# Oracle® Banking Liquidity Management Cloud Service Configuration Guide





Oracle Banking Liquidity Management Cloud Service Configuration Guide, Release 14.7.3.0.0

F99147-01

Copyright © 2023, 2024, 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, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

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

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

# Contents

Purpose	,
Audience	,
Documentation Accessibility	,
Diversity and Inclusion	,
Related Resources	V
Conventions	V
Screenshot Disclaimer	V
Acronyms and Abbreviations	V
Basic Actions	V
Symbols and Icons	V
EOD Configuration	
1.1 Mapping Functional Activity Code	1-:
1.2 Upload DSL	1-1
1.3 Configure EOD	1-3
1.4 Run EOD for branch	1-4
Job Definition Naming Convention	
Oracle Banking Liquidity Management Job	
Intraday Jobs	
4.1 Create Task	4-3
4.2 Configure Tasks	4-4
Functional Activity Codes	



Index



#### **Preface**

- Purpose
- Audience
- Documentation Accessibility
- · Diversity and Inclusion
- Related Resources
- Conventions
- Screenshot Disclaimer
- Acronyms and Abbreviations
- Basic Actions
- Symbols and Icons

#### Purpose

This guide quickly get acquainted with the many functions every day on a routine basis as part of the End of Day (EOD).

#### **Audience**

This guide is intended for Back Office Data Entry Clerk, Back Office Managers/Officers, Product Managers, End of Day Operators, and Financial Controller users.

## **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.

## **Diversity and Inclusion**

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.



## **Related Resources**

The related documents are as follows:

- Oracle Banking Common Core User Guide
- · Oracle Banking Liquidity Management User Guide
- Tasks User Guide

#### Conventions

The following text conventions are used in this document:

Convention	Meaning	
<b>boldface</b> Boldface type indicates graphical user interface elements associated action, or terms defined in text or the glossary.		
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.	
monospace Monospace type indicates commands within a paragraph, URLs examples, text that appears on the screen, or text that you enter.		

#### Screenshot Disclaimer

Personal information used in the interface or documents is dummy and does not exist in the real world. It is only for reference purposes.

## **Acronyms and Abbreviations**

The list of the acronyms and abbreviations that are used in this guide are as follows:

Table 1 Acronyms and Abbreviations

Abbreviation	Description	
API	Application Programming Interface	
EOD	End of Day	

#### **Basic Actions**

The basic actions performed in this guide are as follows:

**Table 2 Basic Actions** 

Actions	Description
Save	Click <b>Save</b> to save the details entered or selected in the screen. Saved record details will be available in 'View Screen'



Table 2 (Cont.) Basic Actions

Actions	Description	
Close	Click <b>Close</b> to close a record. The system displays a warning message to the user that any unsaved data would be lost. User can either choose to ignore the message and close the screen or choose to 'save' the record	
Cancel	Click <b>Cancel</b> to cancel the action performed without saving any data. The user is alerted that the input data would be lost before confirming the cancellation.	
Next	Click <b>Next</b> to navigate to the next data segment, after successfully capturing the data.	
Create	Click <b>Create</b> to capture the data entered and create the new record. Created record details will be available in 'View Screen'	
Back	Click <b>Back</b> to navigate to the previous data segment, without lost of any data entered or captured from current screen.	
Delete	Click <b>Delete</b> to delete the task listed.	
Fetch	Click <b>Fetch</b> to fetch the EOD details.	
Start	Click Start to invoke the EOD operation.	
Reset	Click <b>Reset</b> to clear the EOD records invoked.	
Retry	Click <b>Retry</b> to restart the EOD operation.	
Refresh	Click <b>Refresh</b> to view the EOD operation.	

# Symbols and Icons

This guide has the following list of symbols and icons.

Table 3 Symbols and Icons - Common

Symbol/Icon	Function
J L	Minimize
7 6	
X	Close
Q	Perform Search
K	Navigate to the first record
>	Navigate to the last record

Table 3 (Cont.) Symbols and Icons - Common

Symbol/Icon	Function
4	Navigate to the previous record
•	Navigate to the next record



1

# **EOD Configuration**

This topic provide information about the EOD Configuration process.

