Oracle® Banking Liquidity Management

Configuration Guide

Release 14.8.0.0.0
G28880-02
April 2025

Oracle Banking Liquidity Management Configuration Guide, Release 14.8.0.0.0

G28880-02

Contents

Preface

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.

Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at Critical Patches, Security Alerts and Bulletins. All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by Oracle Software Security Assurance.

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
boldface	Boldface type indicates graphical user interface elements associated with an 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, code in 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 **Save** to save the details entered or selected in the screen. Saved record details will be available in 'View Screen'
Close	Click **Close** 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 **Cancel** 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 **Next** to navigate to the next data segment, after successfully capturing the data.
Create	Click **Create** to capture the data entered and create the new record. Created record details will be available in 'View Screen'
Back	Click **Back** to navigate to the previous data segment, without lost of any data entered or captured from current screen.
Delete	Click **Delete** to delete the task listed.
Fetch	Click **Fetch** to fetch the EOD details.
Start	Click **Start** to invoke the EOD operation.
Reset	Click **Reset** to clear the EOD records invoked.
Retry	Click **Retry** to restart the EOD operation.
Refresh	Click **Refresh** 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
┘ └ ┐ ┌	Minimize
✕	Close
🔍	Perform Search
⇤	Navigate to the first record
⇥	Navigate to the last record

Table 3   (Cont.) Symbols and Icons - Common

Symbol/Icon	Function
◀	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

> **Note:**
>
> 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

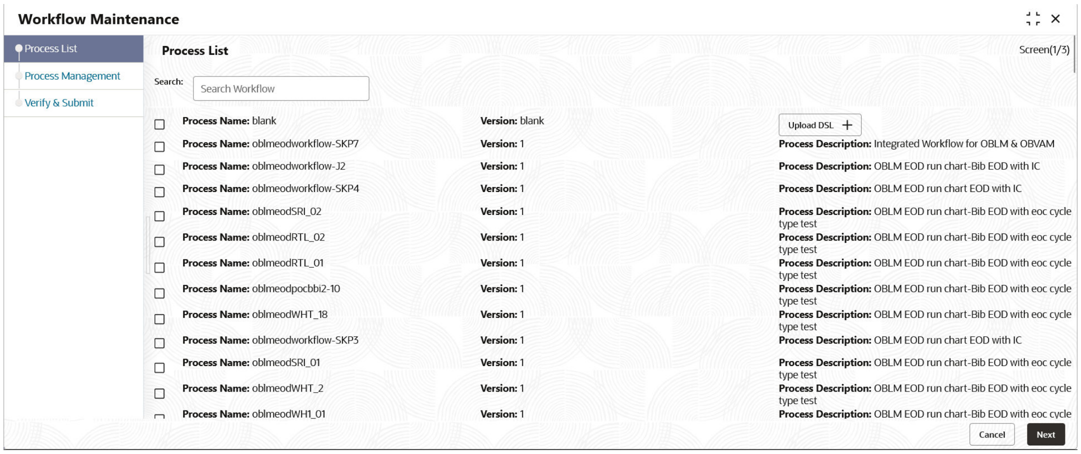
This topic describes the systematic instructions to upload DSL in Business Process maintenance.

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

1. 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.
2. 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



Workflow Maintenance

- Process List
- Process Management
- Verify & Submit

Process List Screen(1/3)

