

Configuration Guide

Oracle Banking Liquidity Management

Release 14.5.4.0.0

Part Number F54228-01

February 2022

Configuration Guide

Oracle Financial Services Software Limited
Oracle Park
Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax: +91 22 6718 3001

<https://www.oracle.com/industries/financial-services/index.html>

Copyright © 2018, 2022, Oracle and/or its affiliates. All rights reserved.

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 failsafe, 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.

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.

This software or hardware and documentation may provide access to or information on 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. 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.

Contents

1	Preface	1
1.1	Background	1
1.2	Introduction.....	1
1.3	Document Accessibility	1
1.4	Acronyms, Abbreviations and Definitions	1
1.5	Related Documents.....	1
2	EOD Configuration	2
2.1	Mapping Functional Activity Code.....	2
2.2	Upload DSL	2
2.3	Configure EOD	4
2.4	Steps to run EOD for a branch.....	5
3	Job definition Naming Convention	6
4	Oracle Banking Liquidity Management Job	12
5	Intraday Jobs.....	14
5.1	Create Task.....	14
5.2	Configure Tasks	15
6	Functional Activity Codes	16

1 Preface

1.1 Background

Oracle Banking Liquidity Management allows you to execute several functions every day on a routine basis as part of the End of Day (EOD) process. These functions can be run at various stages of the EOD process.

1.2 Introduction

The End of Day process is to tie up all the operations for a financial day and prepare the system for the next day. The EOD process should be defined for a branch and executed separately for each branch. When the process is running, you could choose to monitor it from Invoke EOD screen.

EOD uses Oracle Banking Microservice Architecture Orchestrator and Batch service for orchestrating all the jobs required to complete End of Day processing. This document helps in the required set up to run EOD.

1.3 Document Accessibility

1. Oracle Banking Microservice Architecture Orchestrator needs to be deployed as per installation guide.
2. Oracle Banking Microservice Architecture Batch needs to be deployed as per installation guide.

1.4 Acronyms, Abbreviations and Definitions

Acronyms	Definition
EOD	End of Day

1.5 Related Documents

The related documents are as follows:

- Oracle Banking Common Core User Guide
- Oracle Banking Liquidity Management User Guide
- Oracle Banking Liquidity Management Data Migration User Guide

2 EOD Configuration

2.1 Mapping Functional Activity Code

The following Functional Activities needs to be maintained in user's role to perform EOD operations

CMC_FA_BRANCH_EOD_PROCESS

2.2 Upload DSL

1. Save the below attachment to local folder. 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.



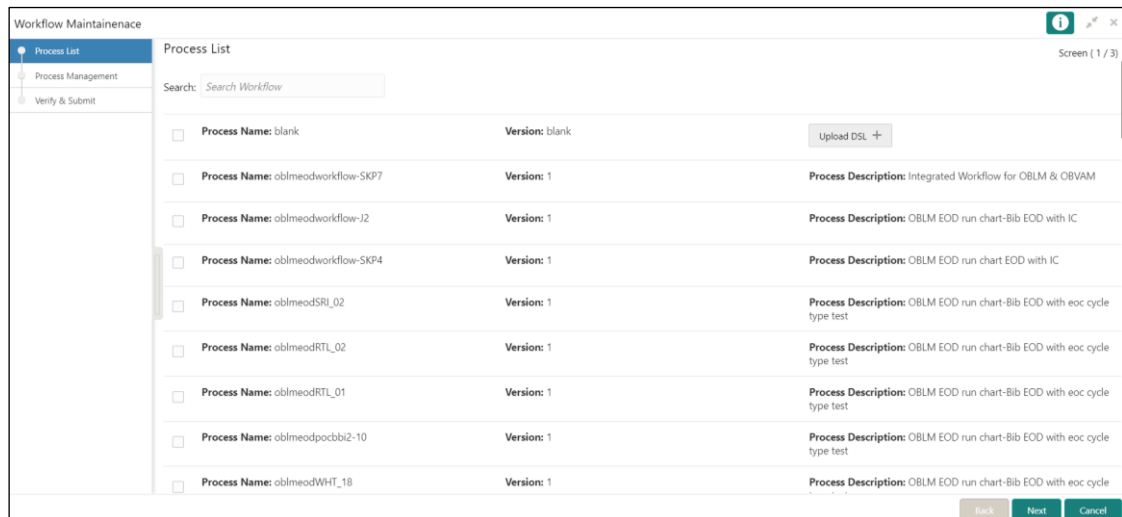
OBLMEOD.json

Note: Refer to PDF attachments for downloading JSON file.

