Oracle® Banking Liquidity Management Cloud Service

Configuration Guide

Release 14.8.2.0.0
G54667-01
April 2026

ORACLE

Oracle Banking Liquidity Management Cloud Service Configuration Guide, Release 14.8.2.0.0

G54667-01

Copyright © 2023, 2026, 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

Preface

Purpose i
Audience i
Documentation Accessibility i
Diversity and Inclusion i
Related Resources ii
Conventions ii
Screenshot Disclaimer ii
Acronyms and Abbreviations ii
Basic Actions ii
Symbols and Icons iii

1 EOD Configuration

1.1 Mapping Functional Activity Code 1
1.2 Upload DSL 1
1.3 Configure EOD 3
1.4 Run EOD for Branch 4

2 Job Definition Naming Convention

3 Oracle Banking Liquidity Management Job

4 Intraday Jobs

4.1 Create Task 1
4.2 Configure Tasks 4

A Functional Activity Codes

Copyright © 2023, 2026, Oracle and/or its affiliates.

Index

Copyright © 2023, 2026, Oracle and/or its affiliates.

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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

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 user supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that user 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'

Copyright © 2023, 2026, Oracle and/or its affiliates.

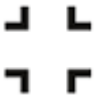
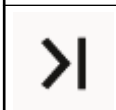
Table 2 (Cont.) Basic Actions

Actions	Description
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

Copyright © 2023, 2026, Oracle and/or its affiliates.

Table 3 (Cont.) Symbols and Icons - Common

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

Copyright © 2023, 2026, Oracle and/or its affiliates.

1

EOD Configuration

This topic provide information about the EOD Configuration process.

This topic contains the following sub-topics:

- 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

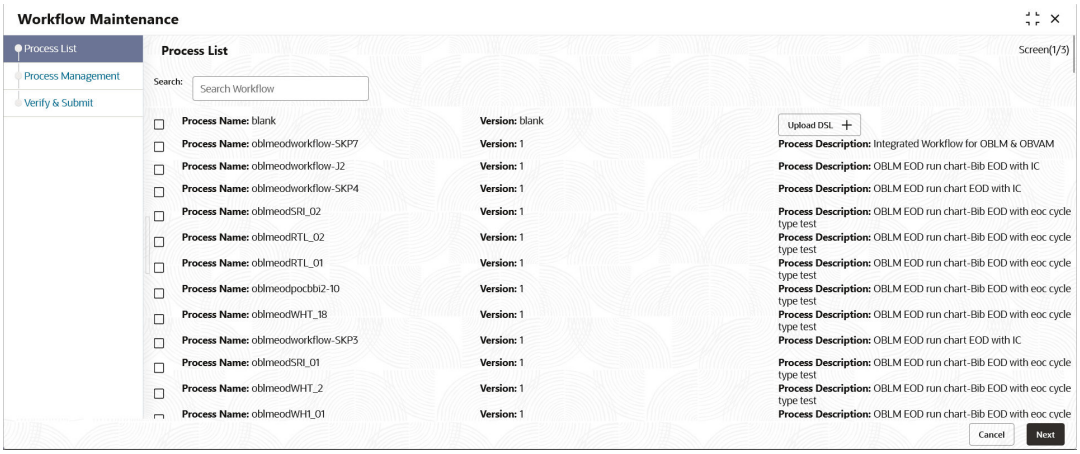
This topic describes the systematic instructions to upload DSL in Business Process maintenance.

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 **Process List** screen displays.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 1-1 Process List



3. Specify the fields on **Process List** screen.

   For more information on fields, refer to the field description table.

