

EOD Configuration User Guide

Oracle Banking Virtual Account Management

Release 14.4.0.3.0

Part Number F39510-01

February 2021



EOD Configuration User 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, 2021, 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

Welcome to Configuration User Guide	4
Pre-requisites	4
EOD Configuration	4
EOD Workflow Registration	4
EOD to Branch Mapping	5
Run EOD	8
Annexure	8
Index	81
Reference and Feedback	82
References	82
Documentation Accessibility	82
Feedback and Support	82

Welcome to Configuration User Guide

Oracle Banking Virtual Account Management allows you to execute several functions everyday 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.

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 OBMA 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.

This section contains the following topics:

Pre-requisites	EOD Configuration
--------------------------------	-----------------------------------

Pre-requisites

The following are the pre-requisites for the EOD configuration:

1. OBMA Orchestrator needs to be deployed.
2. OBMA Batch needs to be deployed.



Note

For more details, refer to Annexure - 1 installation guide and Plato Infrastructure Services Installation Guide.

EOD Configuration

EOD Workflow Registration

The factory shipped EOD Workflow should be registered to OBMA Orchestrator. This workflow will contain a sequence of tasks that will be triggered automatically as we invoke EOD. The below API should be used to register EOD Workflow.

Http Method	POST	
URL	http://<hostname>:<port>/plato-orch-service/api/metadata/workflow	
Headers	Content-Type: application/json userId: <user name> branchCode: <branch code> Accept: application/json appld: platoorch	userId – Name of the user who creates the workflow (Optional header parameter) branchCode – Logged in branch code (Optional header parameter)
Body	Please refer the section EOD Flow Definition in Annexure .	

The following job parameters are mandatory to trigger the job using Plato Batch, and should not be modified.

Job Parameter	Description
appld	Application ID of a microservice
microServiceName	Microservice name where the job exists
contextRoot	Context root of the microservice

Plato Batch triggers the job in an asynchronous manner. To update the actual status of a job, there is a call back made by Plato Batch. To use this call back feature following job parameters are mandatory.

Job Parameter	Description
workflowId	EOD Workflow instance ID that gets generated when EOD is invoked successfully
referenceTaskName	Unique name given to each task. The value of this job parameter should be same as the value of <i>taskReferenceName</i> in a task.
userId	Logged in user who triggers the job
branchCode	Logged in branch where the job is triggered
isCallback	Flag to indicate if call back is needed Y – Call back is required N – Call back is not required
callbackType	Value of callbackType should be PLATOORCH as it makes a call back to Plato Orchestrator to update the status of a task.

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

1. CMC_FA_BRANCH_EOD_PROCESS
2. PLATO_FA_JOB_LAUNCH
3. PLATO_FA_JOB_CALLBACK

EOD to Branch Mapping

Once the above workflow gets registered to OBMA Orchestrator successfully, the same should be configured for a branch where EOD has to run. The below API should be used to save EOD configuration for a branch. Ensure that the KeyId (Unique key for a configuration) in the response and use the same for approval.

Http Method	POST	
URL	http://<hostname>:<port>/cmc-branch-services/brancheodmap	

Headers	Content-Type: application/json userId: <user> branchCode: <branch> Accept: application/json appld: CMNCORE	userId – Name of the user who creates the workflow (Mandatory) branchCode – Logged in branch code (Mandatory)
Body	{ "branchCode": <branchcode>, "eodFlowName": <eodwfname> }	Branchcode – Branch code for which EOD has to run eodwfname – Name of the EOD workflow

Authorize EOD configuration for a branch using following API:

Http Method	PATCH	
URL	http://<hostname>:<port>/cmc-branch-services/brancheodmap/<Keyld>/approve	
Headers	Content-Type: application/json userId: <user> branchCode: <branch> Accept: application/json appld: CMNCORE	userId – Name of the user who creates the workflow (Mandatory) branchCode – Logged in branch code (Mandatory)
Body	{ "modNos":[1], "remarks":"approve" }	

Modify EOD configuration for a branch using following API:

Http Method	PUT	
URL	http://<hostname>:<port>/cmc-branch-services/brancheodmap/<Keyld>	
Headers	Content-Type: application/json userId: <user> branchCode: <branch> Accept: application/json appld: CMNCORE	userId – Name of the user who creates the workflow (Mandatory) branchCode – Logged in branch code (Mandatory)

Body	<pre>{ "modNo":<Modification_No>, "branchCode": <branchcode>, "eodFlowName": <eodwfname> }</pre>	<p>Modification_no – Latest modification number of the maintenance</p> <p>Branchcode – Branch code for which EOD Flow has to be modified</p> <p>eodwfname – Name of the EOD workflow</p>
------	--	--

Delete EOD configuration for a branch using following API:

Http Method	DELETE	
URL	http://<hostname>:<port>/cmc-branch-services/brancheodmap/<Keyld>	
Headers	<p>Content-Type: application/json</p> <p>userId: <user></p> <p>branchCode: <branch></p> <p>Accept: application/json</p> <p>appld: CMNCORE</p>	<p>userId – Name of the user who creates the workflow (Mandatory)</p> <p>branchCode – Logged in branch code (Mandatory)</p>
Body	<pre>{ "modNos": [<Modification_no>], "remarks": "<remarks>" }</pre>	<p>Modification_no – Modification number that has to be deleted</p> <p>remarks – Any string</p>

Get EOD configuration for a branch using following API:

Http Method	GET	
URL	http://<hostname>:<port>/cmc-branch-services/brancheodmap/<Keyld>	
Headers	<p>Content-Type: application/json</p> <p>userId: <user></p> <p>branchCode: <branch></p> <p>Accept: application/json</p> <p>appld: CMNCORE</p>	<p>userId – Name of the user who creates the workflow (Mandatory)</p> <p>branchCode – Logged in branch code (Mandatory)</p>
Query Params	<p>branchCode</p> <p>eodFlowName</p>	



For Step 1 and 2, UI will be provided in future releases.