This topic contains the following subtopics:

- Mapping Functional Activity Code
   The topic describes the information to map the functional activity code to perform EOD operations.
- Upload DSL

This topic describes the systematic instructions to upload DSL in Business Process maintenance.

- Configure EOD
   This topic describes the systematic instructions to configure EOD operations
- Run EOD for branch
   This topic describes the systematic instructions to run the EOD for a branch.

## 1.1 Mapping Functional Activity Code

The topic describes the information to map the functional activity code to perform EOD operations.

The following functional activity code needs to be maintained in user's role to perform EOD operations:

CMC\_FA\_BRANCH\_EOD\_PROCESS



Refer to **Oracle Banking Security Management System User Guide** for the procedure to map the functional activity code in user's role.

## 1.2 Upload DSL

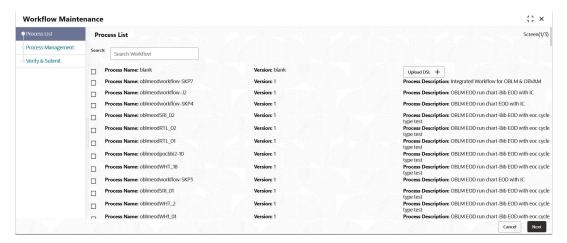
This topic describes the systematic instructions to upload DSL in Business Process maintenance.

Specify **User ID** and **Password**, and login to **Home** screen.

- Download the OBLMEOD.json file. This is a standard batch process definition script for Oracle Banking Liquidity Management that includes the list of batch tasks to be automatically executed in a sequence.
- On Home Screen, under Tasks menu, click Business Process Maintenance to import, create or modify batch process definition

The **Product List** screen displays.

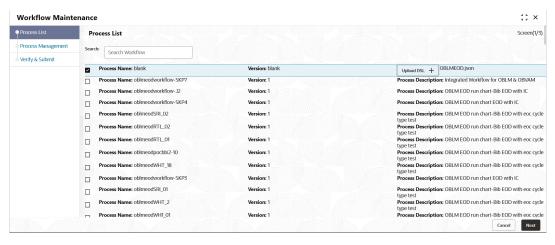
Figure 1-1 Process List



- 3. Select the Process Name: blank checkbox.
- 4. Click the **Upload DSL+** button to upload batch process definition.
- 5. Select the file **OBLMEod.json** from the local folder.

The Process List - Upload DSL screen displays

Figure 1-2 Process List - Upload DSL

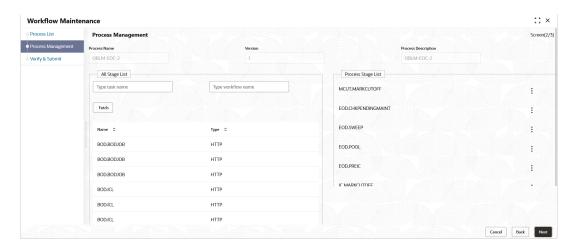


6. Click Next button.

The **Product Management** screen displays.



Figure 1-3 Process Management



7. Click Next button.

The Verify and Submit screen displays.

8. Click **Review** or **Create Process** to register the batch.

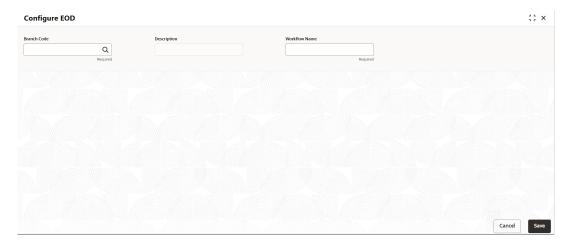
## 1.3 Configure EOD

This topic describes the systematic instructions to configure EOD operations

Specify **User ID** and **Password**, and login to **Home** screen.

On Core Maintenance menu, under Branch EOD, click Configure EOD.
 The Configure EOD screen displays.

Figure 1-4 Configure EOD





To configure batch for a branch, refer the **Configure Branch EOD** section in *Oracle Banking Common Core User Guide*.