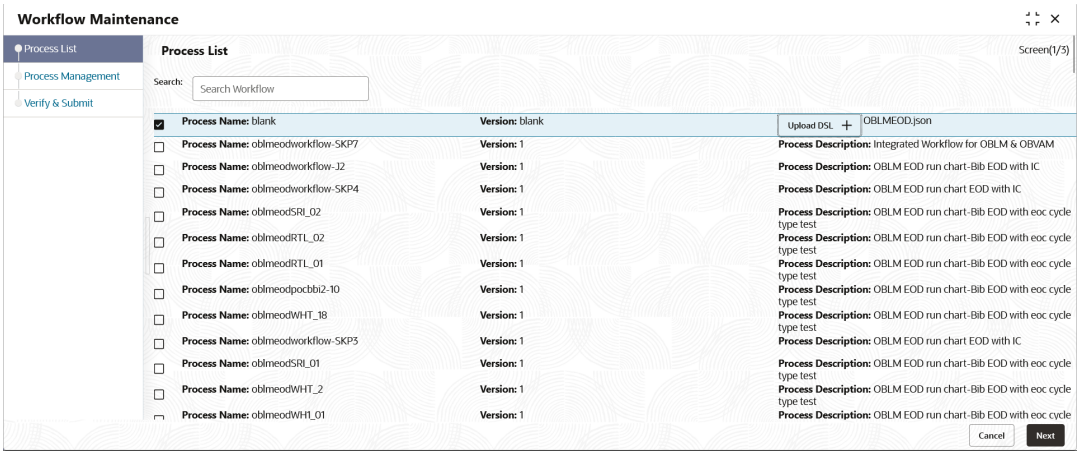
Table 1-1 Process List - Field Description

Field	Description
Search	Specify the search criteria to search the workflow.
Process Name	Displays the process name to be selected for the workflow maintenance.
Version	Displays the version of the process.
Process Description	Displays the description of the process workflow.
Region Code	Displays the region code of the process.

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

   The **Process List – Upload DSL** screen displays

Figure 1-2 Process List – Upload DSL

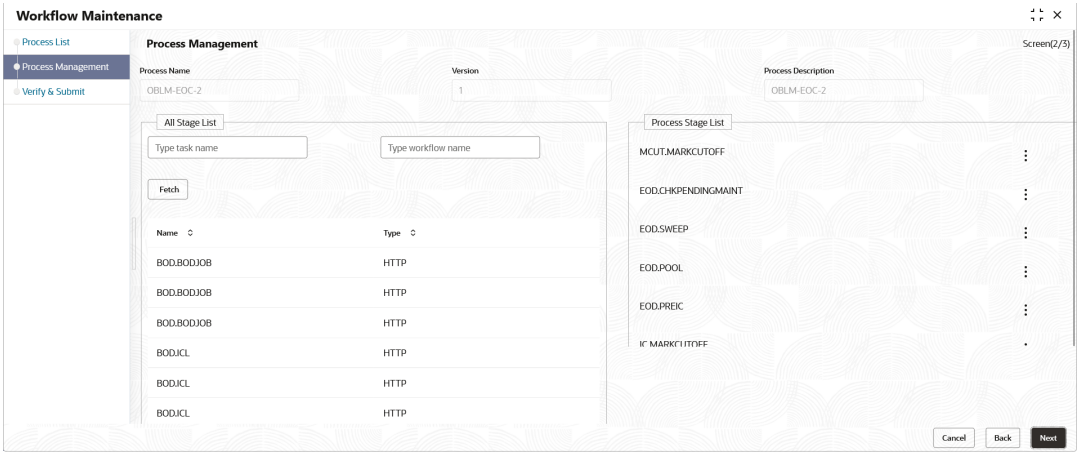


Copyright © 2023, 2026, Oracle and/or its affiliates.

7. Click **Next** button.

   The **Process Management** screen displays.

Figure 1-3 Process Management



For more information on fields, refer to the field description table.

Table 1-2 Process List - Field Description

Field	Description
Process Name	Displays the process name to be selected for the workflow maintenance.
Version	Displays the version of the process.
Process Description	Displays the description of the process workflow.
All Stage List	This section displays the list of all stages of the process.
Type task name	Specify the task name of the process.
Type workflow Name	Specify the workflow name of the process.
Name	Displays the task name of the process.
Type	Displays the workflow name of the process.
Process Stage List	This section displays the process stage list.

8. Click **Next** button.

   The **Verify and Submit** screen displays.

