffs.ksh` program passing in 'delta' as a parameter
 - * `EXPORT_DIFFS_FULL_JOB` (under `EXPORT_DIFFS_FULL_PROCESS`) – for the same `export_diffs.ksh` program but this time passing in 'full'
 - For non-Merchandising jobs, the job name should start with the Application, (for example, `REIM_POSTING_JOB`, `ALC_DAILY_CLEANUP_JOB`, `RDE_SEASNSDE_JOB`)
 - If the job is part of the nightly batch cycle and is also part of a recurring flow, define two separate jobs for it.

Examples:

- * UPLOADSALES_JOB belonging to SALESPROCESS_PROCESS – for the uploadsales.ksh program run as part of the nightly batch cycle
- * UPLOADSALES_CYCLE_JOB belonging to SALESPROCESS_CYCLE_PROCESS – for the same uploadsales.ksh program run as part of a recurring flow

- **Description**

- Short description of the batch program. There should be no special characters.

- **RmsBatch**

- This holds the exact batch executable that the job refers to. For Pro*C programs, this should be the binary without the .pc extension. For KSH scripts, include the .ksh extension. This field is case-sensitive.
- For service type jobs (non-shell script EXEC type jobs), this field should be left blank.

- **RmsWrapper**

- If applicable, this holds the exact batch wrapper name used to call the batch program. This field is case-sensitive.
- For service type jobs (non shell script EXEC type jobs), this field should be left blank.

- **ScriptFolder**

- This holds the directory path where the Wrapper file resides in the system.
- For service type jobs (non-shell script EXEC type jobs), this field should be left blank.

- **ParameterValue**

- This holds the entire parameter value to be passed in to a shell script type job. This field is case sensitive.
- For parameters that can have multiple values (for example, purge days parameter), provide a default value so that the batch can still be executed.
- Placeholder parameter #JobCtxt.businessDate can be defined on the Job sheet for jobs that require a POM business date parameter (DDMMYYYY format).

- **ApplicationName**

- This holds the application that the batch program belongs to.
- It should be a valid application mentioned in the Application tab.

- **Module**

- This holds the name of the module under the application which the batch program belongs to. The Application / Module entered here should be a valid combination in the Application tab.
- If there are no modules defined for the corresponding application in the Application tab or the Modules tab, then this column can be left blank. Otherwise it is required to enter the module(s) it belongs to.
- A job can belong to multiple modules under the same application. In this case the list of those modules should be entered here with || as a delimiter.

For example:

```
MODULE1 || MODULE2 || MODULE3
```

- If the corresponding Application have some modules defined in Application Sheet then this column cannot be left blank.

- **FixedParameterInd – VARCHAR(1)**
 - Valid values are Y and N.
 - This indicates whether the parameter value can be changed on the Batch Administration screen. If this is Y, the parameter will be changeable on the UI. This is applicable if the parameter cannot have a default value, or if the parameter can have different values.
 - If this is N, the parameter value is fixed and will not be changeable on the UI.
- **ParameterUpdated – VARCHAR(1)**
 - Valid values are Y and N.
 - This indicates whether the parameter value column has changed from the previous value.
 - If Y, then the parameter value is passed and updates the existing job information with the newly passed parameter value. If N the parameter value won't be updated for the existing job information.
- **SkipOnError – VARCHAR(1)**
 - Valid values are Y and N.
 - A value of Y indicates that the job should be skipped if an error occurs and the batch schedule continues to run. Otherwise the batch schedule is stopped.
- **JobType – VARCHAR (50)**
 - The default value set is EXEC, which represents shell-script based Jobs. Other pre-defined service based job types are RI, RASE, BDI and RPAS.
 - Custom Job types defined in the JobType tab can also be used in this field. In this case POM is directed to execute the job using the endpoint defined on the JobType tab.
- **KillCleanupScript – VARCHAR (1000)**
 - Absolute path of the cleanup script to be run after killing a shell script (EXEC) type job from the POM UI.
 - Can include arguments as well along with the script.