2. Click **Search** icon to view and select the **Branch Code** to configure the batch.

#### 1.4 Run EOD for branch

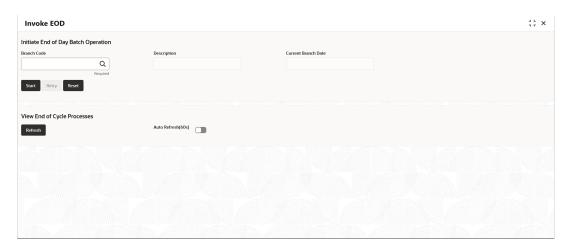
This topic describes the systematic instructions to run the EOD for a branch.

Specify User ID and Password, and login to Home screen.

- On Home Screen, click Core Maintenance. Under Core Maintenance menu, click Branch EOD.
- 2. Under Branch EOD, click Invoke EOD.

The Invoke EOD screen displays.

Figure 1-5 Invoke EOD



- 3. Click **Search** icon to view and select the branch code to run EOD.
- 4. Click **Refresh** to view the current status of the branch.



# Job Definition Naming Convention

This topic describes the naming convention that to be followed when a custom job is introduced as a task into EOD process.

1. Milestone task name must be prefixed with "MS-". Ex: MS-BranchCutOff

#### Milestone stage

Milestone stage will pause the batch execution till it is manually resumed.

#### Sample template for milestone stage

```
{
     "name": "MS-CHKAFTEREOTI",
     "taskReferenceName": "MS-CHKAFTEREOTI",
     "inputParameters": {
         "http request": {
             "connectionTimeOut": "0",
             "readTimeOut": "0",
             "vipAddress": "CMC-BRANCH-SERVICES",
             "uri": "/cmc-branch-services/brancheod/milestone",
             "method": "POST",
             "headers": {
                 "appId": "CMNCORE",
                 "branchCode": "${workflow.input.branchCode}",
                 "userId": "${workflow.input.userID}"
             },
             "body": {
                 "data": [
                         "workflowId": "${workflow.workflowId}",
                         "taskId": "${CPEWF TASK ID}",
                          "waitTime": "5000"
                 1
             }
         }
     "type": "HTTP",
     "startDelay": 0,
     "optional": false,
     "asyncComplete": true
```

#### **Steps to integrate Custom Jobs**

1. If the custom job uses Oracle Banking Microservices Architecture Batch service, then use the below template to include the job as a task in EOD Flow definition.

```
"type":"HTTP",
"name": "<MilestoneCode.JobName>",
"taskReferenceName": "<MilestoneCode.JobName>",
"inputParameters":{
   "http request":{
      "connectionTimeOut":"0",
      "readTimeOut":"0",
      "vipAddress": "PLATO-BATCH-SERVER",
      "uri": "/plato-batch-server/jobLauncher/launch/",
      "method": "POST",
      "headers":{
         "appId":"${workflow.input.appId}",
         "branchCode": "${workflow.input.branchCode}",
         "userId": "${workflow.input.userId}"
      },
      "body": {
         "jobName": "<JobName>",
         "jobParameters":[
                "key": "appId",
                "value":"<Application ID of microservice>"
            },
             {
                "key": "microServiceName",
                "value":"<Microservice name>"
            },
             {
                "key": "contextRoot",
                "value":"<Context root of microservice>"
            },
             {
                "key": "workflowId",
                "value": "${workflow.workflowId}"
            },
             {
                "key": "referenceTaskName",
                "value": "<MilestoneCode.JobName>"
            },
             {
                "key": "userId",
                "value": "${workflow.input.userId}"
            },
                "key": "branchCode",
                "value": "${workflow.input.branchCode}"
            },
             {
                "key": "isCallback",
                "value":"Y"
            },
                "key": "callbackType",
```

2. If the custom job doesn't use the Oracle Banking Microservices Architecture Batch service. The Batch API is implemented as a synchronous call, use the below template to include the job as a task in EOD Flow definition.

```
"type": "HTTP",
"name": "<MilestoneCode.JobName>",
"taskReferenceName": "<MilestoneCode.JobName>",
"inputParameters":{
   "http request":{
      "connectionTimeOut":"0",
      "readTimeOut":"0",
      "vipAddress":"<Microservice name registered in eureka>",
      "uri":"<relative URL>",
      "method": "<HTTP Method>",
      "headers":{
         "appId": "${workflow.input.appId}",
         "branchCode": "${workflow.input.branchCode}",
         "userId":"${workflow.input.userId}"
   },
   "asyncComplete":false
},
"startDelay":0,
"optional": false,
"asyncComplete":true
```