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

1.3 Configure EOD

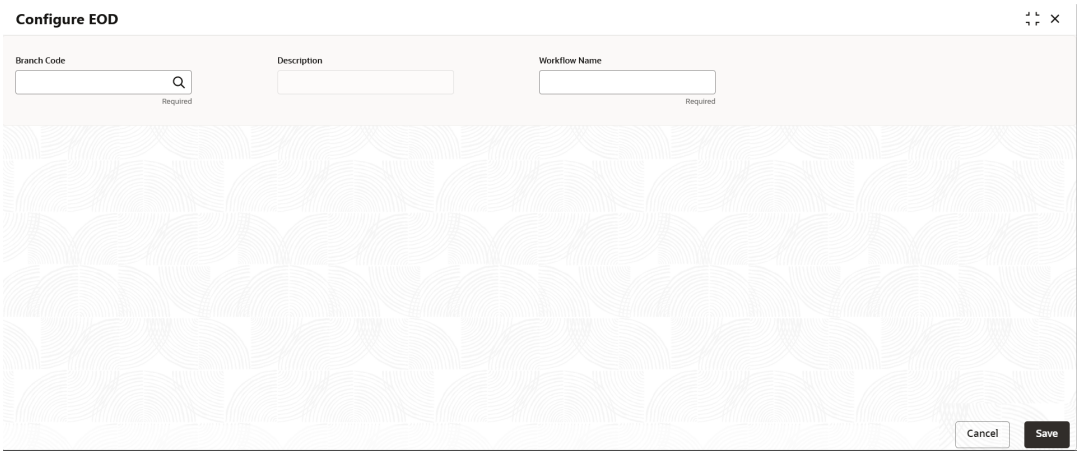
This topic describes the systematic instructions to configure EOD operations

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

   The **Configure EOD** screen displays.

Copyright © 2023, 2026, Oracle and/or its affiliates.

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. Specify the fields on **Configure EOD** screen.

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

For more information on fields, refer to the field description table.

Table 1-3 Configure EOD - Field Description

Field	Description
Branch Code	Specify or click **Search** icon and select the branch code that is associated with the logged in user.
Description	Displays the description of the branch, based on the Branch code selected.
Workflow Name	Specify the work-flow name that is already created.

3. Click **Search** icon to view and select the **Branch Code**.
4. Specify the work-flow name that is already created.
5. Click **Save** to save the details

1.4 Run EOD for Branch

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

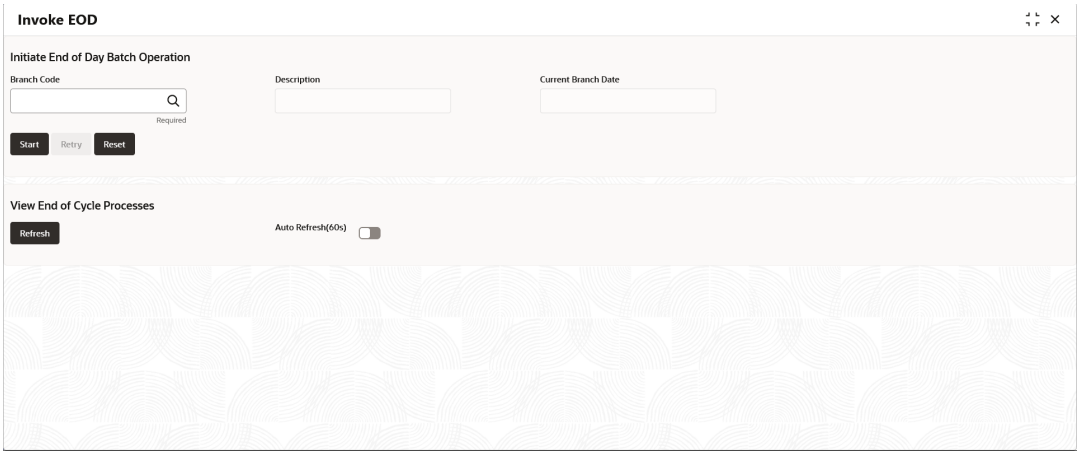
1. On **Home** Screen, click **Core Maintenance**. Under **Core Maintenance** menu, click **Branch EOD**.

Copyright © 2023, 2026, Oracle and/or its affiliates.

2. Under **Branch EOD**, click **Invoke EOD**.

   The **Invoke EOD** screen displays.

Figure 1-5 Invoke EOD



3. Specify the fields on **Invoke EOD** screen.

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

For more information on fields, refer to the field description table.

Table 1-4 Invoke EOD - Field Description

Field	Description
Branch Code	Specify or click **Search** icon and select the branch code that is associated with the logged in user.
Description	Displays the description of the branch, based on the Branch code selected.
Current Branch Date	Displays the current branch date, based on the Branch code selected.
View End of Cycle Processes	This section displays View End of Cycle Processes related fields.
Auto Refresh(60s)	Switch on the toggle to enable auto refresh.