For example:

D	E	F	G
ScriptFolder	RmsWrapper	ParameterValue	KillCleanupScript
/u01/retail/rms/batch	rmswrap.ksh	dlyprg #SysOpt.dbwallet	/u01/retail/rms/batch/outgoing/app_cleanup_script.sh
	rmswrap.ksh	prepost #SysOpt.dbwallet dlyprg post	/u01/retail/rms/batch/outgoing/app_cleanup_script.sh

H	I	J	K	L	M
ApplicationName	Modules	FixedParameterInd	ParameterUpdated	JobType	SkipOnError
RMS		N	N	BBB	N
RMS		N	N		N

Process Job Mapping

The Process Job Mapping tab allows you to map or group jobs within processes.

- **ProcessName – VARCHAR(50)**

- This should match the **ProcessName** in the Process tab.
- **Job Name – VARCHAR(50)**
 - This should match the **JobName** in the Job tab.
- **DayOfTheWeek – VARCHAR(100)**
 - Contains the day(s) of the week on which the specific Process/Job needs to run. POM automatically skips these Jobs on the remaining days.
 - This field is optional, leaving it blank will cause the Job to run on a daily basis.

Example

If a process has 3 jobs in it, then there should be 3 entries in this tab for each job.

ProcessName	JobName	DayOfTheWeek
SAPURGE_PROCESS	SAPURGE_PRE_JOB	Sunday
SAPURGE_PROCESS	SAPURGE_JOB	Tuesday, Wednesday
SAPURGE_PROCESS	SAPURGE_POST_JOB	Thursday, Friday

Dependency Tab

The Dependency tab allows you to define the dependencies between jobs. For each job in a process, you can define the predecessor job and the corresponding process.

- **ProcessName – VARCHAR(50)**
 - This should match the **ProcessName** in the ProcessJobMapping tab.
- **JobName – VARCHAR(50)**
 - This should match the **JobName** in the ProcessJobMapping tab.
- **PredecessorProcessName – VARCHAR(50)**
 - This holds the process name that should run before the current job. This can be the same process (if there are multiple jobs in the process) or a different process than the current one.
- **PredecessorJobName – VARCHAR(50)**
 - This holds the job name in the predecessor process.

Example

ProcessName	JobName	PredecessorProcessName	PredecessorJobName
SAPURGE_PROCESS	SAPURGE_PRE_JOB	SAEXPRMS_PROCESS	SAEXPRMS_POST_JOB
SAPURGE_PROCESS	SAPURGE_JOB	SAPURGE_PROCESS	SAPURGE_PRE_JOB
SAPURGE_PROCESS	SAPURGE_POST_JOB	SAPURGE_PROCESS	SAPURGE_JOB

If an ad hoc process only contains one job, there's no need to fill up the Dependency tab for such process.

But if an ad hoc process has more than 1 job, the dependencies of jobs 2, 3, and so on need to be defined. There's no need to define the dependency of the first job, because this will not be dependent on another process/job.

Example

The ad hoc process FCUSTOMERUPLOAD_PROCESS_ADHOC has 2 jobs:

- FCUSTOMERUPLOAD_JOB
- FCUSTOMERPROCESS_JOB

Only the dependency of the second job (FCUSTOMERPROCESS_JOB) needs to be defined:

ProcessName	JobName	PredecessorProcessName	PredecessorJob Name
FCUSTOMERUPLOAD_PROCESS_ADHOC	FCUSTOMERPROCESS_JOB	FCUSTOMERUPLOAD_PROCESS_ADHOC	FCUSTOMERUPLOAD_JOB

Recurring Flow Tab

The Recurring Flow tab allows you to define jobs that runs hourly or multiple times a day.

- **RecurringFlowName – VARCHAR(50)**
 - Uniquely identifies the recurring flow. The Recurring Flow Name should be in uppercase only with no spaces. Use an underscore if needed. It should end with XXX_CYCLE.
- **Description – VARCHAR(50)**
 - Short description of the recurring flow. There should be no special characters.
- **NumberOfRuns – NUMBER(2)**
 - This contains the number of runs for this recurring flow in a day.
 - Valid values are 1-24. For hourly runs, maximum value is 12. For half hourly runs, the maximum value is 24.
- **Interval – NUMBER(2)**
 - The interval between job runs. The Interval is specified in hours. Upper Limit is 12.
- **StartTime – VARCHAR(50)**
 - The time for the first batch run.