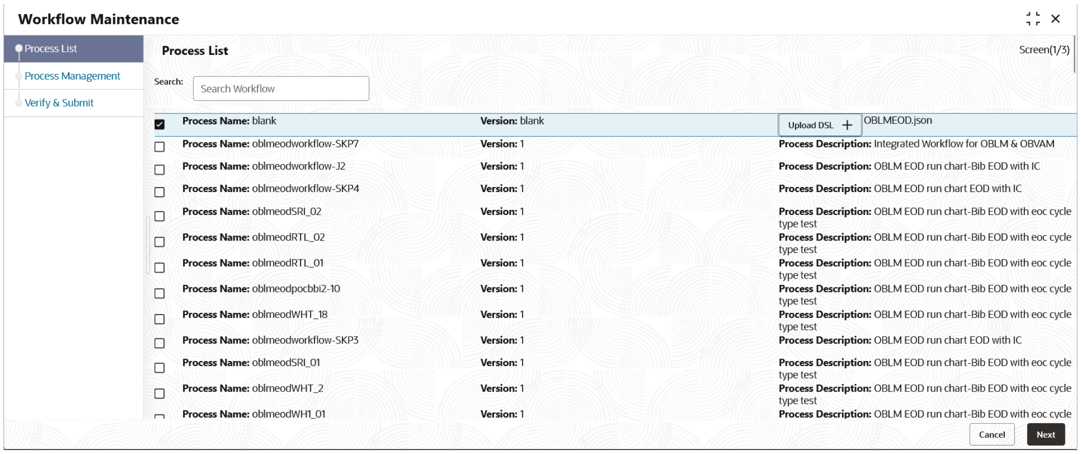
Search: Search Workflow

□	**Process Name:** blank	**Version:** blank	Upload DSL +
□	**Process Name:** oblmeodworkflow-SKP7	**Version:** 1	**Process Description:** Integrated Workflow for OBLM & OBVAM
□	**Process Name:** oblmeodworkflow-J2	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with IC
□	**Process Name:** oblmeodworkflow-SKP4	**Version:** 1	**Process Description:** OBLM EOD run chart EOD with IC
□	**Process Name:** oblmeodSRI_02	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodRTL_02	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodRTL_01	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodpocbbi2-10	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodWHT_18	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodworkflow-SKP3	**Version:** 1	**Process Description:** OBLM EOD run chart EOD with IC
□	**Process Name:** oblmeodSRI_01	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodWHT_2	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodWH1_01	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle

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



Workflow Maintenance

- Process List
- Process Management
- Verify & Submit

Process List Screen(1/3)

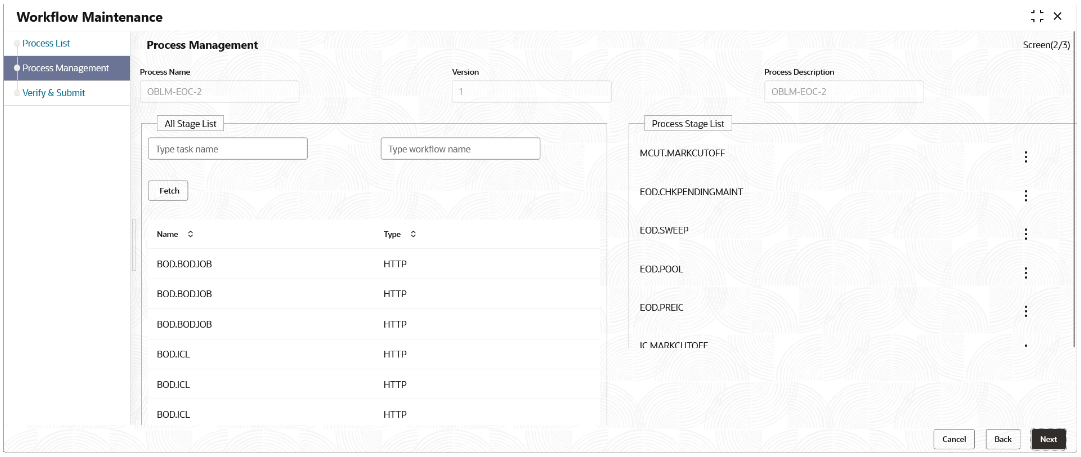
Search: Search Workflow

☑	**Process Name:** blank	**Version:** blank	Upload DSL + OBLMEOD.json
□	**Process Name:** oblmeodworkflow-SKP7	**Version:** 1	**Process Description:** Integrated Workflow for OBLM & OBVAM
□	**Process Name:** oblmeodworkflow-J2	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with IC
□	**Process Name:** oblmeodworkflow-SKP4	**Version:** 1	**Process Description:** OBLM EOD run chart EOD with IC
□	**Process Name:** oblmeodSRI_02	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodRTL_02	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodRTL_01	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodpocbbi2-10	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodWHT_18	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodworkflow-SKP3	**Version:** 1	**Process Description:** OBLM EOD run chart EOD with IC
□	**Process Name:** oblmeodSRI_01	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodWHT_2	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle type test
□	**Process Name:** oblmeodWH1_01	**Version:** 1	**Process Description:** OBLM EOD run chart-Bib EOD with eoc cycle

6. Click **Next** button.

   The **Product Management** screen displays.