#### Note:

HTTP Method - One of the GET, PUT, POST, DELETE, OPTIONS, HEAD

3. If the custom job doesn't uses Oracle Banking Microservice Architecture Batch service and if the Batch API is implemented as an asynchronous call, then call back needs to be implemented in the respective API. Use the below template to include the job as a task in EOD Flow Definition.

```
{
  "type":"HTTP",
  "name":"<MilestoneCode.JobName>",
  "taskReferenceName":"<MilestoneCode.JobName>",
  "inputParameters":{
```



```
"http request":{
      "connectionTimeOut":"0",
      "readTimeOut":"0",
      "vipAddress":"<Microservice name registered in eureka>",
      "uri":"<relative URL>",
      "method":"<HTTP Method>",
      "headers":{
         "appId": "${workflow.input.appId}",
         "branchCode": "${workflow.input.branchCode}",
         "userId":"${workflow.input.userId}"
   },
   "asyncComplete":true
},
"startDelay":0,
"optional":false,
"asyncComplete":true
```

#### Table 2-1 Batch API

Method	Post	Description
URL	http:// <hostname>:<port>/plato-orch- service/api/tasks</port></hostname>	-
Headers	userId : <logged id="" in="" user=""></logged>	userId – User who updates the task
	branchCode : <logged branch="" code="" in=""> appld : platoorch</logged>	branchCode – Branch where the update is performed.
	Content-Type : application/json	
	Accept : application/json	
Body	{	EOD_Workflow_ID – A Workflow ID gets generated when EOD is invoked
	<pre>"workflowInstanceId": "<eod_workflow_id",< pre=""></eod_workflow_id",<></pre>	Task_ID – Unique task ID gets generated for each task once it starts
	"taskId": " <task_id>", "status": "<status>"}</status></task_id>	Status – COMPLETED / FAILED_WITH_TERMINAL_ERROR / FAILED / IN_PROGRESS

#### Note:

asyncComplete – field in EOD workflow definition should be set to true if the Http task makes an asynchronous call and the task has to be updated explicitly by calling above update APIs. Only after successful update, next task will get executed.



# Oracle Banking Liquidity Management Job

The topic describes the Oracle Banking Liquidity Management Job names and its descriptions.

Table 3-1 Oracle Banking Liquidity Management Job

S.No	EOD stage	Job Name	Description	Input Parame ters
1	MCUT	markcutoff	Job will check for pending tasks and any existing running process before starting EOD	BranchC ode
2	EOD	CHKPENDINGMAI NT	Job will check pending maintenances that required approval.	BranchC ode
3	EOD	SWEEP	Job will execute sweep process scheduled to run during EOD The following optional change can be done to the existing EOD.SWEEP to enable parallel processing feature:	BranchC ode
			<ol> <li>Modify -         inputParameters.http_request.body.j         obName = "eodSweepJobV2"</li> </ol>	
			<ol> <li>Add -         inputParameters.http_request.body.j         obParameters.key = "gridSize"</li> </ol>	
			3. Add - inputParameters.http_request.body.j obParameters.value = 5 (Can be changed as per need)	
4	EOD	CLEAR_BALANCE _CACHE	Job will clear the Value Dated balance cache that's populated during EOD Sweep executions.	BranchC ode
5	EOD	POOL	Job will execute all pool structures. The following optional change can be done to the existing EOD.POOL to enable parallel processing feature:	BranchC ode
			Modify -     inputParameters.http_request.body.j     obName = "eodPoolJobV2"	
			<ol> <li>Add -         inputParameters.http_request.body.j         obParameters.key = "gridSize"</li> </ol>	
			<ol> <li>Add -         inputParameters.http_request.body.j         obParameters.value = 5 (Can be         changed as per need)</li> </ol>	