Run EOD

This section helps in invoking EOD for a branch and monitoring the same.

1. Launch **Invoke EOD** screen
2. Select a branch for which EOD has to run from.
3. Select the branch code from drop-down list.
4. Click **Start**.

Now, EOD will get triggered and status will be displayed under **View End of Cycle Processes**. Click **Refresh** to view the latest status.

Annexure

EOD Flow Definition

```
{
  "createTime": 1594656285069,
  "name": "endofdaywf",
  "description": "End of Day Workflow",
  "version": 1,
  "tasks": [
    {
      "type": "HTTP",
      "name": "MCUT.PendingCheck",
      "taskReferenceName": "MCUT.PendingCheck",
      "inputParameters": {
        "http_request": {
          "connectionTimeout": "0",
          "readTimeout": "0",
          "vipAddress": "OBVAM-STATEMENT-SERVICES",
          "uri": "/obvam-statement-services/pendingcheck/${workflow.input.branchCode}",
          "method": "GET",
          "headers": {
```



```

"appld": "${workflow.input.appld}",
"branchCode": "${workflow.input.branchCode}",
"userId": "${workflow.input.userId}"
}
},
"asyncComplete": false
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
"type": "HTTP",
"name": "MCUT.MarkCutOff",
"taskReferenceName": "MCUT.MarkCutOff",
"inputParameters": {
"http_request": {
"connectionTimeOut": "0",
"readTimeOut": "0",
"vipAddress": "PLATO-BATCH-SERVER",
"uri": "/plato-batch-server/jobLauncher/launch/",
"method": "POST",
"headers": {
"appld": "${workflow.input.appld}",
"branchCode": "${workflow.input.branchCode}",
"userId": "${workflow.input.userId}"
},
"body": {
"jobName": "markCutOffJob",
"jobParameters": [

```

```
{
  "key": "appld",
  "value": "CMNCORE"
},
{
  "key": "microServiceName",
  "value": "cmc-batch-services"
},
{
  "key": "contextRoot",
  "value": "cmc-batch-services"
},
{
  "key": "workflowId",
  "value": "${workflow.workflowId}"
},
{
  "key": "referenceTaskName",
  "value": "MCUT.MarkCutOff"
},
{
  "key": "userId",
  "value": "${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
},
{
"type": "HTTP",
"name": "MCUT.pauseVDTurnOver",
"taskReferenceName": "MCUT.pauseVDTurnOver",
"inputParameters": {
"http_request": {
"connectionTimeOut": "0",
"readTimeOut": "0",
"vipAddress": "PLATO-BATCH-SERVER",
"uri": "/plato-batch-server/jobTriggerRegistry/
triggers?jobDefinitions=valueDateUpdateJob,turnOverBalanceUpdateJob",
"method": "DELETE",
"headers": {
"appId": "PLATOBATCH",

```

```

        "branchCode": "${workflow.input.branchCode}",
        "userId": "${workflow.input.userId}"
    }
},
    "asyncComplete": false
},
    "startDelay": 0,
    "optional": false,
    "asyncComplete": false
},
{
    "type": "HTTP",
    "name": "MCUT.VDBalanceUpdate",
    "taskReferenceName": "MCUT.VDBalanceUpdate",
    "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": "valueDateUpdateJob",
            "jobParameters": [

```

```
{
  "key": "appld",
  "value": "VAM"
},
{
  "key": "microServiceName",
  "value": "obvam-account-services"
},
{
  "key": "contextRoot",
  "value": "obvam-account-services"
},
{
  "key": "workflowId",
  "value": "${workflow.workflowId}"
},
{
  "key": "referenceTaskName",
  "value": "MCUT.VDBalanceUpdate"
},
{
  "key": "userId",
  "value": "${workflow.input.userId}"
},
{
  "key": "branchCode",

  "value": "${workflow.input.branchCode}"
},
{
```

```

"key": "isCallback",
"value": "Y"
},
{
"key": "callbackType",
"value": "PLATOORCH"
},
{
"key": "vdBatchCount",
"value": "50"
},
{
"key": "eodDate",
"value": "${workflow.input.eodDate}"
}
]
}
},
"asyncComplete": true
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
"type": "HTTP",
"name": "MCUT.turnOverBalanceUpdate",
"taskReferenceName": "MCUT.turnOverBalanceUpdate",
"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": "${workflow.input.userId}"
},
"body": {
"jobName": "turnOverBalanceUpdateJob",
"jobParameters": [
{
"key": "appld",
"value": "VAM"
},
{
"key": "microServiceName",
"value": "obvam-account-services"
},
{
"key": "contextRoot",
"value": "obvam-account-services"
},
{
"key": "workflowId",
"value": "${workflow.workflowId}"
},
{