Figure 1-3   Process Management



Process Management

Process Name: OBLM-EOC-2
Version: 1
Process Description: OBLM-EOC-2

Name	Type
BOD.BODJOB	HTTP
BOD.BODJOB	HTTP
BOD.BODJOB	HTTP
BOD.ICL	HTTP
BOD.ICL	HTTP
BOD.ICL	HTTP

MCUT.MARKCUTOFF
EOD.CHKPENDINGMAINT
EOD.SWEEP
EOD.POOL
EOD.PREIC
IC.MARKCUTOFF

7. Click **Next** button.

   The **Verify and Submit** screen displays.

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

1.3 Configure EOD

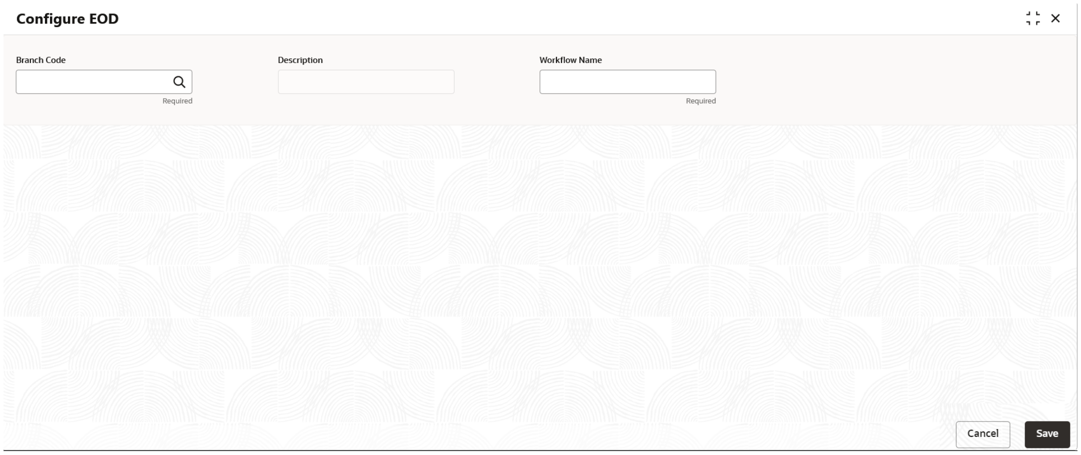
This topic describes the systematic instructions to configure EOD operations

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

1. On **Core Maintenance** menu, under **Branch EOD**, click **Configure EOD**.

   The **Configure EOD** screen displays.

Figure 1-4   Configure EOD



Note:

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

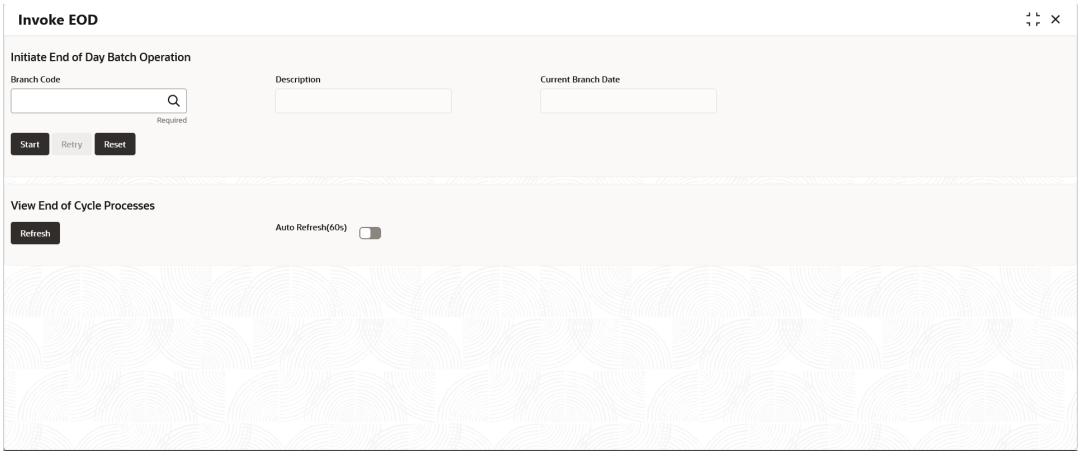
This topic describes the systematic instructions to run the EOD for a branch.

