Common Core - Scheduler User Guide

Oracle FLEXCUBE Universal Banking

Release 14.8.3.0.0

Part No. G61377-01

October 2026

ORACLE

Common Core - Scheduler 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 © 2007, 2026, 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

1.1 Introduction 1-1
1.2 Audience 1-1
1.3 Documentation Accessibility 1-1
1.4 Critical Patches 1-1
1.5 Acronyms and Abbreviations 1-1
1.6 Organization 1-1
1.7 Glossary of Icons 1-2

2. Job Scheduling 2-1

2.1 Defining Jobs 2-1
2.2 Scheduling Jobs 2-4
2.3 Controlling Jobs 2-5
2.4 Notification Process 2-6
2.5 Notification Retry 2-8
2.6 Viewing Notification Parameters 2-9
2.7 EMS Process with Scheduling Architecture 2-10
2.7.1 EMS Process: 2-10
2.8 Approach 2-10

3. Function ID Glossary 3-1

1. Preface

1.1 Introduction

This manual is designed to help acquaint you with the Job scheduling process in Oracle FLEXCUBE.

1.2 Audience

This manual is intended for the following User/User Roles:

Role	Function
Back office data entry Clerks	Input functions for maintenance related to the interface.
Back office Managers/Officers	Authorization functions.

1.3 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.

1.4 Critical Patches

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

1.5 Acronyms and Abbreviations

Abbreviation	Description
System	Unless and otherwise specified, it shall always refer to Oracle FLEX-CUBE system

1.6 Organization

This manual is organized as follows:

Chapter	Description
Chapter 1	*About this Manual* gives information on the intended audience. It also lists the various chapters covered in this User Manual.
Chapter 2	*Job Scheduling* explains about the process of defining, scheduling, and maintaining Jobs within the module.
Chapter 3	*Function ID Glossary* has alphabetical listing of Function/Screen ID's used in the module with page references for quick navigation.

1.7 Glossary of Icons

This User Manual may refer to all or some of the following icons.

Icons	Function
	Exit
	Add row
	Delete row
	Option List

2. Job Scheduling

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

- Quartz - provides scheduler interface to enable operations such as scheduling and un-scheduling of jobs and starting, stopping, pausing the scheduler
- Flux - software component used for performing enterprise job scheduling

This chapter contains the following sections

- Section 2.1, "Defining Jobs"
- Section 2.2, "Scheduling Jobs"
- Section 2.3, "Controlling Jobs"
- Section 2.4, "Notification Process"
- Section 2.5, "Notification Retry"
- Section 2.6, "Viewing Notification Parameters"
- Section 2.7, "EMS Process with Scheduling Architecture"
- Section 2.8, "Approach"

2.1 Defining Jobs

A job is a business activity which the system performs repeatedly on timely basis. Oracle FLEXCUBE enables you to define a job and schedule it using ‘Job Maintenance’ screen. You

can invoke this screen by typing ‘STDJOBMT’ in the field at the top right corner of the Application tool bar and clicking on the adjoining arrow button.



You can specify the following fields in this screen.

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. The following options are available for selection:

- PL/SQL
- JAVA

Max Number Instances

Specify the maximum number of instances that needs to be queued up.

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 job’s behavior in such cases,

If you maintain the job 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 you maintain the job as ‘STATELESS’, it indicates the number of threads that can be executed in parallel.

If you maintain the max number instances 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, you can modify this name. This signifies the scheduler name which is configured as part of infra.

Trigger Type

Select the type of the trigger from the drop-down list. The following options are available:

- Simple - Interval based jobs.(i.e., every one hour)
- Cron - Time based jobs.(i.e., Friday 4:30PM)

Note

When 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. The following options are available:

- Quartz
- Flux

Priority

Select the priority on which the system should execute the jobs in the scheduler from the drop-down list. The following options are available.

- Normal
- High
- Low

If two jobs with different priorities fire at the same time, then system gives preference to the job with higher priority.

Message Queue

Specify the default JMS queue to which a job needs to send message. You can specify this only if the job has to send messages to JMS.

Cron Expression

Specify the corresponding Cron expression for a job with trigger type as ‘Cron’. You need to do this to determine the time and interval of job firing.

Class or Procedure

Specify the Java class file name if 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 scheduler. This applies only to trigger types maintained as ‘Simple’.

