Oracle® Banking Payments Cloud Service

Common Core - Scheduler User Guide

Release 14.8.1.0.0
G46782-01
October 2025

ORACLE

Oracle Banking Payments Cloud Service Common Core - Scheduler User Guide, Release 14.8.1.0.0

G46782-01

Contents

Preface

This topic contains the following sub-topics:

Purpose

This user manual is designed to help you quickly get acquainted with the many functions routinely executed everyday in Oracle Banking Payments Cloud Service.

To access information specific to a particular field, place the cursor on the relevant field and press **F1** on the keyboard.

Audience

Table 1 Audience

Role	Function
Back office clerk	Input functions for contracts
Back office managers/officers	Authorization functions
Product Managers	Product definition and authorization
End of Day operators	Processing during End of Day/Beginning of Day
Financial Controller/Product Managers	Generation of reports

Acronyms and Abbreviations

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

Table 2 Abbreviation

Abbreviation	Description
POSTEOPD	Post End of Previous Day
MARKEOPD	Mark End of Previous Day
MARKTI	Mark Transaction Input
POSTEOBOD	Post End of Beginning of Day
MARKBOD	Mark Beginning of Day
MARKEOD	Mark End of Day
POSTEOED	Post End of End of Day
MARKEOFI	Mark End of Financial Input
POSTEOFI	Post End of Financial Input
MARKEOTI	Mark End of Transaction Input
POSTEOTI	Post End of Transaction Input
TI	Transaction Input
EOC	End of Cycle
BOD	Beginning of Day
EOD	End of Day
EOPD	End of Previous Day
FI	Financial Input
EOTI	End of Transaction Input

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.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

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.

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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Related Resources

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

- *Getting Started User Guide*
- *Oracle Banking Security Management System User Guide*
- *Oracle Banking Microservices Platform Foundation User Guide*
- *Routing Hub Configuration User Guide*
- *Oracle Banking Common Core User Guide*
- *Interest and Charges User Guide*
- *Oracle Banking Liquidity Management Configuration Guide*
- *Oracle Banking Liquidity Management File Upload User Guide*

Screenshot Disclaimer

The personal information used in the interface or documents is sample data and does not exist in the real world. It is provided for reference purposes only.

Acronyms and Abbreviations

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

Table 3 Acronyms and Abbreviations

Abbreviation	Description
DDA	Demand Deposit Accounts
ECA	External Credit Approval
EOD	End of Day
IBAN	International Bank Account Number

Basic Actions

The basic actions performed in the screens are as follows:

Table 4 Basic Actions

Actions	Description
Approve	Click **Approve** to approve the initiated record. - This button is displayed once the user click **Authorize**.
Audit	Click **Audit** to view the maker details, checker details of the particular record. - This button is displayed only for the records that are already created.
Authorize	Click **Authorize** to authorize the record created. A maker of the screen is not allowed to authorize the same. Only a checker can authorize a record. - This button is displayed only for the already created records. For more information on the process, refer Authorization Process.
Cancel	Click **Cancel** to cancel the action performed.
Close	Click **Close** to close a record. This action is available only when a record is created.
Collapse All	Click **Collapse All** to hide the details in the sections. - This button is displayed once the user click **Compare**.
Compare	Click **Compare** to view the comparison through the field values of old record and the current record. - This button is displayed in the widget once the user click **Authorize**.
Confirm	Click **Confirm** to confirm the action performed.
Expand All	Click **Expand All** to expand and view all the details in the sections. - This button is displayed once the user click **Compare**.
New	Click **New** to add a new record. The system displays a new record to specify the required data. The fields marked with asterisk are mandatory. - This button is displayed only for the records that are already created.
OK	Click **OK** to confirm the details in the screen.
Save	Click **Save** to save the details entered or selected in the screen.
Unlock	Click **Unlock** to update the details of an existing record. The system displays an existing record in editable mode. - This button is displayed only for the records that are already created.
View	Click **View** to view the details in a particular modification stage. - This button is displayed in the widget once the user click **Authorize**.
View Difference only	Click **View Difference only** to view a comparison through the field element values of old record and the current record, which has undergone changes. - This button is displayed once the user click **Compare**.