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

1. 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.

2 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"
                     }
                 ]
             }
         }
     },
     "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",
                  "value":"PLATOORCH"
               }
            ]
         }
      },
      "asyncComplete":true
   },
   "startDelay":0,
   "optional":false,
   "asyncComplete":true
}
```

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	-
Headers	userId : <Logged in user id> branchCode : <Logged in branch code> appId : platoorch Content-Type : application/json Accept : application/json	userId – User who updates the task branchCode – Branch where the update is performed.
Body	`{ "workflowInstanceId": "<EOD_Workflow_ID", "taskId": "<Task_ID>", "status": "<Status>"}`	EOD_Workflow_ID – A Workflow ID gets generated when EOD is invoked Task_ID – Unique task ID gets generated for each task once it starts 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.

3 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 Parameters
1	MCUT	markcutoff	Job will check for pending tasks and any existing running process before starting EOD.	BranchCode
2	EOD	CHKPENDINGMAINT	Job will check pending maintenances that required approval.	BranchCode
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: 1. **Modify** - inputParameters.http_request.body.jobName = "eodSweepJobV2" 2. **Add** - inputParameters.http_request.body.jobParameters.key = "gridSize" 3. **Add** - inputParameters.http_request.body.jobParameters.value = 5 (Can be changed as per need)	BranchCode
4	EOD	CLEAR_BALANCE_CACHE	Job will clear the Value Dated balance cache that's populated during EOD Sweep executions.	BranchCode
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: 1. **Modify** - inputParameters.http_request.body.jobName = "eodPoolJobV2" 2. **Add** - inputParameters.http_request.body.jobParameters.key = "gridSize" 3. **Add** - inputParameters.http_request.body.jobParameters.value = 5 (Can be changed as per need)	BranchCode

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

S.No	EOD stage	Job Name	Description	Input Parameters
6	EOD	PREIC	Job will execute the tasks that are required to run before starting Interest batch. The existing EOD.PREIC needs to be updated to incorporate parallel processing for handling large data volumes. 1. **Modify** - inputParameters.http_request.body.jobName = "eodPreICJobV2" 2. **Add** - inputParameters.http_request.body.jobParameters.key = "gridSize" 3. **Add** - inputParameters.http_request.body.jobParameters.value = 5 (Can be changed as per need) 4. **Add** - inputParameters.http_request.body.jobParameters.key = "chunkSize" 5. **Add** - inputParameters.http_request.body.jobParameters.value = 1000 (Can be changed as per need)	BranchCode
7	EOD	IC.MARKCUTOFF	Job will check for pending tasks and any existing running process before starting Interest batch.	BranchCode
8	EOD	OBLM-IC	Job will execute Interest batch.	BranchCode
9	MS-EOFI	MS-EOFI	Milestone for the end of financial input.	BranchCode