Recurring Flow Process Tab

The Recurring Flow Process tab allows you to define the processes for each recurring job defined in the Recurring Flow tab.

- **RecurringFlowName – VARCHAR(50)**
 - This should match the **RecurringFlowName** in the Recurring Flow tab.
- **ProcessName – VARCHAR(50)**
 - This holds the process name(s) that are part of the recurring flow. This should match the **ProcessName** in the Process tab. A recurring flow can have more than 1 process.

- **FirstProcessInd – VARCHAR(1)**
 - Valid values are Y and N.
 - If the process holds the first job of the recurring flow, mark as Y. Otherwise, mark as N.
- **LastProcessInd – VARCHAR(1)**
 - Valid values are Y and N.
 - If the process holds the last job of the recurring flow, mark as Y. Otherwise, mark as N.

Example

The SALESPROCESS_CYCLE recurring flow has only 1 process, so the first and last jobs of the recurring flow is in the same process.

The REPLENISHMENT_CYCLE recurring flow has multiple processes. The first job of the recurring flow is in the first process, and the last job is in the last process.

RecurringFlowName	ProcessName	FirstProcessInd	LastProcessInd
SALESPROCESS_CYCLE	SALESPROCESS_CYCLE_PROCESS	Y	Y
REPLENISHMENT_CYCLE	REPLENISHMENT_CYCLE_PROCESS	Y	N
REPLENISHMENT_CYCLE	SUPSPPLIT_CNTRPRSS_CYCLE_PROCESS	N	N
REPLENISHMENT_CYCLE	INVESTMENT_BUY_CYCLE_PROCESS	N	N
REPLENISHMENT_CYCLE	RPLBLD_CYCLE_PROCESS	N	N
REPLENISHMENT_CYCLE	REPLENISHMENT_END_CYCLE_PROCESS	N	Y

Obsolete Tab

The Obsolete tab allows you to keep track of any process or job that has been deleted from an active environment. Enter the Name and Type of deleted items on this tab.

- **Name – VARCHAR**
 - Holds the name (for example, Job Name, Process Name, Recurring Flow Name) of the item of type (Job, Process, Flow) being removed from the schedule.
 - For ProcessJobMapping, Dependency / Inter Schedule Dependency/ External Dependency type, it holds the pattern.
 - For ProcessJobMapping type, the pattern should be `process#job`
 - For dependency type, the pattern should be `preJob#preProcess#job#process`
 - For InterScheduleDependency type, the pattern should be `job#process#externalScheduleName#externalJob#externalProcess`
 - For ExternalDependency type, the pattern should be `job#process#externalJob`

- For BatchLink (Execution Links) type, the pattern should be
job#process#externalScheduleName#externalJob#externalProcess
- **Type – VARCHAR**
 - Valid values are Process, Job, ProcessJobMapping, Flow, Dependency, InterScheduleDependency, ExternalDependency, BatchLink, Application and Module.

Example

Name	Type
RMSE_MFP_INVENTORY_JOB	Job
RMSE_RDF_DAILY_SALES_PROCESS	Process
DELETE_TAB_STATS_PROCESS#DELETE_TAB_STATS_JOB	ProcessJobMapping
SAIMPTLOGI_POST_JOB#SA_TRANSACTION_LOADING_PROCES S_ADHOC#SAVOUCH_JOB#SA_TRANSACTION_LOADING_PROCE SS_ADHOC	Dependency
DELETE_TAB_STATS_JOB#DELETE_TAB_STATS_PROCESS#RDE#R DE_RTLRDEZIP_JOB#RDE_RTLRDEZIP_PROCESS	InterSchedule Dependency
ORBATCH_VERIFY_RCI_JOB#FILE_VALIDATION_PROCESS#EXTE RNAL_JOB_NAME	ExternalDependency
DELETE_TAB_STATS_JOB#DELETE_TAB_STATS_PROCESS#RDE#R DE_RTLRDEZIP_JOB#RDE_RTLRDEZIP_PROCESS	BatchLink
RCI	Application
RCICUSTOMER	Module
BDI_PRICING_PC_TX_CYCLE	Flow

System Option Tab

The System Option tab contains the system level options used to control certain aspects of the POM application, such as whether to enable throttling. It can also be used to define shell script error or warning exit codes. If any such codes are specified, they will cause a shell script based job to fail or complete with warning.