```

```
"key": "referenceTaskName",
"value": "MCUT.turnOverBalanceUpdate"
},
{
"key": "turnOverBatchCount",
"value": "50"
},
{
"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
},
{
"type": "HTTP",
"name": "MCUT.ICMarkCutoff",
"taskReferenceName": "MCUT.ICMarkCutoff",
"inputParameters": {
"http_request": {
"connectionTimeOut": "0",
"readTimeOut": "0",
"vipAddress": "OBVAM-IC-INTEREST-BATCH-SERVICES",
"uri": "/obvam-ic-interest-batch-services/cutOff/markCutOff",
"method": "POST",
"headers": {
"appld": "VAM",
"branchCode": "${workflow.input.branchCode}",
"userId": "${workflow.input.userId}"
},
"body": {
"branchCode": "${workflow.input.branchCode}"
}
},
"asyncComplete": true
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
"type": "HTTP",

```

```

"name": "MCUT.ICBEOD",
"taskReferenceName": "MCUT.ICBEOD",
"inputParameters": {
"http_request": {
"connectionTimeout": "0",
"readTimeout": "0",
"vipAddress": "OBVAM-IC-INTEREST-BATCH-SERVICES",
"uri": "/obvam-ic-interest-batch-services/icbeod",
"method": "POST",
"headers": {
"appld": "VAM",
"branchCode": "${workflow.input.branchCode}",
"userId": "${workflow.input.userId}"
},
"body": {
"branchCode": "${workflow.input.branchCode}"
}
},
"asyncComplete": true
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
"type": "HTTP",
"name": "EOFIMilestone",
"taskReferenceName": "MS-EOFI",
"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"
}
]
},
"asyncComplete": true
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
"type": "HTTP",
"name": "EOF1.MarkEOF1",
"taskReferenceName": "EOF1.MarkEOF1",

```

```



```

```
},
{
  "key": "referenceTaskName",
  "value": "EOF1.MarkEOF1"
},

{
  "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
```

```

},
{
  "type": "HTTP",
  "name": "EOF1.EodStatement",
  "taskReferenceName": "EOF1.EodStatement",
  "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": "EodStatementJob",
        "jobParameters": [
          {
            "key": "appld",
            "value": "VAS"
          },
          {
            "key": "microServiceName",
            "value": "obvam-statement-services"
          }
        ],
      }
    }
  }
}

```

```
"key": "contextRoot",
"value": "obvam-statement-services"
},
{
"key": "workflowId",
"value": "${workflow.workflowId}"
},

{
"key": "referenceTaskName",
"value": "EOF1.EodStatement"
},
{
"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
},
{
  "name": "EOF1.ResetSequenceWorkflow",
  "taskReferenceName": "EOF1.ResetSequenceWorkflow",
  "inputParameters": {
    "appId": "${workflow.input.appId}",
    "branchCode": "${workflow.input.branchCode}",
    "userID": "${workflow.input.userID}",
    "id": "${workflow.input.id}"
  },
  "type": "SUB_WORKFLOW",
  "startDelay": 0,
  "subWorkflowParam": {
    "name": "ResetSequenceSubWorkflow"
  },
  "optional": false,
  "asyncComplete": false
},
{
  "type": "HTTP",
  "name": "EOF1.ForgetEntity",
  "taskReferenceName": "EOF1.ForgetEntity",
  "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": "${workflow.input.userId}"
},
"body": {
"jobName": "forgetEntitiesJob",
"jobParameters": [
{
"key": "appId",
"value": "VAE"
},
{
"key": "microServiceName",
"value": "obvam-entity-services"
},
{
"key": "contextRoot",
"value": "obvam-entity-services"
},
{
"key": "workflowId",
"value": "${workflow.workflowId}"
},

```

```
{
  "key": "referenceTaskName",
  "value": "EOF1.ForgetEntity"
},
{
  "key": "userId",
  "value": "${workflow.input.userId}"
},
{
  "key": "branchCode",
  "value": "${workflow.input.branchCode}"
},
{
  "key": "isCallback",
  "value": "Y"
},
{
  "key": "callbackType",
  "value": "PLATOORCH"
},
{
  "key": "retentionPeriod",
  "value": "90"
},
{
  "key": "eodDate",

  "value": "${workflow.input.eodDate}"
}
]
```

```

}
},
"asyncComplete": true
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
"type": "HTTP",
"name": "EOF1.ForgetVirAccount",
"taskReferenceName": "EOF1.ForgetVirAccount",
"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": "forgetAccountJob",
"jobParameters": [
{
"key": "appld",