10	EOFI	MARKEOFI	Job will mark the end of financial input	BranchCode
11	MS-CHKB4FLIPDATE	MS-CHKB4FLIPDATE	Milestone for date flip.	BranchCode
12	EOD	CMC.DATEFLIP	Job will change system date to next working date in common core.	BranchCode
13	EOD	OBLM.DATEFLIP	Job will change system date to next working date in Oracle Banking Liquidity Management.	BranchCode
14	EOD	RCUT.RELEASECUTOFF	Job will mark release cutoff after EOD.	BranchCode
15	EOD	IC.RELEASECUTOFF	Job will mark release cutoff for IC Batch.	BranchCode

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

S.No	EOD stage	Job Name	Description	Input Parameters
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: 1. **Modify** - inputParameters.http_request.body.jobName = "bodReallocationJobV2" 2. **Add** - inputParameters.http_request.body.jobParameters.key = "gridSize" 3. **Add** - inputParameters.http_request.body.jobParameters.value = 5 (Can be changed as per need)	BranchCode
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: 1. **Modify** - inputParameters.http_request.body.jobName = "bodSweepJobV2" 2. **Add** - inputParameters.http_request.body.jobParameters.key = "gridSize" 3. **Add** - inputParameters.http_request.body.jobParameters.value = 5 (Can be changed as per need)	BranchCode
18	BOD	TI.MARKTI	Job will mark the transaction inputs.	BranchCode
19	BOD	BOD.ICL	Job will execute the ICL.	BranchCode
20	BOD	chargeCalculationJob	Job will calculate the LM charges for given branch code.	BranchCode
21	BOD	chargePostingJob	Job will post the LM charges for given branch code.	BranchCode
22	OBLM	OBLM.UPDATEDEFERREDENTRY	Job will update the Deferred entries for pool contribution for given BranchCode.	BranchCode
23	OBLM	OBLM.MBCVBALANCEFETCH	Job will fetch balance from respective external system for all CM accounts under the EOD branch .	BranchCode
24	OBLM	OBLM.MBCVACCBALARCHIVAL	Job will do archival for balances into balance history table for all CM accounts under the EOD branch.	BranchCode

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

S.No	EOD stage	Job Name	Description	Input Parameters
25	OBLM	OBLM.UPLOADINTERESTACCRUALREPORT	Job will update the Accrual Interest details for given BranchCode.	BranchCode
26	IC	IC.INTRADAYBATCH	Job will update the Intra day batch to notify UBS.	BranchCode
27	OBLM	OBLM.UPDATESTRUCTURESTATUS	Job will update the status of all structures under the EOD branch.	BranchCode
28	OBLM	OBLM.DATEFLIP	Job will update the branch date of the EOD branch in LM table.	BranchCode
29	OBLMIC	OBLMIC.DATEFLIP	Job will update the branch dates of the EOD branch in OBLM IC tables.	BranchCode
30	EOD	CHECKINSTANCE	Job will check the current EOD instance is updated in CMC table CMC_TB_BRANCHEOD_INSTANCES.	BranchCode and WorkflowId

Note:

The **Charge Calculation** (ChargeCalculation) and **Charge Posting** (ChargePosting) jobs are allowed to be configured either before date flip (EOD) or after date flip (BOD). Based on the bank's requirement, this can be configured. By default, these jobs are configured after date flip (BOD) and is shipped with EOD configuration.

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