This tab can be left blank for service (non-EXEC) type jobs.

- **Name – VARCHAR(255)**
 - Holds the name of the System Option value needed in JOS, (for example, enableThrottling, WarningCode110, ErrorCode40).
- **Type – VARCHAR**
 - Valid values are Process and Job (for example, Job for system option enableThrottling or error/warning shell script exit codes).
 - This defines the JOS admin component the System Option is for. Job is for JosJobAdmin and Process is for BdiProcessFlowAdmin.
- **Description – VARCHAR(255)**
 - This holds the value for the System Option (for example, TRUE for system option enableThrottling or the error/warning message to be conveyed in the case of an error/warning shell script exit codes)

Application Tab

The Application tab allows you to define the Application name and its modules.

- **Name – VARCHAR(50)**
 - Holds the name of the application supported in the schedule (for example, RMS, RI, RDS).
 - This cannot be blank.
- **Description – VARCHAR(1000)**
 - Description of the application.
- **Modules – VARCHAR(50)**
 - List of module name(s) associated with the Application.
 - || is used as delimiter (for example, MODULE1 ||MODULE2 ||MODULE3)
 - This field can be blank when there are no modules defined for the Application.
 - If the module list is large (> 3000 chars), rather than filling modules in this field, list all modules in **Modules** tab. If larger than 3000 chars, the schedule generation will fail. If you decide to use the **Modules** tab, the modules in this field in the **Application** tab need to be blank, as it directs POM to look for modules in the **Modules** tab instead.
- **JosJobAdminName – VARCHAR(50)**
 - Corresponding Job Admin name (in GBUCS) or Agent name (in CFS) for the application. This usually is the same or similar name to the application name.
 - This cannot be blank.

Example

- Application with modules defined.

Name	Description	Modules	JosJobAdminName
COMMON	All apps required programs	COMMONMAINTAIN PRE_VERIFY CALENDAR COMMONPRODUCT COMMONORG COMMONPROMO EMPLOYEE COMMONCO COMMONSALES REASON COMMONAC COMMONCS COMMONASO COMMONCDT COMMONDT	RIS1
ORASERCI	ORASE and RCI required programs	ORASERCICUSTOMER CONSUMER CATMAN ORASERCIMARKET TRADEAREA CUSTSEG MARKETAGG ORASERCIAAC ORASERCICS ORASERCIASO ORASERCICDT ORASERCIDT	RIS2
RCI	RCI required programs	RCICUSTOMER LOYALTY PROMOFORCAST RCIMARKET	RIS1
RSP	RSP required programs	AC CS ASO CDT DT	RIS1

- Applications without any modules defined.

Name	Description	Modules	JosJobAdminName
RMS	RMS		RMS
ALLOC	ALLOC		RMS
RDE	RDE		RDE
REIM	REIM		RMS
RESA	RESA		RMS
RPM	RPM		RMS

Schedule Tab

The Schedule tab contains the name of the schedule along with version and description. Increasing the version is necessary to upload a spreadsheet with any changes to POM.

- **ScheduleName – VARCHAR(10)**
 - Holds the name of the Schedule (for example, MERCH, RDE, or RI).
- **Description – VARCHAR(100)**
 - Description of the Schedule.
- **Version – VARCHAR(50)**
 - Version of the Schedule.

Example

ScheduleName	Description	Version
Merch	RMS Schedule	21.1.102.0

This tab cannot be empty for any given schedule.

Throttling Configuration Tab

The Throttling Configuration tab contains the Application / Module Level throttling values. Throttling determines how many jobs can run concurrently for the given application, so a value of 10 means a maximum of ten jobs can run concurrently. It is advisable to leave this tab empty unless you have good knowledge of the server memory and CPU capacity.