```

```
"value": "VAM"
},
{
"key": "microServiceName",
"value": "obvam-account-services"
},
{
"key": "contextRoot",
"value": "obvam-account-services"
},
{
"key": "workflowId",
"value": "${workflow.workflowId}"
},
{
"key": "referenceTaskName",
"value": "EOF1.ForgetVirAccount"
},
{
"key": "userId",
"value": "${workflow.input.userId}"
},
{
"key": "branchCode",
"value": "${workflow.input.branchCode}"
},
{
"key": "isCallback",

"value": "Y"
```

```
},  
{  
  "key": "callbackType",  
  "value": "PLATOORCH"  
},  
{  
  "key": "retentionPeriod",  
  "value": "90"  
},  
{  
  "key": "eodDate",  
  "value": "${workflow.input.eodDate}"  
}  
]  
}  
},  
"asyncComplete": true  
},  
"startDelay": 0,  
"optional": false,  
"asyncComplete": true  
},  
{  
  "type": "HTTP",  
  "name": "EOFI.ForgetCoreCustomer",  
  "taskReferenceName": "EOFI.ForgetCoreCustomer",  
  "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": "${workflow.input.userId}"
},
"body": {
"jobName": "forgetCoreCustomersJob",
"jobParameters": [
{
"key": "appld",
"value": "CMNCORE"
},
{
"key": "microServiceName",
"value": "cmc-batch-services"
},
{
"key": "contextRoot",
"value": "cmc-batch-services"
},
{
"key": "workflowId",
"value": "${workflow.workflowId}"
},
{
"key": "referenceTaskName",
```

```
"value": "EOF1.ForgetCoreCustomer"
},
{
"key": "userId",
"value": "${workflow.input.userId}"
},
{
"key": "branchCode",
"value": "${workflow.input.branchCode}"
},
{
"key": "isCallback",
"value": "Y"
},
{
"key": "callbackType",
"value": "PLATOORCH"
},
{
"key": "retentionPeriod",
"value": "90"
},
{
"key": "eodDate",
"value": "${workflow.input.eodDate}"
}
]
},
"asyncComplete": true
```

```

},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
"type": "HTTP",
"name": "EOFI.ForgetCoreAccount",
"taskReferenceName": "EOFI.ForgetCoreAccount",
"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": "forgetCoreAccountsJob",
"jobParameters": [
{
"key": "appld",
"value": "CMNCORE"
},
{

```



```
"key": "microServiceName",

"value": "cmc-batch-services"
},
{
"key": "contextRoot",
"value": "cmc-batch-services"
},
{
"key": "workflowId",
"value": "${workflow.workflowId}"
},
{
"key": "referenceTaskName",
"value": "EOF1.ForgetCoreAccount"
},
{
"key": "userId",
"value": "${workflow.input.userId}"
},
{
"key": "branchCode",
"value": "${workflow.input.branchCode}"
},
{
"key": "isCallback",
"value": "Y"
},
{
"key": "callbackType",
```

```

"value": "PLATOORCH"
},

{
"key": "retentionPeriod",
"value": "90"
},
{
"key": "eodDate",
"value": "${workflow.input.eodDate}"
}
]
}
},
"asyncComplete": true
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
"type": "HTTP",
"name": "ChangdateMilestone",
"taskReferenceName": "MS-CHGDATE",
"inputParameters": {
"http_request": {
"connectionTimeOut": "0",
"readTimeOut": "0",
"vipAddress": "CMC-BRANCH-SERVICES",
"uri": "/cmc-branch-services/brancheod/milestone",

```

```

"method": "POST",
"headers": {
"appld": "CMNCORE",
"branchCode": "${workflow.input.branchCode}",
"userId": "${workflow.input.userId}"
},
"body": {
"data": [
{
"workflowId": "${workflow.workflowId}",
"taskId": "${CPEWF_TASK_ID}",
"waitTime": "5000"
}
]
},
"asyncComplete": true
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
"type": "HTTP",
"name": "BOD.ChangeDate",
"taskReferenceName": "BOD.ChangeDate",
"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": "${workflow.input.userId}"
},
"body": {
"jobName": "changeDateJob",
"jobParameters": [
{
"key": "appId",
"value": "CMNCORE"
},
{
"key": "microServiceName",
"value": "cmc-batch-services"
},
{
"key": "contextRoot",
"value": "cmc-batch-services"
},
{
"key": "workflowId",
"value": "${workflow.workflowId}"
},
{
"key": "referenceTaskName",

```

```
"value": "BOD.ChangeDate"
},
{
"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
},
{
"type": "HTTP",
```

```

"name": "BOD.ICFlipDate",
"taskReferenceName": "BOD.ICFlipDate",
"inputParameters": {
"http_request": {
"connectionTimeout": "0",
"readTimeout": "0",
"vipAddress": "CMC-BRANCH-SERVICES",
"uri": "/cmc-branch-services/ batch/changeicdate/${workflow.input.branchCode}",
"method": "POST",
"headers": {
"appld": "VAM",
"branchCode": "${workflow.input.branchCode}",
"userId": "${workflow.input.userId}"
}
},
"asyncComplete": true
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
"type": "HTTP",
"name": "RCUT.ReleaseCutOff",
"taskReferenceName": "MCUT.ReleaseCutOff",
"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": "${workflow.input.userId}"
},
"body": {
"jobName": "releaseCutOffJob",
"jobParameters": [
{
"key": "appld",
"value": "CMNCORE"
},
{
"key": "microServiceName",
"value": "cmc-batch-services"
},
{
"key": "contextRoot",
"value": "cmc-batch-services"
},
{
"key": "workflowId",
"value": "${workflow.workflowId}"
},
{
"key": "referenceTaskName",
"value": "MCUT.ReleaseCutOff"
},
}
}

```

```

{
  "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
},
{
  "type": "HTTP",
  "name": "RCUT.ICReleaseCutoff",
  "taskReferenceName": "RCUT.ICReleaseCutoff",
  "inputParameters": {