Symbols & Icons

This guide has the following list of symbols and icons.

Table 5 Symbols and Icons - Common

Icons	Function
Figure 1 Exit 	Exit
Figure 2 Add Row 	Add row
Figure 3 Delete Row	Delete row
Figure 4 Option List 	Option List

1

Job Scheduling

This topic describes the job scheduling process.

Job scheduling is the process where different tasks get executed at a pre-determined time or when the right event happens. A job scheduler is a system that can be integrated with other software systems to execute or notify other software components when a pre-determined, scheduled time arrives. The two types of job schedulers used in Oracle Banking Corporate Lending FCJ architecture are as follows:

- **Quartz** - Provides a scheduler interface to enable operations such as scheduling and unscheduling of jobs and starting, stopping, and pausing the scheduler
- **Flux** - Software component used for performing enterprise job scheduling

This topic contains the following sub-topics:

1.1 Define Jobs

This topic explains systematic instructions to define jobs.

A job is a business activity that the system performs repeatedly on a timely basis. Oracle Banking Corporate Lending enables to define a job and schedule it using the **Job Maintenance** screen.

1. On **Homescreen**, type **STDJOBMT** in the text box, and click **Next**.

   The **Job Maintenance** screen displays.

Figure 1-1 Job Maintenance

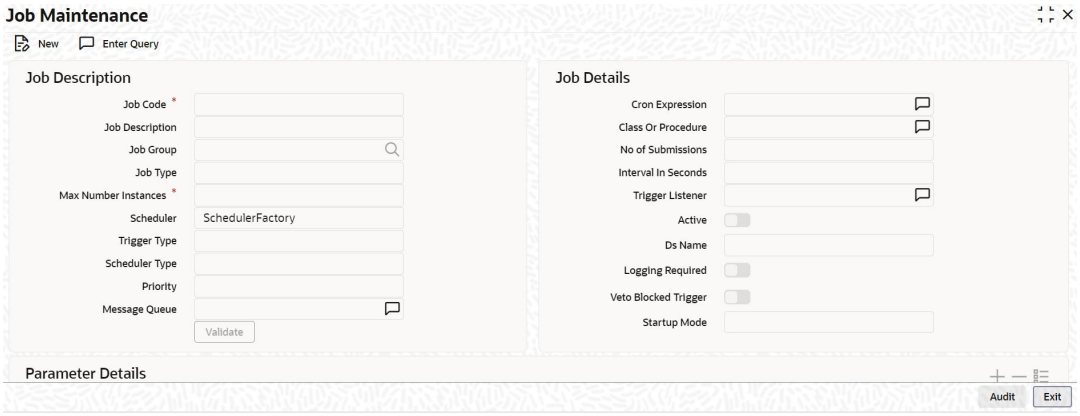
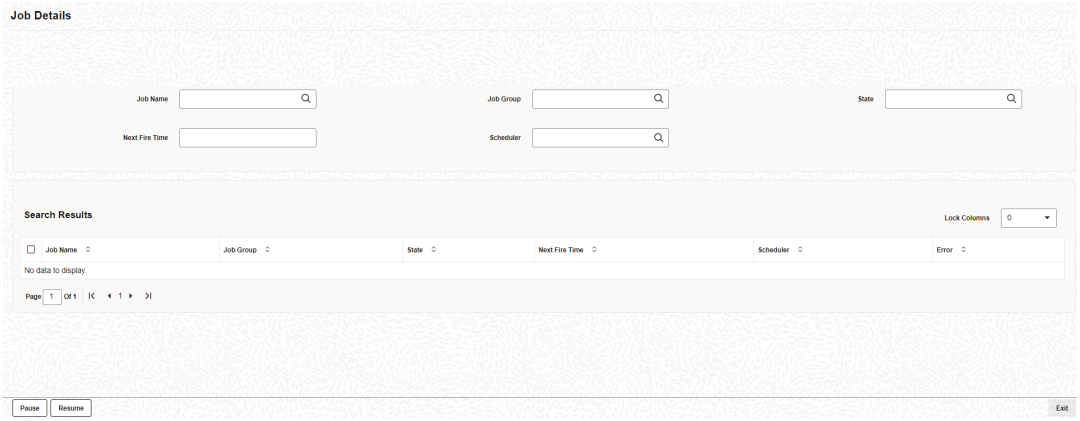


Figure 1-2 Job Details



2. On the **Job Maintenance** screen, specify the fields.

> **Note**
>
> The fields, which are marked with an asterisk, are mandatory.