Interval

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 job fired, after job completed, misfired jobs.

Active

Check this box to set the job as active. The scheduler does not pick the inactive jobs for scheduling.

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 FLEXCUBE application.

Logging Required

Check this box to indicate that system should log each firing of job. This helps in logging the firing time of job and key log info as part of that firing. This also enables tracking of each job's firing times and helps in identifying miss-fired jobs.

Startup Mode

Specify start up mode of the job from the drop-down list. The following options are available:

- Auto - The job starts automatically when Oracle FLEXCUBE application starts
- Manual - You should start the job manually in job controller by resuming the job.

Parameter Details

You can specify the job specific parameters, which are passed to job class or procedure at runtime. The following details are captured here:

Parameter Name

Specify the name of the job parameter. The parameter name you specify here is passed to job class or procedure at run time.

Data Type

Specify the data type of the parameter.

Parameter Value

Specify the value of the parameter.

2.2 Scheduling 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 static data store will get cached. These cached jobs will then be scheduled using either the quartz or flux scheduler.

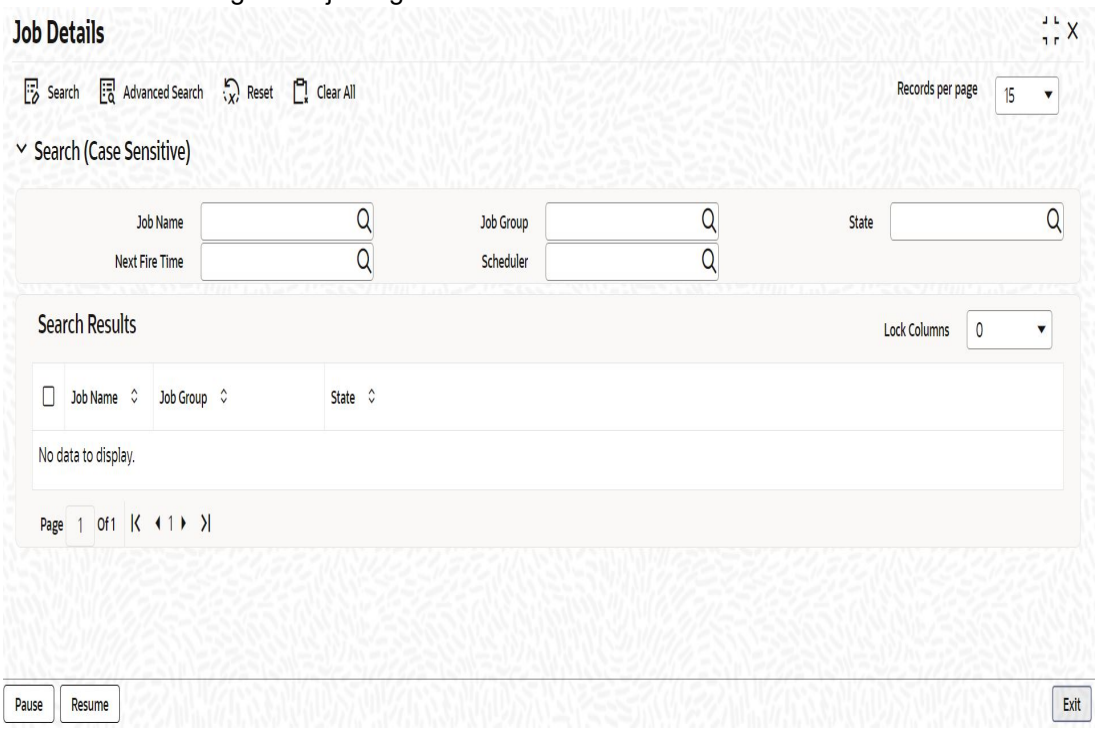
For example, the notification process can be handled by the job schedulers as follows:

1. When a contract is created in Oracle FLEXCUBE, 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.
   - 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.
   - If notification is not required, then the process status is set to ‘N’ (not required) in notification log table.

2.3 Controlling Jobs

The details of jobs that are scheduled can be viewed using the ‘Job Details’ screen. In this screen you can pause or resume a job that has been scheduled. You can submit the records as a job for replication in the branch database through this screen. You can invoke the ‘Job Details’ screen by typing ‘SMSJBBRW’ in the field at the top right corner of the Application tool bar and clicking the adjoining arrow button.