4. Click **Search** icon to view and select the branch code to run EOD.
5. Click **Start** to invoke EOD for selected branch
6. Click **Retry** to restart the EOD work-flow from the failed task.
7. Click **Reset** to clear the branch selected.
8. Click **Refresh** to view the current status of the branch.

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

Copyright © 2023, 2026, Oracle and/or its affiliates.

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",
```


Copyright © 2023, 2026, Oracle and/or its affiliates.

```
                    "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":{
```


Copyright © 2023, 2026, Oracle and/or its affiliates.

```
        "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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
5	EOD	POOL	Job will execute non-main pool (i.e. subPool is 'S' or null) structures. The workflow JSON is revised for Plato Batch framework adoption. The job name was changed from "eodPoolJob" to "OBLM_POOL_EOD_001". • The "branchCode" parameter was removed. • New parameters were introduced: – "invokeBatchFrm": "Y" triggers the Plato Batch framework logic instead of the core logic. – "isDynamicParamReq": "Y" signals that dynamic parameters, such as branchCode and runDate, will be fetched from the domain service. – "isCallback": "Y" indicates that a callback is required. "callbackType": "PLATOORCH" specifies that the callback should be sent to the PLATO Orchestration service for task status updates. – "subPool": "S" indicates that Pool Batch will be processing the NonMain Pool Structures. • Batch-related parameters and configurations—such as stream size, commit frequency, driver table page size, and skip policy—are now controlled through entries in the PLATO_BATCH_TM_JOB_DETAILS table in the batch schema. The relevant record uses JOB_NAME = 'OBLM_POOL_EOD_001'. These changes align the workflow configuration with the Plato Batch framework's standards, supporting enhanced integration, dynamic parameter management, centralized batch configuration, and streamlined callback handling.	BranchCode subPool
6	EOD	MAINPOOL	Job will execute main pool(i.e. subPool is 'M') structures. Other job details would be same as above POOL Job except subPool parameter value	BranchCode subPool

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
7	EOD	EOD.BALANCEUPDATE	Job Fetches the balance for all the accounts of the branch. If there are any BVT transactions on the current branch date, the job will update the account balance from the BVT value date and also marks Recalc in IC. Performs End-Of-Day balance updates for all eligible accounts in the branch. It fetches latest balances, updates batch driver tables, and triggers further downstream calculations or processes as needed. Job Parameters: • workflowId`: Unique workflow execution identifier. • `referenceTaskName`: Should be set to `EOD.BALANCEUPDATE` for traceability. • `microServiceName`: Service name for routing; typically `oblm-integration-services`. • `contextRoot`: REST API root; typically `oblm-integration-services`. • `appId`: Application code; for OBLM, always `LMX`. • `invokeBatchFrm`: Always `Y` (true); indicates batch mode invocation. • `isDynamicParamReq`, `isCallback`: Always `Y` (true); indicates batch mode invocation. • `callbackType`: Always 'PLATOORCH'	BranchCode

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
8	EOD	EOD.INTEREST-OPTIMIZATION	Executes Interest Optimization SDE (System Defined Elements) batch processing for the branch, updating batch data and writing interest optimization results as required by day-end routines. Runs advanced interest allocation/ optimization for all structures and linked accounts for financial close. Job Parameters: • `workflowId`: Workflow execution ID. • `referenceTaskName`: Should be `EOD.INTEREST-OPTIMIZATION`. • `microServiceName`, `contextRoot`, `appId`: As above; always `oblm-integration-services`, `LMX`. • `appId`: Application code; for OBLM, always `LMX`. • `invokeBatchFrm`: Always `Y` (true); indicates batch mode invocation. • `isDynamicParamReq`, `isCallback`: Always `Y` (true); indicates batch mode invocation. • `callbackType`: Always 'PLATOORCH'	BranchCode
9	EOD	EOD.RATIO	Processes ratio method SDE calculations at End-Of-Day, aggregating and storing critical ratio metrics for branch accounts. Feeds results for pool/ structure analytics and compliance reporting. Job Parameters: • `workflowId`: ID for this workflow instance. • `referenceTaskName`: Should be `EOD.RATIO`. • `microServiceName`, `contextRoot`, `appId`: Service/application routing; generally `oblm-integration-services`, `LMX`. • `appId`: Application code; for OBLM, always `LMX`. • `invokeBatchFrm`: Always `Y` (true); indicates batch mode invocation. • `isDynamicParamReq`, `isCallback`: Always `Y` (true); indicates batch mode invocation. • `callbackType`: Always 'PLATOORCH'	BranchCode

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
9	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

Copyright © 2023, 2026, Oracle and/or its affiliates.