- **Application – VARCHAR(50)**
 - Holds the valid application name for which we need to set a throttle value.
 - If an entry is made in this tab, the Application can't be blank and should be the value listed in Application tab.
- **Module – VARCHAR(50)**
 - Holds the module name in the case where we need to set a throttle value at module level.

- It is possible to just set throttling at the application level. Modules would then inherit the application's throttle value. In this case, leave module blank.
- This Application / Module combination must be valid per the Application tab.
- **ThrottledValue – NUMBER**
 - Holds the throttle value for this Application / Module.
 - This can't be blank and should be a non-zero positive number.

Example

- For an Application without any modules defined.

Application	Module	ThrottledValue
RMS		2
ALLOC		5
REIM		3
RDE		2

- For an Application with modules defined, define throttle values at either application or module level or both.

Application	Module	ThrottledValue
COMMON		3
COMMON	COMMONMAINTAIN	1
ORASE		2

If you don't want any jobs to be throttled at Application / Module level, then leave this tab empty.

Inter Schedule Dependency Tab

The Inter Schedule Dependency tab allows you to define dependencies from jobs on this schedule to jobs on other schedules on the same instance of POM. For example, job A of this schedule can be made dependent on job B from another schedule. Job A will then wait for completion of job B.

- **ProcessName – VARCHAR(50)**
 - Holds the name of the process from the current schedule that is dependent on a different schedule.
 - This process name should be listed in the **Process** tab.
- **JobName – VARCHAR(50)**
 - Holds the name of the corresponding job within the above process that has the dependency on another external schedule.
 - This Process Job combination should be valid per the **Process Job Mapping** tab.
- **ExternalScheduleName – VARCHAR(10)**
 - Holds the name of the external schedule that contains the dependency.
- **ExternalPredecessorProcessName – VARCHAR(50)**

- Holds the name of the process that contains the job from an external schedule which needs to complete before the job in the current schedule starts running.
- **ExternalPredecessorJobName – VARCHAR(50)**
 - Holds the name of the corresponding job (from the external schedule) that must complete before the job in the current schedule starts running.

Example

ProcessName	JobName	ExternalScheduleName	ExternalPredecessorProcessName	ExternalPredecessorJobName
ALLOCBT_PROCESS	ALLOCBT_JOB	RDE	RDE_SETUP_PROCESS	RDE_REFRESHODIVARIABLES_JOB
DISTROPCPUB_PROCESS	DISTROPCPUB_JOB	RDE	RDE_SUPSDE_PRODIMSUPSDE_PROCESS	RDE_SUPSDE_JOB
CMPUPLD_PROCESS	CMPUPLD_JOB	RPAS	RPAS_PROCESS	RPAS_JOB
ALC_PURGE_WRK_PROCESS	ALC_PURGE_WRK_JOB	RIORASE	RIORASE_PROCESS	RIORASE_JOB

If there are no inter schedule dependencies for any jobs in the schedule, then leave this tab empty.

Modules Tab

The Modules tab allows you to define the mapping between the POM Modules and the Module Definition Framework (MDF) Modules. It also serves a second purpose by defining the modules for an application. If the module field on the **Application** tab is not blank, POM will use those modules. Otherwise, if blank, POM will use the modules from this tab.

MDF holds the applications and modules to which a customer has subscribed. POM can then optionally sync with MDF to activate or deactivate applications and/or modules.

- **Application – VARCHAR(50)**
 - Holds the valid application name for which we need to set an MDF Module Path.
 - This can't be blank and should be listed in **Application** tab.
- **Module – VARCHAR(50)**
 - Holds the module name for cases when the MDF Module Path must be set at module level.
 - This can be blank when the MDF Module Path is only set at the application level.
 - This Application / Module combination should be valid per the **Application** tab.
- **MDFModulePath – VARCHAR(4000)**
 - Holds the module path in the MDF application.
 - This can't be blank. This should be a slash (/) separated path.
For example: /RMS, /COMMON/COMMONAC

Example

Application	Module	MDFModulePath
COMMON	COMMONAC	/COMMON/COMMONAC
COMMON	CALENDAR	/COMMON/CALENDAR
ORASE		/ORASE
COMMON		/COMMON
ORASERCI		/ORASERCI

External Dependencies Tab

This tab allows you to define the dependencies of POM jobs on external schedulers like Ctrl-M.