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

Table 1-1 Job Maintenance - Field Description

Field	Description
Job Code	Specify the unique code to identify the job.
Job Description	Specify a brief description of what the job is supposed to do.
Job Group	Specify the job group name to represent the same group of jobs for identification.
Job Type	Select the type of job from the drop-down list: • **PL/SQL** • **JAVA**

Table 1-1 (Cont.) Job Maintenance - Field Description

Field	Description
Max Number Instances	Specify the maximum number of instances that needs to be queued up. For example, If a job runs for more than the duration defined, the next instance of the same job will be ready for processing. This parameter defines the jobs behavior in such cases. If the job is maintained as **STATEFUL**, then the number of such missed instances will be queued up so that it would start executing once this long-running job ends. This field specifies the number of such job instances that needs to be queued up. If the job is maintained as **STATELESS**, it indicates the number of threads that can be executed in parallel. If the max number of instances is maintained as **0**, no instances are queued or parallel processed till the current running instance is completed.
Scheduler	Specify the name of the scheduler. The system defaults the name to **SchedulerFactory**. However, the scheduler name can be modified. This signifies the scheduler name which is configured as part of infra.
Trigger Type	Select the type of trigger from the drop-down list: • **Simple** - Interval based jobs (that is every hour) • **Cron**: Time based jobs (that is Friday 4:30PM). **Note:** When the scheduler trigger type is set to **CRON**, and the parameter is **TIMEZONE**, then the system triggers the jobs based on the time zone set. If the **TIMEZONE** property is not available, then the server time zone will be used to trigger the scheduler job.
Scheduler Type	Select the type of scheduler from the drop-down list: • **Quartz** • **Flux**
Priority	Select the priority on which the system should execute the jobs in the scheduler from the drop-down list: • **Normal** • **High** • **Low** If two jobs with different priorities fire at the same time, then the system gives preference to the job with higher priority.
Message Queue	Specify the default JMS queue to which a job needs to send the message. Specify this only if the job has to send messages to JMS.
Cron Expression	Specify the corresponding Cron expression for a job with the trigger type as **Cron**. The user needs to do this to determine the time and interval of job firing.
Class or Procedure	Specify the Java class file name if the job type is **Java** or the **PL/SQL** procedure name if the job type is **PL/SQL**. This denotes which java class or PL/SQL procedure the system should call when a job fires.
Number of Submissions	Specify the number of times a job can fire before it is unscheduled from the scheduler. This applies only to trigger types maintained as **Simple**.
Interval In Seconds	Specify the time interval between jobs. This applies only to trigger types maintained as **Simple**.
Trigger Listener	Specify a java class as a trigger listener which will be notified of events such as before a job is fired, after a job is completed, and after misfired jobs.
Active	Check this box to set the job as active. The scheduler does not pick the inactive jobs for scheduling.

Table 1-1 (Cont.) Job Maintenance - Field Description

Field	Description
Ds Name	Specify the name of the database schema to which the job has to connect. This attribute is used in case of multi-instance deployment of Oracle Banking Corporate Lending application.
Logging Required	Check this box to indicate that system should log each firing of the job. This helps in logging the firing time of the job and key log info as part of that firing. This also enables tracking of each jobs firing times and helps in identifying miss-fired jobs.
Veto Blocked Trigger	Check this box to trigger a veto block.
Start up Mode	Specify the start up mode of the job from the drop-down list: • **Auto** - The job starts automatically when the Oracle Banking Corporate Lending application starts. • **Manual** - Start the job manually in the job controller by resuming the job.