Table 3-1 (Cont.) Oracle Banking Liquidity Management Job

S.No	EOD stage	Job Name	Description	Input Parame ters
6	EOD	PREIC	Job will execute the tasks that are required to run before starting Interest batch	BranchC ode
7	EOD	IC.MARKCUTOFF	Job will check for pending tasks and any existing running process before starting Interest batch	BranchC ode
8	EOD	OBLM-IC	Job will execute Interest batch	BranchC ode
9	MS-EOFI	MS-EOFI	Milestone for the end of financial input	BranchC ode
10	EOFI	MARKEOFI	Job will mark the end of financial input	BranchC ode
11	MS- CHKB4FLIPDATE	MS- CHKB4FLIPDATE	Milestone for date flip	BranchC ode
12	EOD	CMC.DATEFLIP	Job will change system date to next working date in common core	BranchC ode
13	EOD	OBLM.DATEFLIP	Job will change system date to next working date in Oracle Banking Liquidity Management	BranchC ode
14	EOD	RCUT.RELEASEC UTOFF	RELEASEC Job will mark release cutoff after EOD.	
15	EOD	IC.RELEASECUTO FF	Job will mark release cutoff for IC Batch.	BranchC ode
16	BOD	BOD.REALLOC	Job will execute reallocation. The following optional change can be done to the existing BOD.REALLOC to enable parallel processing feature:	BranchC ode
			<ol> <li>Modify -         inputParameters.http_request.body.j         obName = "bodReallocationJobV2"</li> </ol>	
			<ol> <li>Add -         inputParameters.http_request.body.j         obParameters.key = "gridSize"</li> </ol>	
			<ol> <li>Add -         inputParameters.http_request.body.j         obParameters.value = 5 (Can be         changed as per need)</li> </ol>	



Table 3-1 (Cont.) Oracle Banking Liquidity Management Job

S.No	EOD stage	Job Name	Description	Input Parame ters
17	BOD	BOD.SWEEP	Job will execute the Reverse sweep and BOD sweeps in sequence The following optional change can be done to the existing BOD.SWEEP to enable parallel processing feature:	BranchC ode
			1. Modify - inputParameters.http_request.body.j obName = "bodSweepJobV2"	
			<ol> <li>Add -         inputParameters.http_request.body.j         obParameters.key = "gridSize"</li> </ol>	
			<ol> <li>Add -         inputParameters.http_request.body.j         obParameters.value = 5 (Can be         changed as per need)</li> </ol>	
18	BOD	TI.MARKTI	Job will mark the transaction inputs	BranchC ode
19	BOD	BOD.ICL	Job will execute the ICL	BranchC ode
20	BOD	chargeCalculationJ ob	JoB will calculate the LM charges for given branch code	BranchC ode
21	BOD	chargePostingJob	Job will post the LM charges for given branch code	BranchC ode
22	OBLM	OBLM.UPDATEDE FERREDENTRY	Job will update the Deferred entries for pool contribution for given BranchCode	BranchC ode
23	OBLM	OBLM.UPLOADIN TERESTACCRUAL REPORT	Job will update the Accrual Interest details for given BranchCode	BranchC ode
24	IC	IC.INTRADAYBAT Job will update the Intra day batch to notify UBS		BranchC ode
25	OBLM	OBLM.UPDATEST Job will update the status of all RUCTURESTATUS structures under the EOD branch		BranchC ode
26	OBLM	OBLM.DATEFLIP  Job will update the branch date of the EOD branch in LM table		BranchC ode
27	OBLMIC	OBLMIC.DATEFLI P	Job will update the branch dates of the EOD branch in OBLM IC tables	BranchC ode
28	EOD	CHECKINSTANCE	Job will check the current EOD instance is updated in CMC table CMC_TB_BRANCHEOD_INSTANCES.	BranchC ode and Workflo wld



4

# **Intraday Jobs**

This topic provide information about the Intraday Jobs.

This topic contains the following subtopics:

- Create Task
  - This topic describes the systematic instructions to create the task.
- Configure Tasks

This topic describes the systematic instructions to configure the tasks.

#### 4.1 Create Task

This topic describes the systematic instructions to create the task.

Oracle Banking Liquidity Management Intraday jobs required the following tasks to be created:

1. On **Home** screen, under **Task Management** menu, click **Create Task**.

The Create Task screen displays.