2. On **Home Screen**, under **Tasks** menu, click **Business Process Maintenance** to import, create or modify batch process definition.

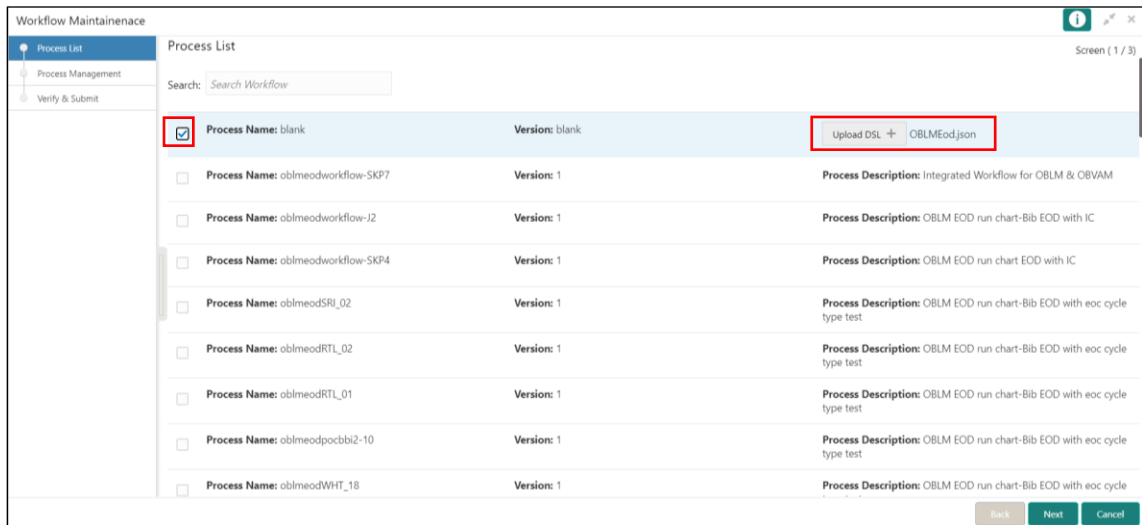
→ The **Product List** screen is displayed.

Figure 1: Process List



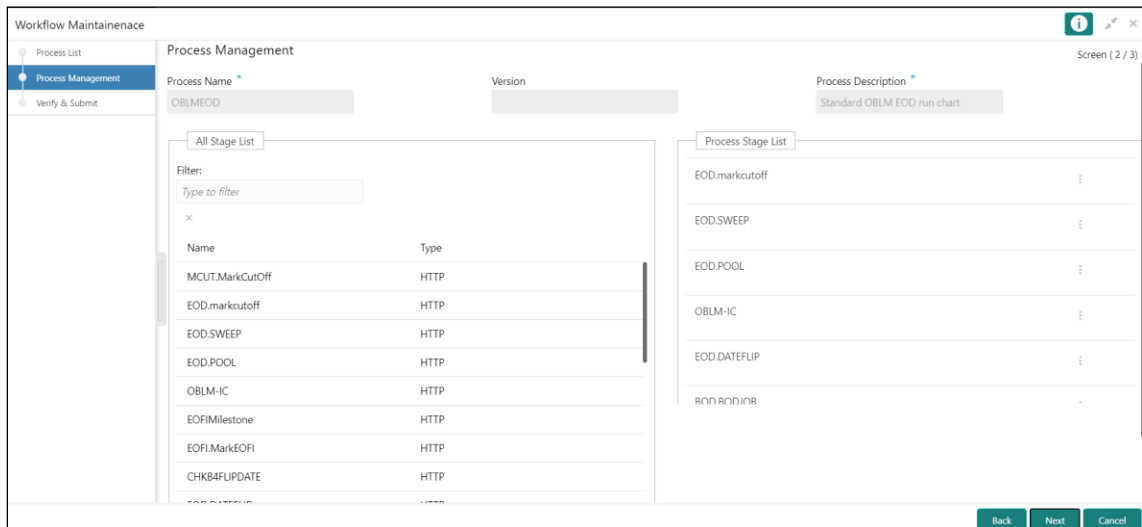
3. Select the **Process Name: blank** checkbox. Click on **Upload DSL+** button and choose file **OBLMEod.json** from local folder.