```



```

"http_request": {
  "connectionTimeout": "0",
  "readTimeout": "0",
  "vipAddress": "OBVAM-IC-INTEREST-BATCH-SERVICES",
  "uri": "/obvam-ic-interest-batch-services/cutOff/releaseCutOff",
  "method": "POST",
  "headers": {
    "appld": "VAM",
    "branchCode": "${workflow.input.branchCode}",
    "userId": "${workflow.input.userId}"
  },
  "body": {
    "branchCode": "${workflow.input.branchCode}"
  },
  "asyncComplete": true
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
  "type": "HTTP",
  "name": "RCUT.resumeVDTurnOver",
  "taskReferenceName": "RCUT.resumeVDTurnOver",
  "inputParameters": {
    "http_request": {
      "connectionTimeout": "0",
      "readTimeout": "0",
      "vipAddress": "PLATO-BATCH-SERVER",

```

```

        "uri": "/plato-batch-server/jobTriggerRegistry/
triggers?jobDefinitions=valueDateUpdateJob,turnOverBalanceUpdateJob",
        "method": "PUT",
        "headers": {
            "appld": "PLATOBATCH",
            "branchCode": "${workflow.input.branchCode}",
            "userId": "${workflow.input.userId}"
        }
    },
    "asyncComplete": false
},
"startDelay": 0,
"optional": false,
"asyncComplete": false
},
{
    "type": "HTTP",
    "name": "RCUT.UntankBalance",
    "taskReferenceName": "RCUT.UntankBalance",
    "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": "untankingBalanceJob",
"jobParameters": [
{
"key": "appld",
"value": "VAM"
},
{
"key": "microServiceName",
"value": "obvam-account-services"
},
{
"key": "contextRoot",
"value": "obvam-account-services"
},
{
"key": "workflowId",
"value": "${workflow.workflowId}"
},
{
"key": "referenceTaskName",
"value": "RCUT.UntankBalance"
},
{
"key": "userId",
"value": "${workflow.input.userId}"
},
{
```

```
"key": "branchCode",
"value": "${workflow.input.branchCode}"
},
{
"key": "isCallback",
"value": "Y"
},
{
"key": "callbackType",
"value": "PLATOORCH"
},
{
"key": "eodDate",
"value": "${workflow.input.eodDate}"
},
{
"key": "tankedTxnCount",
"value": "50"
}
]
},
"asyncComplete": true
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
"type": "HTTP",
```

```

"name": "RCUT.MarkAccountInactive",
"taskReferenceName": "RCUT.MarkAccountInactive",
"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": "markAccountInactiveJob",
"jobParameters": [
{
"key": "appld",
"value": "VAM"
},
{
"key": "microServiceName",
"value": "obvam-account-services"
},
{
"key": "contextRoot",
"value": "obvam-account-services"
},
{

```

```
"key": "workflowId",
"value": "${workflow.workflowId}"
},
{
"key": "referenceTaskName",
"value": "RCUT.MarkAccountInactive"
},
{
"key": "userId",
"value": "${workflow.input.userId}"
},
{
"key": "branchCode",
"value": "${workflow.input.branchCode}"
},
{
"key": "isCallback",
"value": "Y"
},
{
"key": "callbackType",
"value": "PLATOORCH"
},
{
"key": "eodDate",
"value": "${workflow.input.eodDate}"
}
]
}
},
```

```

"asyncComplete": true
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
"type": "HTTP",
"name": "RCUT.AmountBlockExpiry",
"taskReferenceName": "RCUT.AmountBlockExpiry",
"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": "amountBlockExpiryJob",
"jobParameters": [
{
"key": "appld",
"value": "VAB"
},
{

```

```
"key": "microServiceName",
"value": "obvam-eca-services"
},
{
"key": "contextRoot",
"value": "obvam-eca-services"
},
{
"key": "workflowId",
"value": "${workflow.workflowId}"
},
{
"key": "referenceTaskName",
"value": "RCUT.AmountBlockExpiry"
},
{
"key": "userId",
"value": "${workflow.input.userId}"
},
{
"key": "branchCode",
"value": "${workflow.input.branchCode}"
},
{
"key": "isCallback",
"value": "Y"
},
{
"key": "callbackType",
```



```

"value": "PLATOORCH"

},

{
"key": "eodDate",
"value": "${workflow.input.eodDate}"
}
]
}
},

"asyncComplete": true
},

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

{
"type": "HTTP",
"name": "RCUT.CreditlimitUtil",
"taskReferenceName": "RCUT.CreditlimitUtil",
"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": "creditLimitUtilRevalJob",
"jobParameters": [
{
"key": "appld",
"value": "VAM"
},
{
"key": "microServiceName",
"value": "obvam-account-services"
},
{
"key": "contextRoot",
"value": "obvam-account-services"
},
{
"key": "workflowId",
"value": "${workflow.workflowId}"
},
{
"key": "referenceTaskName",
"value": "RCUT.CreditlimitUtil"
},
{
"key": "userId",
"value": "${workflow.input.userId}"
},
},
```

```
{
  "key": "branchCode",
  "value": "${workflow.input.branchCode}"
},
{
  "key": "isCallback",
  "value": "Y"
},
{
  "key": "callbackType",
  "value": "PLATOORCH"
},
{
  "key": "commitFrequency",
  "value": "50"
}
]
},
"asyncComplete": true
},
"startDelay": 0,
"optional": false,
"asyncComplete": true
},
{
  "type": "HTTP",
  "name": "TransactionInputMilestone",
  "taskReferenceName": "MS-TI",
```

```



```

```

"type": "HTTP",
"name": "TI.MarkTI",
"taskReferenceName": "TI.MarkTI",
"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": "markTIJob",
"jobParameters": [
{
"key": "appld",
"value": "CMNCORE"
},
{
"key": "microServiceName",
"value": "cmc-batch-services"
},
{
"key": "contextRoot",
"value": "cmc-batch-services"
}
}
}
}