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) • Job Name: Old: "bodReallocationJob" New: "OBLM_REALLOCATION_009" (standardized, versioned) - Parameters: Old: Fewer, only basics (`branchCode`, `appId`, etc.) New: Adds workflow/context (`workflowId`, `referenceTaskName`), orchestration flags (`invokeBatchFrm`), dynamic params (`isDynamicParamReq`), and callback setup (`isCallback`, `callbackType`). - Purpose: The new DSL is designed for use with orchestration platforms, allowing traceability, callbacks, dynamic params, and cleaner job versioning. Old DSL is for legacy direct batch launch.	BranchCode

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
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. - Job Name: Old: `"bodICLJob"` New: `"OBLM_ICL_010"` (versioned, standardized) - Parameters Added: New version adds: - `workflowId` - `referenceTaskName` - `invokeBatchFrm` - `isDynamicParamReq` - `isCallback` - `callbackType` *(All above enable orchestration, dynamic params, and callback support.)* - Purpose: New DSL is orchestration-ready and supports callback & dynamic execution, while the old one is static and direct.	BranchCode
20	BOD	chargeCalculationJob	Job will calculate the LM charges for given branch code.	BranchCode

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
21	BOD	chargePostingJob	Job will post the LM charges for given branch code.	BranchCode

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
22	OBLM	OBLM.UPDATEDEFERREDENTRY	Job will update the Deferred entries for pool contribution for given BranchCode. 1. Integration Type Changed: - Old: Called OBLM-POOL-SERVICES directly with a GET method to a REST endpoint. - New: Now handled by submitting a batch job to PLATO-BATCH-SERVER using a POST. 2. URL Changed: - Old: `/oblm-pool-services/web/v14.8.2.0.0/pools/deferred-entries/by-branch?branchCode={branchCode}` (REST GET) - New: `/plato-batch-server/jobLauncher/launch/` (Job orchestration POST) 3. Request Method Changed: - Old: GET - New: POST (with job parameters in the body) 4. Headers Changed: - Old: `appId: LMP`, plus branchCode and userId - New: `appId: PLATOBATCH`, plus branchCode, userId, and eocType in headers 5. Request Body: - Old: No request body (REST GET call) - New: Body specifies a batch job with: - `jobName: OBLM_UPDATE_DEF_ENTRIES_005` - `jobParameters`: workflowId, referenceTaskName ("OBLM-UPDATEDEFERREDENTRY"), microServiceName, contextRoot, appId,	BranchCode

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
			invokeBatchFrm, isDynamicParamReq, isCallback, callbackType 6. Async Execution: - Old: Synchronous (`asyncComplete: false`) - New: Asynchronous (`asyncComplete: true`) 7. Purpose of Change: - Deferred entry update was a synchronous direct fetch from the pool service. Now, it's performed as an orchestrated, asynchronous batch job as part of the EOD process, with callback and orchestration support.	

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
23	OBLM	OBLM.MBCVBALANCEFETCH	Job will fetch balance from respective external system for all CM accounts under the EOD branch . Service Endpoint Changed: - Old: Called `PLATO-BATCH-SERVER` with jobName `mbcvBalanceUpdateJob`. - New: Still calls `PLATO-BATCH-SERVER` but now uses jobName `OBLM_CM_ACCBAL_REQ_007`. Job Name Changed: - Old: `"jobName": "mbcvBalanceUpdateJob"` - New: `"jobName": "OBLM_CM_ACCBAL_REQ_007"` jobParameters Updated: - Old: - branchCode - balanceRequestType: "Internal" - microServiceName: oblm-integration-services - contextRoot: oblm-integration-services - appId: LMX - New: - workflowId - referenceTaskName ("OBLM.MBCVBALANCEFETCH") - microServiceName: oblm-integration-services - contextRoot: oblm-integration-services - appId: LMX - invokeBatchFrm: "Y" - isDynamicParamReq: "Y" - isCallback: "Y" - callbackType: "PLATOORCH" > *Notably, jobParameters have been extended and standardized to match	BranchCode

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
			other PLATO batch jobs, adding orchestration/callback info and dropping `balanceRequestType`.* Operation Type Changed: - Old: Synchronous batch job (`asyncComplete: false`) - New: Asynchronous batch job (`asyncComplete: true`) Purpose of Change: - The fetch of MBCV balance is now submitted as a standardized asynchronous plato batch framework job, with additional orchestration and callback parameters, for better integration within EOD processing.	