Parameter Details: Specify the job specific parameters, which are passed to the job class or procedure at run time. The following details are captured here.

Table 1-2 Parameter Details - Field Description

Field	Description
Parameter Name	Specify the name of the job parameter. The parameter name specified here is passed to the job class or procedure at run time.
Data Type	Specify the data type of the parameter.
Parameter Value	Specify the value of the parameter.

3. Click **Exit** to end the transaction.

1.2 Schedule Jobs

This topic provides the detailed information on schedule jobs.

All jobs for scheduling are stored in a static data store and each job is associated with a name indicating where the job has to execute. Jobs are created in the Application Server and are scheduled based on this data.

> **Note**
>
> The job name should be unique across the schedulers available in the system.

When the application server starts, the job details from the static data store will get cached. These cached jobs will then be scheduled using either the quartz or flux scheduler.

For example, the notification process can be handled by the job schedulers as follows:

1. When a contract is created in Oracle Banking Corporate Lending, a database level trigger acting on the contract main table inserts details like base table name, primary key fields, primary key values and branch code into a notification log table and sets the process status of the inserted record as **U** (unprocessed).
2. The scheduled job polls the notification log table for unprocessed records and validates whether notification is required.

3. If notification is not required, then the process status is set to **N** (not required) in notification log table.
4. If notification is required then notifications are sent to the respective destination and the process status of the record is changed to **P** (Processed) in notification log table.

1.3 Maintain Jobs

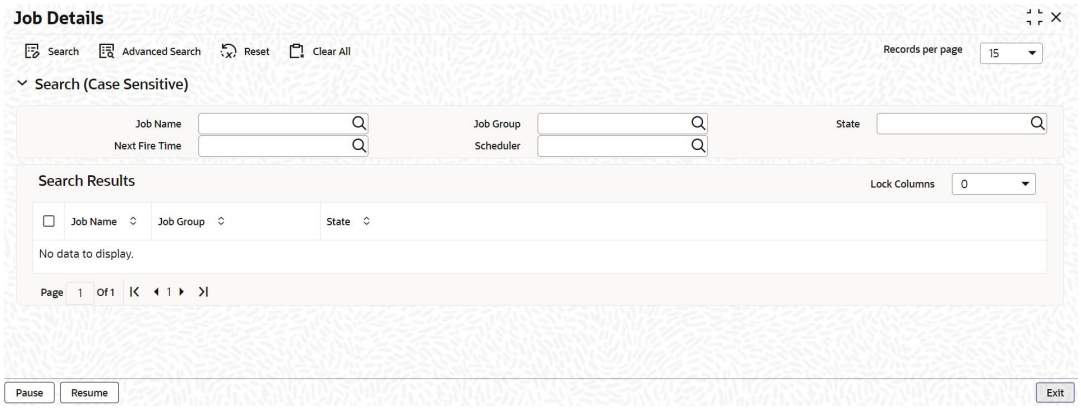
This topic explains systematic instructions to maintain jobs.

Through the **Job Details** screen, the details of scheduled jobs can be viewed. On this screen, the user can pause or resume a job that has been scheduled, and also the user can submit the records as a job for replication in the branch database.

1. On **Homescreen**, type **SMSJBBRW** in the text box, and click **Next**.

   The **Job Details** screen displays.

Figure 1-3 Job Details



2. On the **Job Details** screen, specify the fields.

> **Note**
>
> The fields, which are marked with an asterisk, are mandatory.

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

Table 1-3 Job Details - Field Description

Field	Description
Job Name	Click **Search** and select the name of the job from the list of values.
State	Click **Search** and select the state of the job. The following options are possible for Quartz schedulers: • **Acquired** • **Waiting** • **Blocked** • **Paused**
Scheduler	Click **Search** and select the scheduler to which the job has been assigned.

Table 1-3 (Cont.) Job Details - Field Description

Field	Description
Job Group	Click **Search** and select the group to which the job belongs.
Next Fire Time	Specify the time when the job is scheduled to be run next.