```

```
},
{
  "key": "workflowId",
  "value": "${workflow.workflowId}"
},
{
  "key": "referenceTaskName",
  "value": "TI.MarkTI"
},
{
  "key": "userId",
  "value": "${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
}
],
"outputParameters": {},

"failureWorkflow": "string",
"schemaVersion": 2,
"restartable": true,
"workflowStatusListenerEnabled": true
}

```

Reset Sequence Sub Workflow



Note

Reset Sequence Sub Workflow to be added, if “taskReferenceName” is mentioned as “ResetSequenceSubWorkflow” in EoD Flow Definition.

```

{
  "createTime": 1594656285069,
  "name": "ResetSequenceSubWorkflow",
  "description": "SubWorkflow for Reset Sequence Number",
  "version": 1,
  "tasks": [
    {
      "name": "Transaction journal",
      "taskReferenceName": "VAT",
      "inputParameters": {
        "http_request": {
          "connectionTimeout": "1172000000",
          "readTimeout": "172000000",

```

```

        "vipAddress": "OBVAM-TRANSACTION-JOURNAL-SERVICES",
        "uri": "/obvam-transaction-journal-services/txns/resetSequence/
        ${workflow.input.branchCode}",
        "method": "PATCH",
        "headers": {
            "appld": "VAT",
            "branchCode": "${workflow.input.branchCode}",
            "userId": "${workflow.input.userId}"
        }
    },
    "asyncComplete": false
},
"type": "HTTP",
"startDelay": 0,
"optional": false,
"asyncComplete": false
},
{
    "name": "Internal Transfer",
    "taskReferenceName": "VAN",
    "inputParameters": {
        "http_request": {
            "connectionTimeOut": "1172000000",
            "readTimeOut": "172000000",
            "vipAddress": "OBVAM-INTERNAL-TRANSFER-SERVICES",
            "uri": "/obvam-internal-transfer-services/intbks/resetSequence/${workflow.input.branchCode}",
            "method": "PATCH",
            "headers": {
                "appld": "VAN",
                "branchCode": "${workflow.input.branchCode}",

```



```

        "userId": "${workflow.input.userId}"
    }
},
    "asyncComplete": false
},
    "type": "HTTP",
    "startDelay": 0,
    "optional": false,
    "asyncComplete": false
},
{
    "name": "Statement",
    "taskReferenceName": "VAS",
    "inputParameters": {
        "http_request": {
            "connectionTimeOut": "1172000000",
            "readTimeOut": "172000000",
            "vipAddress": "OBVAM-STATEMENT-SERVICES",
            "uri": "/obvam-statement-services/stmnts/resetSequence/${workflow.input.branchCode}",
            "method": "PATCH",
            "headers": {
                "appld": "VAS",
                "branchCode": "${workflow.input.branchCode}",
                "userId": "${workflow.input.userId}"
            }
        },
        "asyncComplete": false
    },
    "type": "HTTP",
    "startDelay": 0,

```

```

    "optional": false,
    "asyncComplete": false
  },
  {
    "name": "Amount Block",
    "taskReferenceName": "VAB",
    "inputParameters": {
      "http_request": {
        "connectionTimeOut": "1172000000",
        "readTimeOut": "172000000",
        "vipAddress": "OBVAM-ECA-SERVICES",
        "uri": "/obvam-eca-services/eca/resetSequence/${workflow.input.branchCode}",
        "method": "PATCH",
        "headers": {
          "appld": "VAB",
          "branchCode": "${workflow.input.branchCode}",
          "userId": "${workflow.input.userId}"
        }
      },
      "asyncComplete": false
    },
    "type": "HTTP",
    "startDelay": 0,
    "optional": false,
    "asyncComplete": false
  }
],
"outputParameters": {
  "Output": "${VAB.output}"
},

```

```

"failureWorkflow": "string",

"schemaVersion": 2,

"restartable": true,

"workflowStatusListenerEnabled": true

}

```

EOD Milestones

The factory shipped EOD process consists of following stages:

1. Mark Cutoff (**MCUT**)
2. Mark End of Financial Input (**EOFI**)
3. Beginning of Day (**BOD**)
4. Release Cutoff (**RCUT**)
5. Transaction Input (**TI**)

S.NO	Milestone/ Stage	Milestone Code	Job Name	Description
1	Mark Cutoff	MCUT	markCutOffJob	Mark Cutoff is the first stage in EOD operations. All the accounting entries that is received as part of transaction entry will be processed the next day.
2	Mark End of Financial Input	EOFI	markEOFIJob	<p>At this stage no further accounting entries can be passed for the day either through transactions that you have entered or by transactions that are automatically triggered by the system.</p> <p>All the transactions that are entered will be tanked and processed after cutoff is released.</p> <p>After you moved the system to this status, you can generate financial reports for the day. Since all the automatic processes have been run for the day, and since the balances available will be the latest.</p> <p>After EOFI is marked, you can generate all advice related messages like reports, statements and all other information on the day's activities.</p>
3	Beginning of Day	BOD	changeDateJob	This is the stage after system date has been changed to next working day.