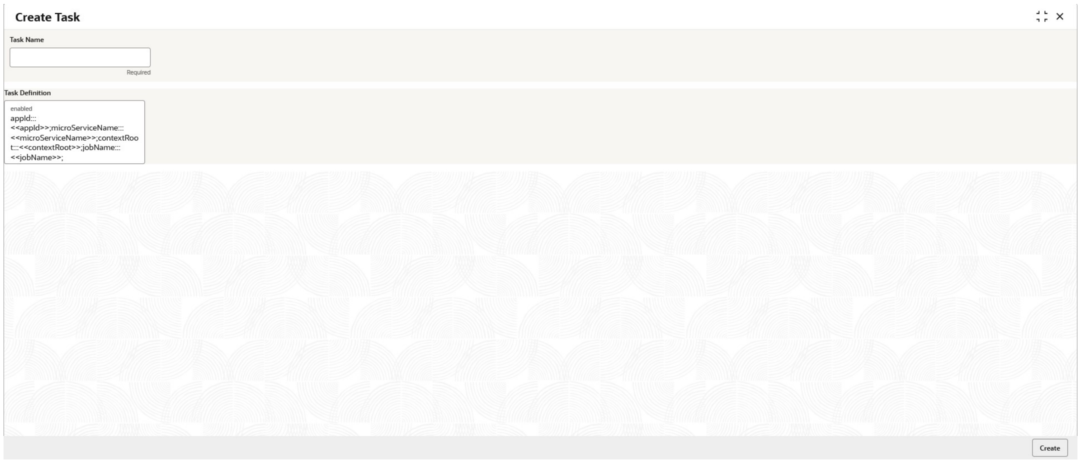
This topic describes the systematic instructions to create the task.

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



Task Definition

enabled

appId:::<<appId>>;microServiceName:::<<microServiceName>>;contextRoot:::<<contextRoot>>;jobName:::<<jobName>>;

> **Note:**
>
> The fields marked as **Required** are mandatory.

2. Specify the values mentioned in the following table.

Table 4-1   Intraday Job - Task Values

Sl. no	Task Name	Description	Task Definition
1	OBLM_intraDayAccountPairSweepJob_INT_001	This task initiates Intraday Account Pair sweep execution	appId:::LMS;microServiceName:::oblm-sweep-services;contextRoot:::oblm-sweep-services;type:::schedule;jobName:::intraDayAccountPairSweepJob;cronExpression:::0 0/5 * * * ?; **Note:** If parallel Processing strategy should be used during intraday execution, following Task Definition is to be used. appId:::LMS;microServiceName:::oblm-sweep-services;contextRoot:::oblm-sweepservices;type:::schedule;jobName:::intraDayAccount PairSweepJobV2;cronExpression:::0 0/5 * * * ?;
2	OBLM_intraDayStructureSweepJob_INT_002	This task initiates Intraday Structure sweep execution	appId:::LMS;microServiceName:::oblm-sweep-services;contextRoot:::oblm-sweep-services;type:::schedule;jobName:::intraDayStructureSweepJob;cronExpression:::0 0/5 * * * ?; **Note:** If parallel Processing strategy should be used during intraday execution, following Task Definition is to be used. appId:::LMS;microServiceName:::oblm-sweep-services;contextRoot:::oblm-sweepservices;type:::schedule;jobName:::intraDayStructure SweepJobV2;cronExpression:::0 0/5 * * * ?;
3	OBLM_processMTHoldMessagesJob_INT_003	This task processes MT Messages that are on hold	appId:::LMG;microServiceName:::oblm-messaging-services;contextRoot:::oblm-messaging-services;type:::schedule;jobName:::processMTHoldMessagesJob;cronExpression:::0 0/5 * * * ?;
4	OBLM_pendingPaymentsJob_INT_004	This task retriggers the sweep handoffs that are in pending state	appId:::LMX;microServiceName:::oblm-integration-services;contextRoot:::oblm-integration-services;type:::schedule;jobName:::pendingPaymentsJob;cronExpression:::0 0/5 * * * ?;
5	OBLM_publishEventsLogJob_INT_005	This task publishes the events to Kafka	appId:::LMX;microServiceName:::oblm-integration-services;contextRoot:::oblm-integration-services;type:::schedule;jobName:::publishEventsLogJob;cronExpression:::0 0/10 * * * ?;

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

Sl. no	Task Name	Description	Task Definition
6	OBLM_pendingReallocationJob_INT_006	This task retriggers the reallocation handoffs that are in pending state	appId:::LMX;microServiceName:::oblm-integration-services;contextRoot:::oblm-integration-services;type:::schedule;jobName:::pendingReallocationJob;cronExpression:::0 0/10 * * * ?;
7	sweepEventsFetch	This task is to fetch and populate the sweep events to dashboard schema	appId:::LMD;microServiceName:::oblm-dashboard-services;contextRoot:::oblm-dashboard-services;type:::schedule;jobName:::sweepLMDEventsJob;cronExpression:::0 0/5 * * * ?'