- **ProcessName – VARCHAR(50)**
 - Holds the name of the process in the current schedule for which the external job is to be configured.
 - This process name should be listed on the **Process** tab.
- **JobName – VARCHAR(50)**
 - Holds the name of the corresponding job of the above process for which the external job is to be configured.
 - This Process Job combination should be valid per **Process Job Mapping** tab.
- **ExternalPredecessorJobName – VARCHAR(300)**
 - Holds the name of the external job that needs to complete before the job in the current schedule can start.

Example

ProcessName	JobName	ExternalPredecessorJobName
FILE_VALIDATION_PROCESS	ORBATCH_VERIFY_RICOMMON_JOB	EXT_JOB_1
FILE_VALIDATION_PROCESS	ORBATCH_VERIFY_RMI_JOB	EXT_JOB_2
FILE_VALIDATION_PROCESS	ORBATCH_VERIFY_RCI_JOB	EXT_JOB_3

Batch Links Tab

This tab contains Batch Links. When you set up a batch link, you direct POM to start a job in schedule B upon completion of a job in schedule A. It's a way of linking two schedules together and having one start the other. You want the job in schedule B to be the first job of the schedule.

If no such link is desired, leave this tab empty.

- **ProcessName – VARCHAR(50)**

- Holds the name of the process in the current schedule that needs to be invoked by a process / job from an External Schedule.
- This process name should be listed in **Process** tab.
- **JobName – VARCHAR(50)**
 - Holds the name of the corresponding job of the above process that needs to be invoked from the other external schedule.
 - This Process Job combination should be valid as per the **Process Job Mapping** tab.
- **InvokerScheduleName – VARCHAR(10)**
 - Holds the name of the external schedule whose job is to invoke the current schedule's job.
- **InvokerProcessName – VARCHAR(50)**
 - Holds the name of the process in the invoking schedule.
- **InvokerJobName – VARCHAR(50)**
 - Holds the name of the corresponding job from the invoking schedule.

Example

	A	B	C	D	E
1	ProcessName	JobName	InvokerScheduleName	InvokerProcessName	InvokerJobName
2	RDE_SETUP_PROCESS	RDE_REFRESHODIVARIABLES_JOB	MERCH	START_BATCH_PROCESS	START_BATCH_JOB
3					

If there are no batch links needed then leave this tab empty.

Job Types Tab

This tab allows users to define custom job types. These are job types other than the POM-provided ones, namely: EXEC, RI, RASE, BDI, and RPAS.

These custom job types are associated with the ReSTful endpoints necessary for POM to execute batch. Once defined on this tab, these can then be used in the JobType field on the **Job** tab.

- **Type – VARCHAR (10)**
 - Name of the job type (for example, RDS).
- **ValidationPath – VARCHAR (4000)**
 - ReST Endpoint path for validating that the endpoints for this job type are reachable (for example, /validation).
 - Can be left blank on this tab but, if so, needs to be provided later on the UI.
- **JobStartPath – VARCHAR (4000)**
 - ReST Endpoint path to start a job (for example, /start).
 - Mandatory field if a job type is defined on this tab.
- **JobRestartPath – VARCHAR (4000)**

- ReST Endpoint path to restart a failed job (for example, /restart).
- Can be left blank on this tab but, if so, needs to be provided later on the UI.
- **JobStatusPath – VARCHAR (4000)**
 - ReST Endpoint path to check the status of a previously submitted job (for example, /status).
 - Mandatory field if a job type is defined on this tab.
- **JobLogPath – VARCHAR (4000)**
 - ReST Endpoint path to fetch log file of a job (for example, /logs).
 - Can be left blank on this tab but, if so, needs to be provided later on the UI.
- **JobKillPath – VARCHAR (4000)**
 - ReST Endpoint path to kill a running job.
 - Can be left blank on this tab but, if so, needs to be provided later on the UI.
- **OAuthScopes – VARCHAR (4000)**
 - Comma-separated list of OAuth scopes for invoking the endpoints.
 - Mandatory field if a job type is defined on this tab.

If there are no custom job types in the schedule, then leave this tab empty.