4	Release Cutoff	RCUT	releaseCutOffJob	At this stage cutoff for the branch will be released i.e. branch will be made available to allow accounting entries for the transactions that you enter. Also, all the transactions that are tanked between mark cutoff and release cutoff will be untanked for processing.
5	Transaction Input	TI	markTIJob	This stage moves the branch to transaction input stage.This stage moves the branch to transaction input stage.

API Details:

Method	POST	
URL	http://<hostname>:<port>/plato-batch-server/jobLauncher/launch/	
Headers	userId : <Logged in user id> branchCode : <Logged in branch code> appld : PLATOBATCH	User should have appropriate functional activity code mapped to their role for the logged-in branch to access this endpoint.
Body	<pre>{ "jobName": "<Job Name>", "jobParameters": [{ "key": "appld", "value": <Application ID> }, { "key": "microServiceName", "value": <Microservice Name> }, { "key": "contextRoot", "value": <Context Root> }] }</pre>	

Custom Jobs

Naming convention to be followed when a custom job is introduced as a task into EOD process

1. Task name must be prefixed with EOD stage name Ex : EOFI.VAMEodStmt.
2. Milestone must be prefixed with "MS-". Ex : MS-BranchCutOff

How to integrate custom job

1. If the custom job use OBMA 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 does not use OBMA 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":false
}

```



Note

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

- 3. If the custom job does not use OBMA 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
}
```

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>{ "workflowInstanceId": "<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. Common Core Jobs

S. No	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
	EOFI	ForgetCoreCustomer	Job to forget core customer	Content-Type: application/json userId : <Logged in user id> branchCode : <Logged in branch code> appld : PLATOBATCH	POST	<pre> "jobName":"forgetCoreCustomersJob", "jobParameters":[{ "key":"appld", "value":"CMNCORE" }, { "key":"microServiceName", "value":"cmc-batch-services" }, { "key":"contextRoot", "value":"cmc-batch-services" }] </pre>

S. No	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
2	EOFI	ForgetCoreAccount	Job to forget core account	http:// <hostname>< port>/plato- batch-server/ jobLauncher/ launch/ Content-Type: application/ json userId : <Logged in user id>	POST	<pre> { "jobName":"fo rgetCoreAccou ntsJob", "jobParameter s":[{ "key":"appld", "value":"CMNC ORE" }, { "key":"microSe rviceName", "value":"cmc- batch-services" }, { "key":"context Root", "value":"cmc- batch-services" }] } </pre>

5. OBVAM Jobs

SNo	Milest one Code	Job Name	Description	URL and Headers	Http Method	Request Body
1	MCUT	Pending Check	Task to check if any pending maintenance or transaction exist. This pending check task will fail if there is any unauthorized maintenance or transaction. If pending check task fails, you should check for unauthorized maintenance or transaction and take necessary action. This action could be authorizing/ deleting maintenance/ transaction.	<p>http:// <hostname>:<port>//obvam-statement-services/ pendingcheck/ {branchCode}</p> <p>Content-Type: application/ json</p> <p>userId: <Logged in user ID></p> <p>branchCode: <Logged in branch code></p> <p>Accept: application/ json</p> <p>appld: VAS</p>	GET	

SNo	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
2	MCUT	VDBalanceUpdate	Job to calculate value dated balances for a virtual account	http:// <hostname>:< port>/plato- batch-server/ jobLauncher/ launch/ Content-Type: application/ json userId : <Logged in user id> branchCode : <Logged in branch code> appld : PLATOBATCH	POST	<pre> { "jobName": "valueDateUpdateJob", "jobParameters": [{ "key": "appld", "value": "VAM" }, { "key": "microServiceName", "value": "obvam-account-services" }, { "key": "contextRoot", "value": "obvam-account-services" }] } </pre>

SNo	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
3	MCUT	turnOver Balance Update	Job to calculate turnover balance for a virtual account which is used for charge calculations	http:// <hostname>:< port>/plato- batch-server/ jobLauncher/ launch/ Content-Type: application/ json userId : <Logged in user id> branchCode : <Logged in branch code> appld : PLATOBATCH	POST	<pre> { "jobName": "valueDateUpdateJob", "jobParameters": [{ "key": "appld", "value": "VAM" }, { "key": "microServiceName", "value": "obvam-account-services" }, { "key": "contextRoot", "value": "obvam-account-services" }] } </pre>

SNo	Milest one Code	Job Name	Description	URL and Headers	Http Method	Request Body
4	MCUT	ICMark Cutoff	Job to mark cutoff so that interest processing can start	http:// <hostname>:< port>/obvam- ic-interest- batch- services/ cutOff/ markCutOff Content-Type: application/ json userId: <Logged in user ID> branchCode: <Logged in branch code> Accept: application/ json appld: VAM	POST	<pre>{ "branchCode": "<Branch code for which cutoff should be marked" }</pre>
5	MCUT	ICBEOD	Job to process interest calculations	http:// 10.40.162.114 :8604/obvam- ic-interest- batch- services/ icbeod Content-Type: application/ json userId: <Logged in user ID> branchCode: <Logged in branch code> Accept: application/ json appld: VAM	POST	<pre>{ "branchCode": "<Branch where interest processing job has to run" }</pre>

SNo	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
6	EOF1	EodStatement	Job to generate EOD statement	http:// <hostname>:< port>/plato- batch-server/ jobLauncher/ launch/ Content-Type: application/ json userId : <Logged in user id> branchCode : <Logged in branch code>	POST	<pre> { "jobName": "EodStatementJob", "jobParameters": [{ "key": "appld", "value": "VAS" }, { "key": "microServiceName", "value": "obva m-statement- services" }] } </pre>