3. Click **Search** after specifying the search criteria.

   The system displays the records that match the search criteria:

   - **Job Name**
   - **Job Group**
   - **State**
   - **Next Fire Time**
   - **Scheduler**
   - **Error**

4. Click **Pause** to pause the job.
5. Click **Resume** to resume the paused job.

> **Note**
>
> The fields, which are marked with an asterisk, are mandatory.

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

On clicking the **Resume**, the job is scheduled for its next fire time. A job can take any of the following states:

Table 1-4 Job State

State	Description
SCHEDULED	This indicates that the message is processed.
NOT SCHEDULED	This indicates that the message processing is not scheduled.
PAUSED	This indicates that the job is manually paused from executing.
ERROR	A job trigger arrives at the **Internal Server Error** state when the scheduler attempts to fire it, but cannot due to an error creating and executing its related job, hence pausing the job. Also, a job arrives at an **ERROR** state for the following reasons: • When the associated class for the job is not present in class path. • If the during setup, the queue has not been created, but a job has been created for that queue. • If call to the Scheduler EJB has failed. • If job related pooling tables are invalid.

> **Note**
>
> Each job is created with the **Location Code** maintained in the **sttm_flexbranch_loc** table runs for the respective branch location.
> If a new branch is added to the bank for branch replication then the application must be restarted to add that particular branch in the **Job Details** screen.

6. Click **Exit** to end the transaction.

1.4 Notification Process

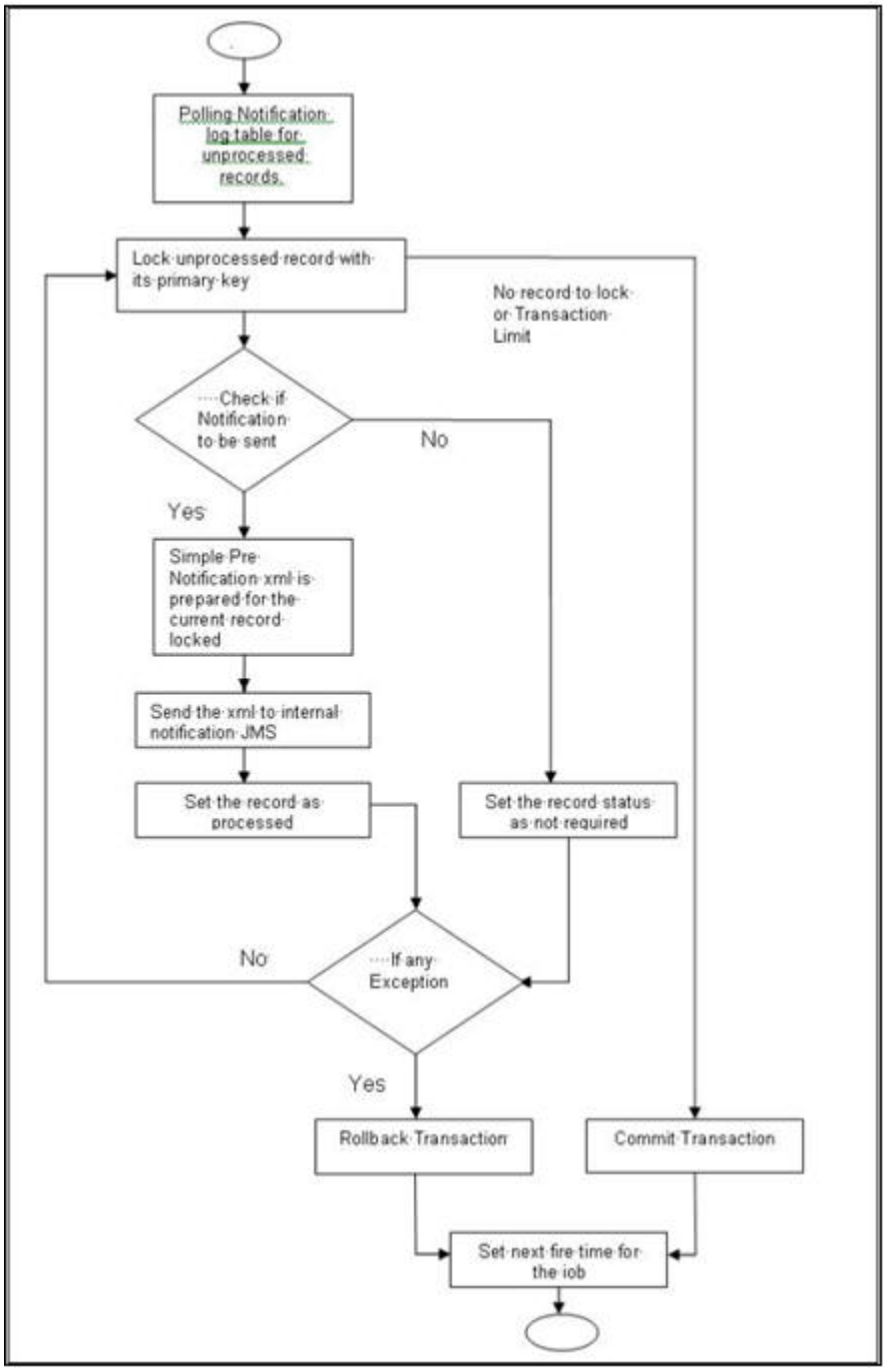
The topic describes overview of the notification process.