You can search for a scheduled job by specifying any of the following:

Job Name

Select the name of the job that you want to search for from the option list provided.

State

Select the state of the job you want to search for from the option list provided. The following options are possible for Quartz schedulers:

- Acquired
- Waiting
- Blocked
- Paused

Scheduler

Select the scheduler to which the job you want to search for has been assigned.

Job Group

Select the group to which the job you want to search for belongs, from the option list provided.

Next Fire Time

Select the time when the job is scheduled to be run next.

Click ‘Search’ button to view the details related to the job. You can pause a job by selecting it and clicking the ‘Pause’ button.

You can resume a paused job by clicking ‘Resume’ button and the job is scheduled for its next fire time.

A job can take any of the following states.

- 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 ERROR state for the following reasons.:
  - When the associated class for the job is not present in class path
  - If 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 table sttm_flexbranch_loc runs for the respective branch location.

If a new branch is added to the bank for branch replication then the application should be restarted to add that particular branch in the ‘Job Details’ screen.

2.4 Notification Process

The notification process is in two layers. In the first layer the notification process as part of jobs in 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.

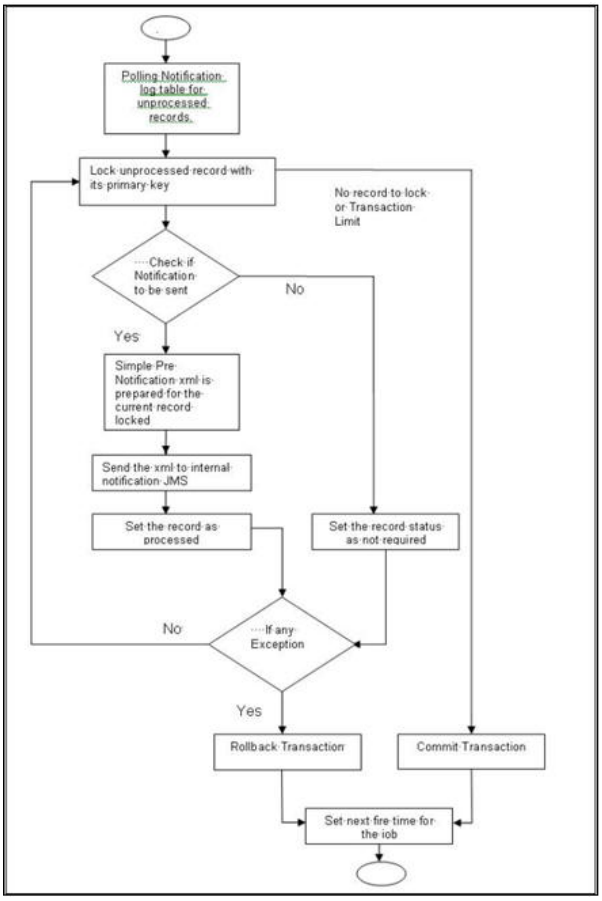
The Notification Process in Oracle FLEXCUBE using the jobs scheduler is as follows:

1. The trigger on the base table inserts key details into a static notification log table instead of Oracle AQ.
2. Once Job is triggered, a request is sent to EJB layer from 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 notification is to be sent, 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.