SNo	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
7	EOFI	ForgetEntity	Job to forget virtual entity	http:// <hostname>:< port>/plato- batch-server/ jobLauncher/ launch/ Content-Type: application/ json userId : <Logged in user id> branchCode : <Logged in branch code> appld : PLATOBATCH	POST	<pre> { "jobName": "forgetEntitiesJob", "jobParameters": [{ "key": "appld", "value": "VAE" }, { "key": "microServiceName", "value": "obvam-entity-services" }, { "key": "contextRoot", "value": "obvam-entity-services" }] } </pre>

SNo	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
8	EOF1	ForgetVirtualAccount	Job to forget virtual account	http:// <hostname>:< port>/plato- batch-server/ jobLauncher/ launch/ Content-Type: application/ json userId : <Logged in user id> branchCode : <Logged in branch code> appld : PLATOBATCH	POST	<pre> { "jobName":"forgetAccountJob", "jobParameters":[{ "key":"appld", "value":"VAM" }, { "key":"microServiceName", "value":"obvam-account-services" }, { "key":"contextRoot", "value":"obvam-account-services" }] } </pre>

SNo	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
9	BOD	ICFlipDate	Job to change branch date	http:// <hostname>:< port>/cmc- branch- services/ batch/ changeicdate/ {branchCodeId} Content-Type: application/ json userId: <Logged in user ID> branchCode: <Logged in branch code> Accept: application/ json appId: CMNCORE	POST	Branch code where change date has to happen

SNo	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
10	RCUT	ICReleaseCutoff	Job to release cutoff after interest processing is done	http:// <hostname>:< port>// obvam-ic- interest- batch- services/ cutOff/ releaseCutOff Content-Type: application/ json userId: <Logged in user ID> branchCode: <Logged in branch code> Accept: application/ json applId: VAM	POST	<pre>{ "branchCode": "<Branch code for which cutoff needs to be released>" }</pre>

SNo	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
11	RCUT	UntankBalance	Job to untank accounting entries	http:// <hostname>:< port>/plato- batch-server/ jobLauncher/ launch/ Content-Type: application/ json userId : <Logged in user id> branchCode : <Logged in branch code> appld : PLATOBATCH	POST	<pre> { "jobName": "untankingBalanceJob", "jobParameters": [{ "key": "appld", "value": "VAM" }, { "key": "microServiceName", "value": "obvamm-account-services" }, { "key": "contextRoot", "value": "obvamm-account-services" }] } </pre>

SNo	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
12	RCUT	MarkAccountInactive	Job to mark virtual accounts inactive	http:// <hostname>:< port>/plato- batch-server/ jobLauncher/ launch/ Content-Type: application/ json userId : <Logged in user id> branchCode : <Logged in branch code> appld : PLATOBATCH	POST	<pre> { "jobName": " markAccountI nactiveJob", "jobParameter s": [{ "key": "appld", "value": "VAM" }, { "key": "microSe rviceName", "value": "obva m-account- services" }, { "key": "context Root", "value": "obva m-account- services" }] } </pre>

SNo	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
13	RCUT	Amount BlockExpiry	Job to mark amount block expired based on expiry date	http:// <hostname>:< port>/plato- batch-server/ jobLauncher/ launch/ Content-Type: application/ json userId : <Logged in user id> branchCode : <Logged in branch code> appld : PLATOBATCH	POST	<pre> { "jobName":"a mountBlockEx piryJob", "jobParameter s":[{ "key":"appld", "value":"VAB" }, { "key":"microSe rviceName", "value":"obva m-eca- services" }, { "key":"context Root", "value":"obva m-eca- services" }] } </pre>

SNo	Milestone Code	Job Name	Description	URL and Headers	Http Method	Request Body
14.	RCUT	CreditlimitUtil	Job to re-valuate credit limit utilization based on updated exchange rates	http://<hostname>:<port>/plato-batch-server/jobLauncher/launch/ Content-Type: application/json userId : <Logged in user id> branchCode : <Logged in branch code> appld : PLATOBATCH	POST	<pre>{ "jobName": "creditLimitUtilRevalJob", "jobParameters": [{ "key": "appld", "value": "VAM" }, { "key": "microServiceName", "value": "obvam-account-services" }, { "key": "contextRoot", "value": "obvam-account-services" }] }</pre>



The following Functional Activity needs to be mapped to user' role to access Pending check API.

VAS_FA_PENDING_CHECK

A

Annexure 8

E

EOD to Branch Mapping 5

EOD Workflow Registration 4

R

Run EOD 7

Reference and Feedback

References

For more information on any related features, you can refer to the following documents:

- Oracle Banking Security Management System User Guide
- Oracle Banking Common Core User Guide
- Oracle Banking Getting Started User Guide
- Oracle Banking Virtual Account Management Overview Guide
- Oracle Banking Virtual Account Management Customer and Accounts User Guide
- Oracle Banking Virtual Account Management Identifier User Guide
- Oracle Banking Virtual Account Management Transactions User Guide
- Oracle Banking Virtual Account Management Installation Guides

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/us/corporate/accessibility/index.html>.

Feedback and Support

Oracle welcomes customers' comments and suggestions on the quality and usefulness of the document. Your feedback is important to us. If you have a query that is not covered in this user guide or if you still need assistance, please contact documentation team.