Figure 4-1 Create Task





The fields marked as **Required** are mandatory.

2. Specify the values mentioned in the following table.

Table 4-1 Intraday Job - Task Values

SI. no	Task Name	Description	Task Definition
1	OBLM_intraDayAccountPair SweepJob_INT_001	This task initiates Intraday Account Pair sweep execution	appId:::LMS;microServiceName:::oblm -sweep-services;contextRoot:::oblm- sweep- services;type:::schedule;jobName:::intr aDayAccountPairSweepJob;cronExpre ssion:::0 0/5 * * * ?;
			If parallel Processing strategy should be used during intraday execution, following Task Definition is to be used. appld:::LMS;microServiceN ame:::oblm-sweepservices;contextRoot: ::oblm-sweepservices;type:::sched ule;jobName:::intraDayAcco unt PairSweepJobV2;cronExpre ssion:::0 0/5 * * * ?;
2	OBLM_intraDayStructureSw eepJob_INT_002	This task initiates Intraday Structure sweep execution	appld:::LMS;microServiceName:::oblm -sweep-services;contextRoot:::oblm- sweep- services;type:::schedule;jobName:::intr aDayStructureSweepJob;cronExpressi on:::0 0/5 * * * ?;
			If parallel Processing strategy should be used during intraday execution, following Task Definition is to be used.  appld:::LMS;microServiceN ame:::oblm-sweepservices;contextRoot: ::oblm-sweepservices;type:::sched ule;jobName::intraDayStruc ture SweepJobV2;cronExpressi on:::0 0/5 * * * ?;

Table 4-1 (Cont.) Intraday Job - Task Values

SI. no	Task Name	Description	Task Definition
3	OBLM_processMTHoldMes sagesJob_INT_003	This task processes MT Messages that are on hold	appld:::LMG;microServiceName:::obl m-messaging- services;contextRoot:::oblm- messaging- services;type:::schedule;jobName:::pr ocessMTHoldMessagesJob;cronExpre ssion:::0 0/5 * * * ?;
4	OBLM_pendingPaymentsJo b_INT_004	This task retriggers the sweep handoffs that are in pending state	appld:::LMX;microServiceName:::oblm -integration- services;contextRoot:::oblm- integration- services;type:::schedule;jobName::: pendingPaymentsJob;cronExpression: ::0 0/5 * * * ?;
5	OBLM_publishEventsLogJo b_INT_005	This task publishes the events to Kafka	appld:::LMX;microServiceName:::oblm -integration- services;contextRoot:::oblm- integration- services;type:::schedule;jobName:::pu blishEventsLogJob;cronExpression:::0 0/10 * * * ?;
6	OBLM_pendingReallocation Job_INT_006	This task retriggers the reallocation handoffs that are in pending state	appld:::LMX;microServiceName:::oblm -integration- services;contextRoot:::oblm- integration- services;type:::schedule;jobName:::pe ndingReallocationJob;cronExpression: ::0 0/10 * * * ?;
7	sweepEventsFetch	This task is to fetch and populate the sweep events to dashboard schema	appld:::LMD;microServiceName:::oblm -dashboard- services;contextRoot:::oblm- dashboard- services;type:::schedule;jobName:::sw eepLMDEventsJob;cronExpression:::0 0/5 * * * ?'
8	savetoChargeCalCollJob_L M	This task is needed for Charges, Only If LM is deployed	appld:::VAMLMCHG;microServiceNa me:::vamlm-charge- services;contextRoot:::vamlm-charge- services;jobName:::savetoChargeCalC ollJob;appCode:::LMCHG;
9	chargePostingJob_LM	This task is needed for Charges, Only If LM is deployed	appld:::VAMLMCHG;microServiceNa me:::vamlm-charge- services;contextRoot:::vamlm-charge- services;jobName:::chargePostingJob; appCode:::LMCHG;
10	chargeCalculationJob_LM	This task is needed for Charges, Only If LM is deployed	appld:::VAMLMCHG;microServiceNa me:::vamlm-charge- services;contextRoot:::vamlm-charge- services;jobName:::chargeCalculation Job;appCode:::LMCHG;