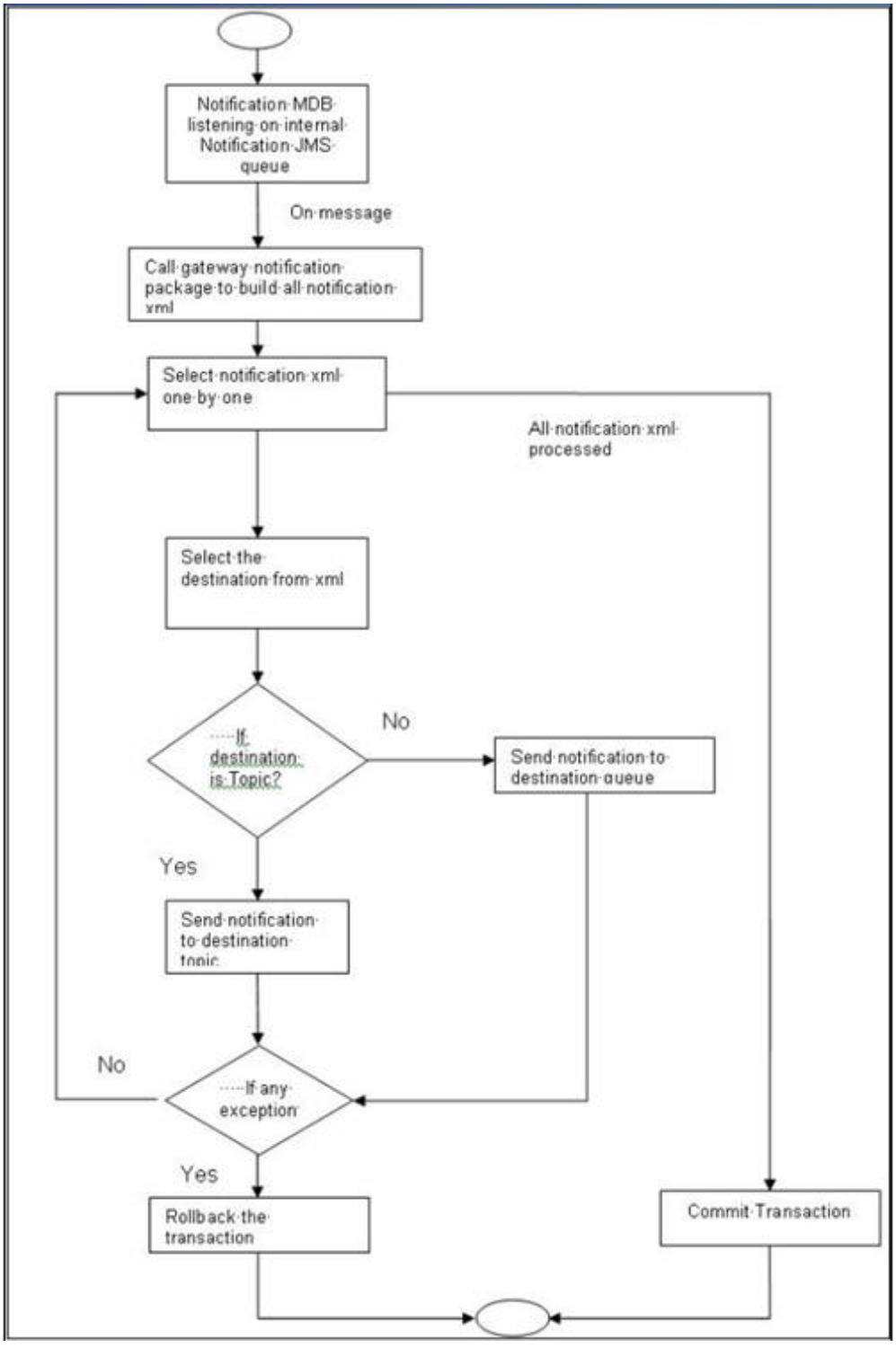
The notification process in MDB is as follows:

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 transaction is committed.

The flow chart of notification process in scheduler:



The flow chart for notification process in MDB:



2.5 Notification Retry

The notification retry feature enables automatic processing of failed notification records without needing user intervention. Failures in notification can occur due to SSL handshake or temporary connection issues, leading to records being unprocessed and marked as ‘E’. This feature ensures that these failed records can be retried according to the configured retry criteria once the root issue is resolved.

A new feature callled NOTIF_RETRY is maintained through the ‘Feature Maintenance’ screen (CSDFTRMN) to retry notification messages. It is enabled for the Head Office branch through the ‘Feature ID Maintenance’ screen (CSDFEMNT). After enabled, the PL/SQL scheduler job can be configured using the Job Maintenance screen (STDJOBMT). This job runs on a Quartz scheduler at a configurable interval.

Note

The shorter interval duration will not help for retries in case of connection failures. The recommended interval is every 2 hours; however, the bank can adjust the interval as required.