The notification process is in two layers. In the first layer the notification process as part of jobs in the FCJ scheduler sends minimal data required for notification to an internal JMS queue. In the second layer the notification process as part of an MDB that listens on internal JMS queue builds final notifications and sends them to their intended destinations.

The Notification Process in Oracle Banking Corporate Lending using the jobs scheduler is as follows:

1. The trigger on the base table inserts key details into a static notification log table instead of Oracle AQ.
2. Once the Job is triggered, a request is sent to the EJB layer from the job execution class and the notification log table is polled for unprocessed records.
3. Each unprocessed record is locked.
4. The record is verified against the notification maintenance and checked whether notification is to be sent or not.
5. If a notification is to be sent, a pre-notification message XML is built and it is sent to internal notify_queue(JMS queue).
6. The job is then rescheduled to fire next time based on the previous execution.

Figure 1-4 Flow Chart of Notification Process in Scheduler

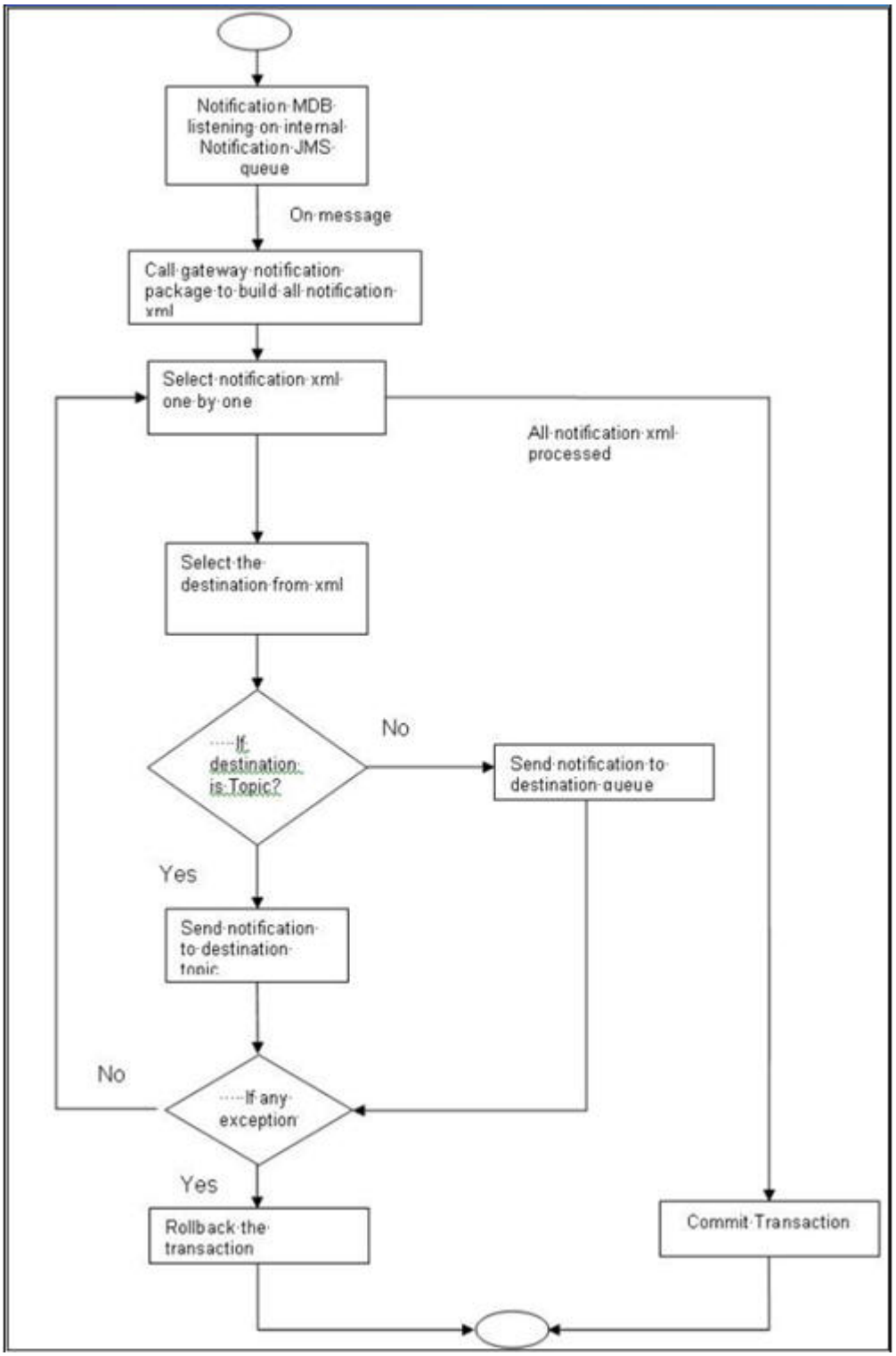


The notification process in MDB is as follows:

1. The Notification MDB listens on the internal notify JMS queue
2. On any message received, the MDB identifies which schema to connect using the JNDI name being present as part of the message XML.
3. Gateway notification processing package is called from MDB in order to build the actual notifications.

4. In MDB the notifications built is processed and sent to the destination specified in corresponding notification.
5. In case of any exception, the whole transaction is rolled back.
6. If all notifications are successfully processed then the transaction is committed.

Figure 1-5 Flow Chart of Notification Process in MDB



1.5 Process Notification Parameters

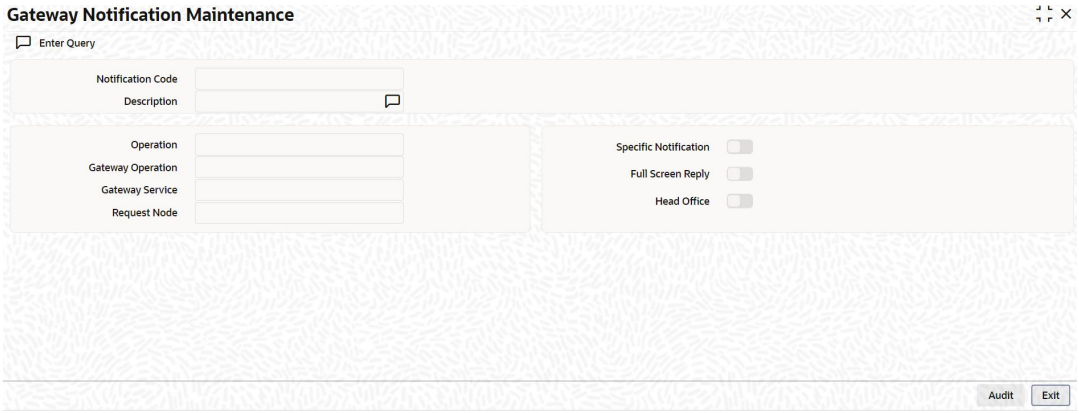
This topic explains systematic instructions to process notification parameters.