Table 4-1 (Cont.) Intraday Job - Task Values

SI. no	Task Name	Description	Task Definition
11	savetoChargeCalCollJob_V AM	This Task is needed for Charges. In case its a codeployed environment for both VAM and LM, then both VAM and LM tasks are required for charges.	appld:::VAMLMCHG;microServiceNa me:::vamlm-charge- services;contextRoot:::vamlm-charge- services;jobName:::savetoChargeCalC ollJob;appCode:::VAMCHG;
12	chargePostingJob_VAM	This task is needed for Charges. In case its a codeployed environment for both VAM and LM, then both VAM and LM tasks are required for charges	appld:::VAMLMCHG;microServiceNa me:::vamlm-charge- services;contextRoot:::vamlm-charge- services;jobName:::chargePostingJob; appCode:::VAMCHG;
13	chargeCalculationJob_VAM	This task is needed for Charges. In case its a codeployed environment for both VAM and LM, then both VAM and LM tasks are required for charges.	appld:::VAMLMCHG;microServiceNa me:::vamlm-charge- services;contextRoot:::vamlm-charge- services;jobName:::chargeCalculation Job;appCode:::VAMCHG;
14	platoOrchArchivejob	This task is needed for Purge, Conductor related tables.	appld:::LMX;microServiceName:::oblm -integration- services;contextRoot:::oblm- integration- services;jobName:::platoOrchArchivej ob;purgeConfigName:::< <purp>value Name&gt;&gt;;</purp>

3. Click **Create** to create the task for each Intraday job.

## 4.2 Configure Tasks

This topic describes the systematic instructions to configure the tasks.

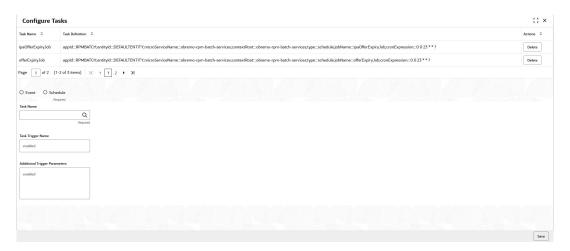
The Configured intra-day jobs will get triggered as per the specified Cron Expression, for the Create Task the scheduler needs to be configured as shown as follows.

1. On Home screen, under Task Management menu, click Configure Tasks.

The **Configure Tasks** screen displays.



Figure 4-2 Configure Tasks





The fields marked as **Required** are mandatory.

- 2. Select the Schedule button.
- 3. Select the task name from the **Task Name** drop-down list.
- 4. Specify the trigger name in **Task Trigger Name** field.
- 5. Specify the required CRON expression in **CRON Expression** field.
- 6. Click **Save** to configure the task.





# **Functional Activity Codes**

Table A-1 List of Functional Activity Codes

Screen Name	Functional Activity Code	Action	Purpose
Batch	LMS_FA_SWEEPDATA_VIE W	View	This functional activity code is used to fetch the sweep data to provide the next execution date in case of Intraday account pair sweeps and to fetch account pairs based on frequency in case of EOD/BOD account pair executions.
Batch	LMS_FA_SWEEPDATA_CRE ATE	Create	This functional activity code is used to create the sweep data during structure creation.
Batch	LMS_FA_SWEEPDATA_UPD ATE	Update	This functional activity code is used to update existing sweep data during structure modification.
Batch	LMX_FA_PENDING_AUTH_ VIEW	Authoriz ation View	This functional activity code is used to view the maintenance pending for authorization.
Batch	LMX_FA_HAS_PENDING_A UTH	Pending Authoriz ation	This functional activity code is used to check whether the branch has any pending maintenance for authorization.

# Index

C			
Configure EOD, 1-3	<u> </u>		
Configure Tasks, 4-4 Create Task, 4-1	Mapping Functional Activity Code, 1-1		
E	Oracle Banking Liquidity Management Job, 3-1		
EOD Configuration, <i>1-1</i>			
F	R		
Functional Activity Codes, A-1	Run EOD for branch, 1-4		
I	Upload DSL, 1-1		
Intraday Jobs, 4-1			
J			
Job Definition Naming Convention, 2-1	_		