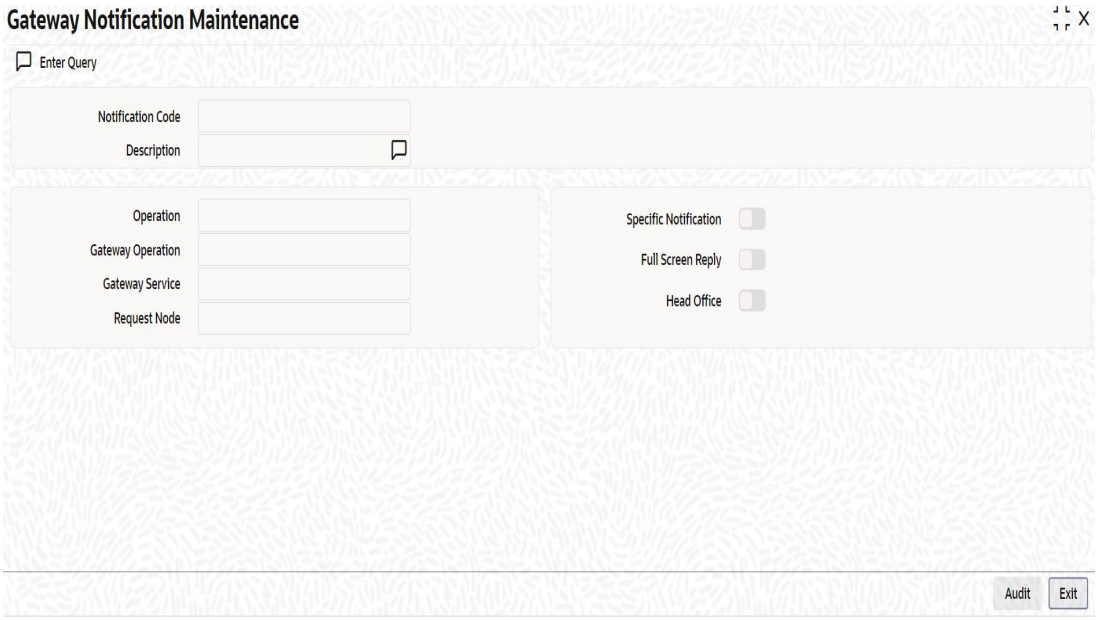
The retry job can be executed only if the NOTIF_RETRY feature is installed. On each execution, the job checks the current retry count against the configured maximum count:

- If the retry count has not reached the maximum, the count is incremented and the status is updated from ‘E’ (Error) to ‘U’ (Unprocessed).
- If the retry count has reached the maximum, the status remains ‘E’ and no further retries are attempted.

The existing Notification job remains unchanged and continues with the status as ‘E’. When the retry job resets the records to ‘U’, they are automatically picked up and processed again. The user can stop or resume the jobs using Job Details screen (SMSJBBRW).

2.6 Viewing Notification Parameters

You can view and amend certain notification parameters in Oracle FLEXCUBE using ‘Gateway Notification Maintenance’ screen. You can invoke this screen by typing ‘IFDNOTIF’ in the field at the top right corner of the Application tool bar and clicking on the adjoining arrow button.



Notification Code

The system displays a unique code to identify a notification.

Description

The system displays a brief description of the notification. However, you can modify the description in this screen.

Operation

Select the type of operation for the notification from the following.

- 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 query for the mentioned service.

Gateway Service

Specify the gateway service to be used to get the full screen response.

IO Request Node

Specify the gateway IO request node to be used in querying operation.

Specific Notification

Check this box to indicate the system to send specific notification. The system handles any deviation from generic notification process by creating specific triggers once you check this field.

Full Screen Reply Required

Check this box to indicate that the full screen notification response has to be sent. Otherwise, the primary key response notification is sent.

Head Office

Check this box to send notification only from head office.

2.7 EMS Process with Scheduling Architecture

2.7.1 EMS Process:

Incoming EMS Process

A job is scheduled to poll the incoming folder on 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.

A MDB, acting upon the internal JMS queue will pick the message from queue and sends the message to appropriate destination (Folder, or e-mail, or JMS queue).

2.8 Approach

The Outgoing EMS Process happens in two layers.

The EMS process as part of jobs in FCJ scheduler, polls the outgoing message table of FLEXCUBE 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 as part of an MDB that listens on internal JMS queue to build 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 as part of an MDB, that listens on internal JMS queue identifies the message from queue and calls the incoming messages service package in backend to process the message. Additionally, the MDB can be made an independent unit to listen on external JMS to process incoming messages.

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

Once 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 incoming message processing package in 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 JMS queue instead of a folder; the same MDB will be configured to listen on specific queue.

3. Function ID Glossary

I

IFDNOTIF 2-9

S

SMSJBBRW 2-5
STDJOBMT 2-2