8	savetoChargeCalCollJob_LM	This task is needed for Charges, Only If LM is deployed	appId:::VAMLMCHG;microServiceName:::vamlm-charge-services;contextRoot:::vamlm-charge-services;jobName:::savetoChargeCalCollJob;appCode:::LMCHG;
9	chargePostingJob_LM	This task is needed for Charges, Only If LM is deployed	appId:::VAMLMCHG;microServiceName:::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	appId:::VAMLMCHG;microServiceName:::vamlm-charge-services;contextRoot:::vamlm-charge-services;jobName:::chargeCalculationJob;appCode:::LMCHG;
11	savetoChargeCalCollJob_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.	appId:::VAMLMCHG;microServiceName:::vamlm-charge-services;contextRoot:::vamlm-charge-services;jobName:::savetoChargeCalCollJob;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	appId:::VAMLMCHG;microServiceName:::vamlm-charge-services;contextRoot:::vamlm-charge-services;jobName:::chargePostingJob;appCode:::VAMCHG;

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

Sl. no	Task Name	Description	Task Definition
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.	appId:::VAMLMCHG;microServiceName:::vamlm-charge-services;contextRoot:::vamlm-charge-services;jobName:::chargeCalculationJob;appCode:::VAMCHG;
14	platoOrchArchivejob	This task is needed for Purge, Conductor related tables.	appId:::LMX;microServiceName:::oblm-integration-services;contextRoot:::oblm-integration-services;jobName:::platoOrchArchivejob;purgeConfigName:::<<purgeConfigName>>;Plato Orch Archival Instructions

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

4.2 Configure Tasks

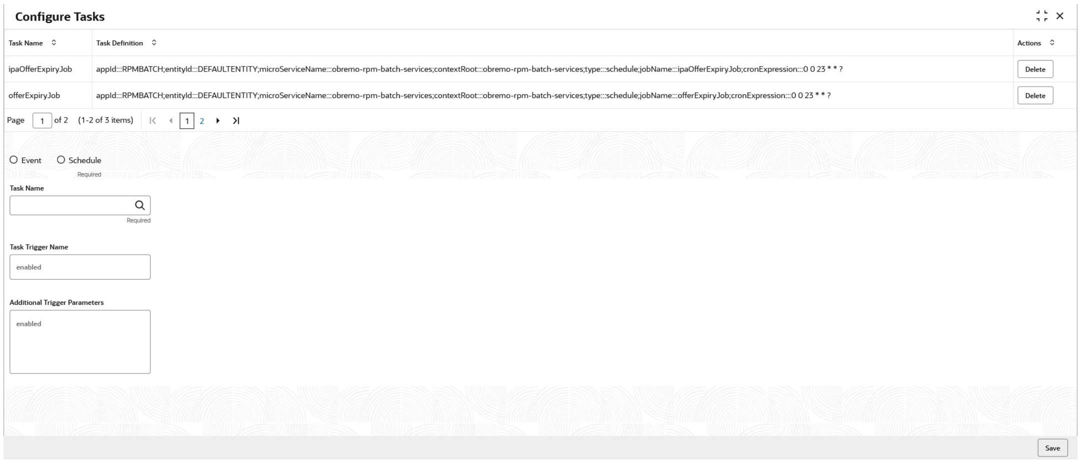
This topic describes the systematic instructions to configure the tasks.

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



Configure Tasks

Task Name	Task Definition	Actions
ipaOfferExpiryJob	appId:::RPMBATCH;entityId:::DEFAULTENTITY;microServiceName:::obremo-rpm-batch-services;contextRoot:::obremo-rpm-batch-services;type:::schedule;jobName:::ipaOfferExpiryJob;cronExpression:::0 0 23 * * ?	Delete
offerExpiryJob	appId:::RPMBATCH;entityId:::DEFAULTENTITY;microServiceName:::obremo-rpm-batch-services;contextRoot:::obremo-rpm-batch-services;type:::schedule;jobName:::offerExpiryJob;cronExpression:::0 0 23 * * ?	Delete

Page 1 of 2 (1-2 of 3 items) 1 2

○ Event ○ Schedule
Required

Task Name
Required

Task Trigger Name
enabled

Additional Trigger Parameters
enabled

Save

Note:

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.

A Functional Activity Codes

Table A-1 List of Functional Activity Codes

Screen Name	Functional Activity Code	Action	Purpose
Batch	LMS_FA_SWEEPDATA_VIEW	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_CREATE	Create	This functional activity code is used to create the sweep data during structure creation.
Batch	LMS_FA_SWEEPDATA_UPDATE	Update	This functional activity code is used to update existing sweep data during structure modification.
Batch	LMX_FA_PENDING_AUTH_VIEW	Authorization View	This functional activity code is used to view the maintenance pending for authorization.
Batch	LMX_FA_HAS_PENDING_AUTH	Pending Authorization	This functional activity code is used to check whether the branch has any pending maintenance for authorization.

Index