Figure 2: Process List – Upload DSL



4. Click **Next** button
→ The **Product Management** screen is displayed.

Figure 3: Process Management



5. Click **Next** button and Click **Review** or **Create Process** in **Verify & Submit** screen to register the batch.

2.3 Configure EOD

1. On **Core Maintenance** menu, under **Branch EOD**, click **Configure EOD** to configure batch for a branch. Refer **Section 2.5** in **Oracle Banking Common Core User Guide**.

→ The **Configure EOD** screen is displayed.

Figure 4: Configure EOD

The screenshot shows the 'Configure EOD' web interface. It features a header with the title 'Configure EOD'. Below the header, there are three input fields: 'Branch Code' with the value 'ME1', 'Description' with the value 'ME1', and 'Workflow Name' with the value 'OBLMEOD'. The 'Workflow Name' field is highlighted with a red rectangular box. At the bottom right of the form, there are two buttons: 'Save' and 'Cancel'.

2. Select branch code to configure the batch.

Note: The value specified in Workflow name field must be same as the workflow **name** attribute specified in 3rd line of batch script **OBLMEod.json** file

The screenshot shows a Notepad window titled 'OBLMEod.json - Notepad'. The window contains the following JSON content:

```
{
  "createTime": 1594656285069,
  "name": "OBLMEOD",
  "description": "Standard OBLM EOD run chart",
  "tasks": [
    {
      "type": "HTTP",
      "name": "EOD.markcutoff",
      "taskReferenceName": "EOD.markcutoff",
      "inputParameters": {
        "http_request": {
          "connectionTimeout": "0",
          "readTimeout": "0",
          "vipAddress": "OBLM-BATCH-SERVICES",
          "uri": "/oblm-batch-services/jobscheduler/markEodCutoff/${workflow.input.branchCode}",
          "method": "GET",
          "headers": {
            "appId": "LMB",
            "branchCode": "${workflow.input.branchCode}",
            "userId": "${workflow.input.userId}",
            "eocType": "EOD"
          }
        }
      }
    }
  ],
  "startDelay": 0,
  "optional": false,
  "asyncComplete": false
},
{
  "type": "HTTP",
```

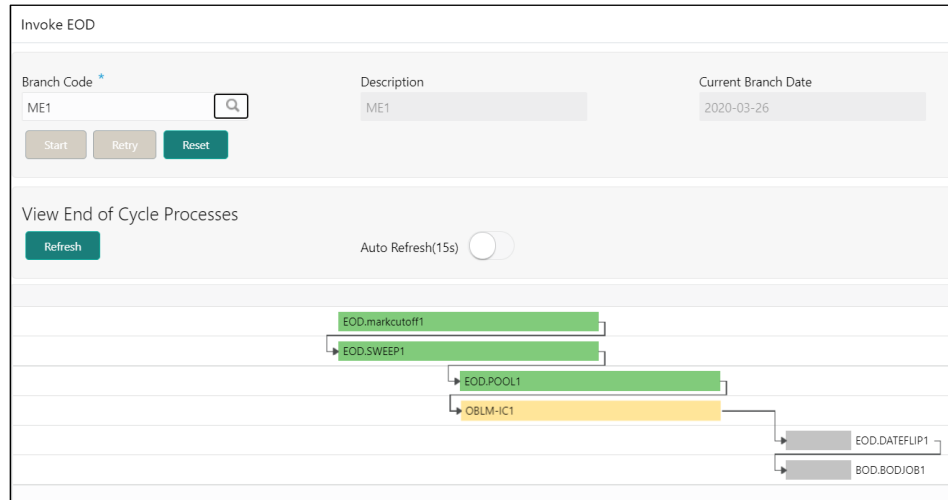
The 'name' attribute in the first task is highlighted with a red box.