The user can view and amend certain notification parameters in Oracle Banking Corporate Lending using the **Gateway Notification Maintenance** screen.

1. On **Homescreen**, type **IFDNOTIF** in the text box, and click **Next**.

   The **Gateway Notification Maintenance** screen displays.

Figure 1-6 Gateway Notification Maintenance



2. On the **Gateway Notification Maintenance** screen, specify the fields.

> **Note**
>
> The fields, which are marked with an asterisk, are mandatory.

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

Table 1-5 Gateway Notification Maintenance - Field Description

Field	Description
Notification Code	The system displays a unique code to identify a notification.
Description	The system displays a brief description of the notification. However, the description can be modified on this screen.
Operation	Select the type of operation for the notification from the drop-down list: • **Insert** - To indicate a new operation of notification function. • **Update** - To indicate a modification operation of notification.
Gateway Operation	Specify the gateway operation name to execute the query for the mentioned service.
Gateway Service	Specify the gateway service to be used to get the full screen response.
Request Node	Specify the gateway IO request node to be used in the querying operation.

Table 1-5 (Cont.) Gateway Notification Maintenance - Field Description

Field	Description
Specific Notification	Switch the toggle button to indicate the system to send a specific notification. The system handles any deviation from the generic notification process by creating specific triggers once this field is checked.
Full Screen Reply Required	Switch the toggle button to indicate that the full-screen notification response has to be sent. Otherwise, the primary key response notification is sent.
Head Office	Switch the toggle button to send notification only from head office.

3. Click **Exit** to end the transaction.

1.6 EMS Process with Scheduling Architecture

This topic provides the information of EMS process with Scheduling Architecture.

The topic contains following sub-topic:

- EMS Process
  The topic describes the EMS process.

1.6.1 EMS Process

The topic describes the EMS process.

- **Incoming EMS Process**
  A job is scheduled to poll the incoming folder on a timely basis. Once a message is received in the folder, the job picks the message and sends it to an internal JMS queue. An MDB listening on the queue will read the message and identifies the media and processes the message.
- **Outgoing EMS Process**
  A job is scheduled to poll the outgoing messages that are generated but not handed off. Each messages polled will be sent to an internal JMS queue. An MDB, acting upon the internal JMS queue will pick the message from the queue and sends the message to the appropriate destination (Folder, e-mail, or JMS queue).

1.7 Approach

This topic provides the approaches used in EMS Process.

The Outgoing EMS Process happens in two layers.

The EMS process as part of jobs in FCJ scheduler, polls the outgoing message table of Oracle Banking Corporate Lending for generated and un-send messages. The job then sends minimal data about the message to be handed off, to an internal JMS queue.

The EMS process is part of an MDB that listens on the internal JMS queue to build the final message and to send to their intended destinations.

The Incoming EMS Process happens in two layers.

The EMS process as part of jobs in FCJ scheduler, which polls the pre-configured folder for messages and sends the messages read, to EMS internal queue.

The EMS process is part of an MDB, that listens to the internal JMS queue identifies the message from the queue, and calls the incoming messages service package in the backend to process the message. Additionally, the MDB can be made an independent unit to listen to external JMS to process incoming messages.

The Incoming EMS Process as part of the jobs scheduler is as follows:

Once the job is triggered, it polls for messages in a folder (Configured for incoming messages).

Each message is then sent to an internal JMS queue.

The job is then rescheduled to fire next time.

EMS processes in MDB are as follows:

An MDB that listens on the internal EMS incoming queue will receive the message.

The media details are identified and the incoming message processing package in the backend is called to process the message.

In case of any exception while processing, message will be sent to a deferred queue.

In case of messages directly arrive to the JMS queue instead of a folder; the same MDB will be configured to listen on a specific queue.

Index

I

S