Copyright © 2023, 2026, Oracle and/or its affiliates.

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. Integration Type Changed: - Old: Called OBLM-POOL-SERVICES directly with a GET method to a REST endpoint. - New: Now handled by submitting a batch job to PLATO-BATCH-SERVER using a POST. URL Changed: - Old: `/oblm-pool-services/web/v14.8.2.0.0/pools/deferred-entries/by-branch?branchCode={branchCode}` (REST GET) - New: `/plato-batch-server/jobLauncher/launch/` (Job orchestration POST) Request Method Changed: - Old: GET - New: POST (with job parameters in the body) Headers Changed: - Old: `appId: LMP`, plus branchCode and userId - New: `appId: PLATOBATCH`, plus branchCode, userId, and eocType in headers Request Body: - Old: No request body (REST GET call) - New: Body specifies a batch job with: - `jobName: OBLM_UPDATE_DEF_ENTRIES_005` - `jobParameters`: workflowId, referenceTaskName ("OBLM-UPDATEDEFERREDENTRY"), microServiceName, contextRoot, appId, invokeBatchFrm, isDynamicParamReq, isCallback, callbackType Async Execution:	BranchCode

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
			- Old: Synchronous (`asyncComplete: false`) - New: Asynchronous (`asyncComplete: true`) Purpose of Change: - Deferred entry update was a synchronous direct fetch from the pool service. Now, it's performed as an orchestrated, asynchronous batch job as part of the EOD process, with callback and orchestration support.	
26	IC	IC.INTRADAYBATCH	Job will update the Intra day batch to notify UBS.	BranchCode

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
27	OBLM	OBLM.UPDATESTRUCTURESTATUS	Job will update the status of all structures under the EOD branch. Service Endpoint Changed: - Old: Called directly on `OBLM-STRUCTURE-SERVICES` using PUT method. - New: Now calls `PLATO-BATCH-SERVER` as a batch job using POST. URL Changed: - Old: `/oblm-structure-services/web/v14.8.2.0.0/structures/structureStatus/{branchCode}` (direct REST update) - New: `/plato-batch-server/jobLauncher/launch/` (batch job submission) Operation Type Changed: - Old: Synchronous update (the call waits for a direct response from service). - New: Asynchronous batch job (system will process and callback; not immediate). Request Payload Changed: - Old: No request body, just parameterized URI and headers. - New: Has a batch job request body, specifying a job name (`OBLM_STR_STATUS_UPDATE_008`) and multiple parameters (like workflowId, task name, microServiceName, contextRoot, appId, callback type, etc). AppId and Header Changes: - Old: `appId` was "LMA" - New: `appId` is now "PLATOBATCH" (header), but the job body includes `appId: LMX`. Purpose of Change: - The update is now handled via plato-batch-framework orchestrator, integrating structure status updates into broader EOD batch workflows, improving manageability and allowing more control/features (like retries, orchestration, async callbacks).	BranchCode

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

S.No	EOD stage	Job Name	Description	Input Parameters
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 Workflowld

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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

4 Intraday Jobs

This topic provide information about the Intraday Jobs.

This topic contains the following sub-topics:

- 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

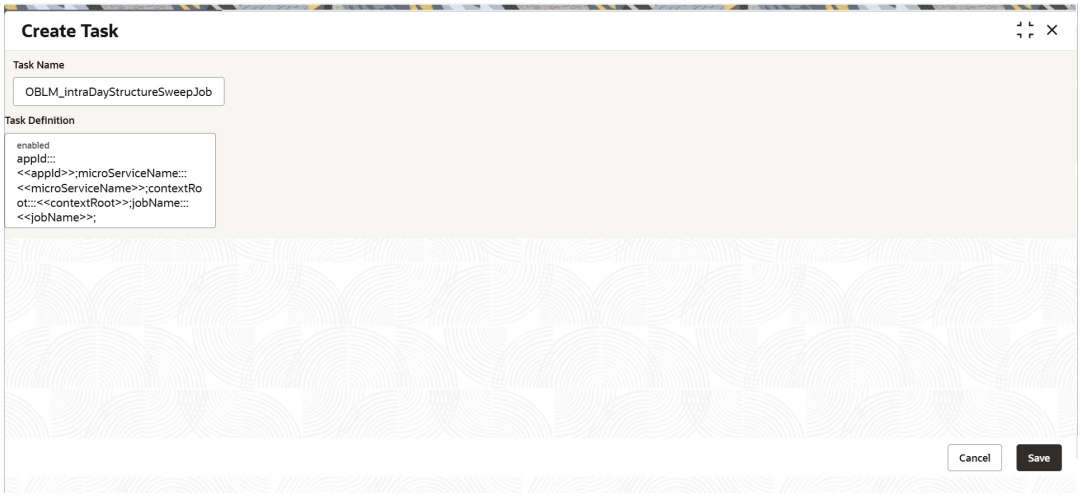
This topic describes the systematic instructions to create the task.