2.4 Steps to run EOD for a branch

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

→ **Invoke EOD** screen is displayed.

Figure 5: Invoke EOD



2. Select the branch to run EOD.
3. Click **Refresh** to view the current status of branch.

3 Job definition Naming Convention

Naming convention 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 Microservice 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": {
        "appld": "${workflow.input.appld}",
        "branchCode": "${workflow.input.branchCode}",
        "userId": "${workflow.input.userId}"
      }
    },
    "body": {
      "jobName": "<JobName>",
      "jobParameters": [
        {
          "key": "appld",
          "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 Oracle Banking Microservice Architecture Batch service and the Batch API is implemented as a synchronous call, 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":"<Microservice name registered in eureka>",
      "uri":"<relative URL>",
      "method":"<HTTP Method>",
      "headers":{
        "appld":"${workflow.input.appld}",
        "branchCode":"${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. Please 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":{
                "appld":"${workflow.input.appld}",
                "branchCode":"${workflow.input.branchCode}",
                "userId":"${workflow.input.userId}"
            }
        }
    }
}

```

```

    }
  },
  "asyncComplete":true
},
"startDelay":0,
"optional":false,
"asyncComplete":true
}

```

The following API should be used as a call back to update the status of a task.

Method	POST	
Url	http://<hostname>:<port>/plato-orch-service/api/tasks	
Headers	userId : <Logged in user id> branchCode : <Logged in branch code> appld : platoorch Content-Type : application/json Accept : application/json	userId – User who updates the task branchCode – Branch where the update is performed
Body	<pre> { "workflowInstancelId": "<EOD_Workflow_ID", "taskId": "<Task_ID>", "status": "<Status>" } </pre>	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.

4 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	CHKPENDING MAINT	Job will check pending maintenances that required approval.	BranchCode
3	EOD	SWEEP	Job will execute sweep process scheduled to run during EOD	BranchCode
4	EOD	POOL	Job will execute all pool structures.	BranchCode
5	EOD	PREIC	Job will execute the tasks that are required to run before starting Interest batch	BranchCode
6	EOD	IC.MARKCUT OFF	Job will check for pending tasks and any existing running process before starting Interest batch	BranchCode
7	EOD	OBLM-IC	Job will execute Interest batch	BranchCode
8	MS-EOF1	MS-EOF1	Milestone for the end of financial input	BranchCode
9	EOF1	MARKEOF1	Job will mark the end of financial input	BranchCode
10	MS-CHKB4FLI PDATE	MS-CHKB4FLIPDATE	Milestone for date flip	BranchCode