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



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

2. Specify the values mentioned in the following table.

Copyright © 2023, 2026, Oracle and/or its affiliates.

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 * * * ?;

Copyright © 2023, 2026, Oracle and/or its affiliates.

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;

Copyright © 2023, 2026, Oracle and/or its affiliates.

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
15	mbcvBalanceUpdateJob	This task is used to fetch balances for accounts to populate into CM Balance tables used for MBCV	appId:::LMX;microServiceName:::oblm-integration-services;contextRoot:::oblm-integration-services;jobName:::mbcvBalanceUpdateJob;branchCode:::<Branch_code>;balanceRequestType:::<balance_request_type>;crDrFloorLimitIndicator:::<Indicator>;camt060Version:::<camt_version>;senderBic:::<sender_bic>;
16	thirdPartyAccountBalanceFetchTriggerJob	This task is used to fetch balances of the third party accounts that are scheduled with frequencies specified at third party account level and pre sweep	appId:::LMX;microServiceName:::oblm-integration-services;contextRoot:::oblm-integration-services;jobName:::thirdPartyAccountBalanceFetchTriggerJob;

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.

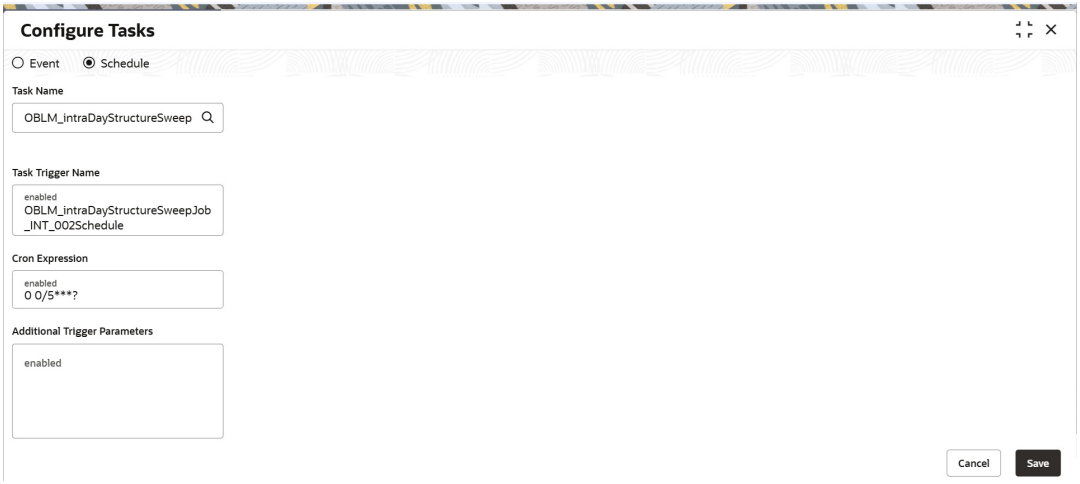
The Configured intra-day jobs will get triggered as per the specified Cron Expression, for the Create Task the scheduler needs to be configured as shown as follows.

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

   The **Configure Tasks** screen displays.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 4-2 Configure Tasks



> **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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Index

C

Configure EOD, *3*
Configure Tasks, *4*
Create Task, *1*

E

EOD Configuration, *1*

F

Functional Activity Codes, *A-1*

I

Intraday Jobs, *1*

J

Job Definition Naming Convention, *1*

M

Mapping Functional Activity Code, *1*

O

Oracle Banking Liquidity Management Job, *1*

R

Run EOD for branch, *4*

U

Upload DSL, *1*

Copyright © 2023, 2026, Oracle and/or its affiliates.