S No	EOD stage	Job Name	Description	Input Parameters
9	EOD	CMC.DATEFL IP	Job will change system date to next working date in common core	BranchCode
10	EOD	OBLM.DATEF LIP	Job will change system date to next working date in Oracle Banking Liquidity Management	BranchCode
11	EOD	RCUT.RELEA SECUTOFF	Job will mark release cutoff after EOD.	BranchCode
12	EOD	IC.RELEASE CUTOFF	Job will mark release cutoff for IC Batch.	BranchCode
13	BOD	BOD.REALLO C	Job will execute reallocation.	BranchCode
14	BOD	BOD.SWEEP	Job will execute the Reverse sweep and BOD sweeps in sequence	BranchCode
15	BOD	TI.MARKTI	Job will mark the transaction inputs	BranchCode
16	BOD	BOD.ICL	Job will execute the ICL	BranchCode

5 Intraday Jobs

5.1 Create 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 6: Create Task

2. Specify the values mentioned in the below table.

S N o	Task Name	Task Definition
1	OBLM_intraDayAccountPairSweepJob_INT_001	appld::LMS;microServiceName::oblm-sweep-services;contextRoot::oblm-sweep-services;type::schedule;jobName::intraDayAccountPairSweepJob;cronExpression::0 0/5 * * * ?;
2	OBLM_intraDayStructureSweepJob_INT_002	appld::LMS;microServiceName::oblm-sweep-services;contextRoot::oblm-sweep-services;type::schedule;jobName::intraDayStructureSweepJob;cronExpression::0 0/5 * * * ?;
3	OBLM_processMTHoldMessagesJob_INT_003	appld::LMG;microServiceName::oblm-messaging-services;contextRoot::oblm-messaging-services;type::schedule;jobName::processMTHoldMessagesJob;cronExpression::0 0/5 * * * ?;

4	OBLM_pendingPaymentsJob_I NT_004	appId::LMX;microServiceName::oblm-integration- services;contextRoot::oblm-integration- services;type::schedule;jobName:: pendingPaymentsJob;cronExpression::0 0/5 * * * ?;
---	-------------------------------------	---

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

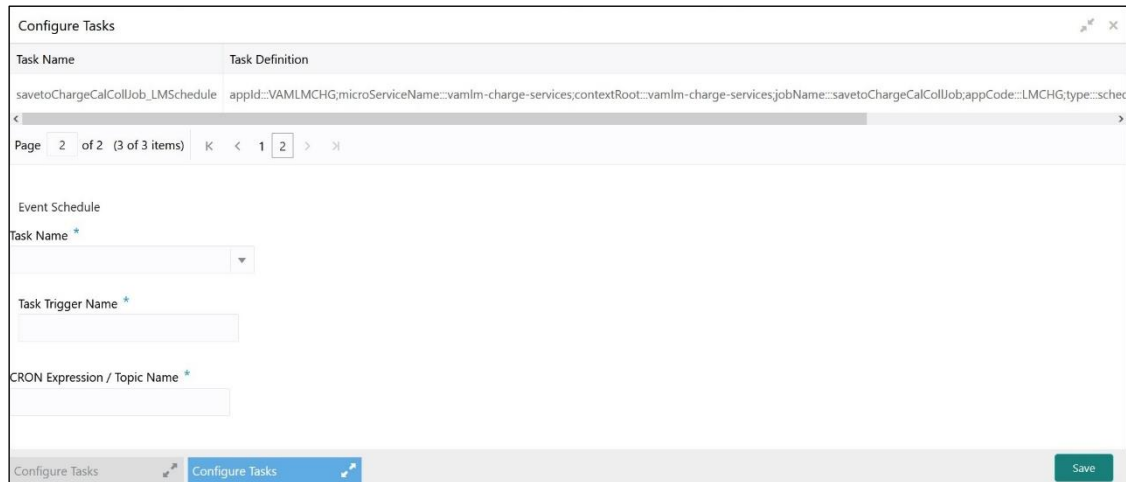
5.2 Configure Tasks

For the above created tasks, the scheduler needs to be configured as shown below. The Configured intra-day jobs will get triggered as per the specified Cron Expression.

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

→ The **Configure Tasks** screen displays.

Figure 7: Configure Tasks



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.

6 Functional Activity Codes

The Functional Activity code and its purpose is described below:

Functional Activity Code	Purpose
LMS_FA_SWEEPDATA_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.
LMS_FA_SWEEPDATA_CREATE	This functional activity code is used to create the sweep data during structure creation
LMS_FA_SWEEPDATA_UPDATE	This functional activity code is used to update existing sweep data during structure modification.
LMX_FA_PENDING_AUTH_VIEW	This functional activity code is used to view the maintenances pending for authorization.
LMX_FA_HAS_PENDING_AUTH	This functional activity code is used to check whether the branch has any pending maintenances for authorization.