Oracle® FLEXCUBE Investor Servicing

Scheduler User Guide

Release 14.7.7.0.0
G31915-01
April 2025

ORACLE

Oracle FLEXCUBE Investor Servicing Scheduler User Guide, Release 14.7.7.0.0

G31915-01

Contents

Preface

Oracle FLEXCUBE Investor Servicing is a comprehensive mutual funds automation software from Oracle® Financial Servicing Software Ltd.©.

You can use the system to achieve optimum automation of all your mutual fund investor servicing processes, as it provides guidelines for specific tasks, descriptions of various features and processes, and general information.

This topic contains the following sub-topics:

Purpose

You are intended to become familiar with the **Oracle Flexcube Investor Servicing** application through this guide. This guide offers responses to particular features and procedures that are necessary for the module to operate effectively.

Audience

This user guide is intended for the Fund Administrator users and System operators in the AMC.

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.

Critical Patches

Oracle advises customers to get all their security 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.

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 or the glossary.
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.

Screenshot Disclaimer

Personal information used in the interface or documents is dummy and does not exist in the real world. It is only for reference purposes.

Acronyms and Abbreviations

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

Table Acronyms and Abbreviations

Abbreviation	Description
CIF	Customer Information File
EOD	End of Day
EPU	Earnings per unit
FCIS	Oracle FLEXCUBE Investor Servicing

Table (Cont.) Acronyms and Abbreviations

Abbreviation	Description
FMG	The Fund Manager component of the system
FPADMIN	Oracle FLEXCUBE Administrator
GTA	Global Transfer Agency
ID	Identification
IHPP	Inflation Hedged Pension Plan
IPO	Initial Public Offering
LEP	Life and Endowment Products
LOI	Letter of Intent
NAV	Net Asset Value
REG	The Registrar component of the system
ROA	Rights of Accumulation
ROI	Return on Investment
SI	Standing Instructions
SMS	Security Management System
URL	Uniform Resource Locator
VAT	Value Added Tax
WAUC	Weighted Average Unit Cost

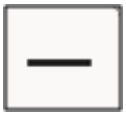
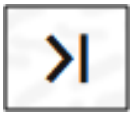
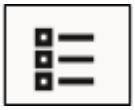
Symbols and Icons

This guide may refer to all or some of the following symbols and icons:

Table Symbols and Icons

Symbol/Icon	Function
	Lists all records maintained
	Minimize
	Maximize
	Close
	Perform Search

Table (Cont.) Symbols and Icons

Symbol/Icon	Function
	Open a list
	Select a Date
	Add a new row to enter details in a record.
	Delete a row, which is already added.
	Navigate to the first record
	Navigate to the last record
	Navigate to the previous record
	Navigate to the next record
	View a single record
	Sort the values in ascending or descending order
	Sort the values in ascending
	Sort the values in ascending

Basic Actions

Following are the basic actions of the screens that an user may require to perform on new or existing records in a screen.

Table Basic Actions

Action	Description
New	Used to add a new record. When the user click **New**, the system displays a new record enabling to specify the required data. **Note**: The fields, which are marked with an asterisk, are mandatory.
Copy	Used to copy the details of a record.
Close	Used to close a record. This action is available only when a record is created.
Unlock	Used to update the details of an existing record. System displays an existing record in editable mode.
Print	Used to print a record. This action is available only when a record is created.
Enter Query	Used to give details of a saved record in a detail screen. When the user click **Enter Query**, the system displays a saved record enabling to specify only the required or primary data.
Execute Query	User need to perform this after entering query. Click **Execute Query** after specifying the details of the record to be fetched, the system retrieves all the information of that particular record.
Audit	Used to view the maker details, checker details and report status.
Cancel	Used to cancel the performed action.
Save	Used to save the details entered or selected in the screen.
Refresh	Used to refresh the details selected in the screen.
Reset	Used to reset the fields to enter a new criteria.
Clear All	Used to clear all the data entered for search criteria.
Details	Used to navigate to Detail screen.
Search	Used to search either the details of a particular record or a list of records by querying particular field.
Advanced Search	Used to search details more precisely.
Approve	Used to approve the initiated report. This button is displayed, once the user click **Authorize**.
Authorize	Used to authorize the report created. A maker of the screen is not allowed to authorize the report. Only a checker can authorize a report, created by a maker.
Confirm	Used to confirm the performed action.
OK	Used to confirm the details in the screen.
Reject	Used to reject the report created. A maker of the screen is not allowed to authorize the report. Only a checker can reject a report, created by a maker.

Table (Cont.) Basic Actions

Action	Description
View	Used to view the report details in a particular modification stage. This button is displayed, once the user click **Authorize**.

Getting Help

Online help is available for all tasks. You can get help for any function or fields by clicking the help icon provided or by pressing **F1**.

Prerequisite

Specify **User ID** and **Password**, and log in to **Home Screen**.

1

Job Scheduling

This topic provides an overview on job scheduling.

Job Scheduling is the process where different tasks get executed at pre-determined time or when the right event happens.

The **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 Investor Servicing** FCJ architecture are as follows:

- **Quartz** provides scheduler interface to enable operations such as scheduling and unscheduling of jobs and starting, stopping, pausing the scheduler.

This topic contains the following subtopics:

1.1 Process Job Maintenance

This topic provides the systematic instructions to define and schedule a job.

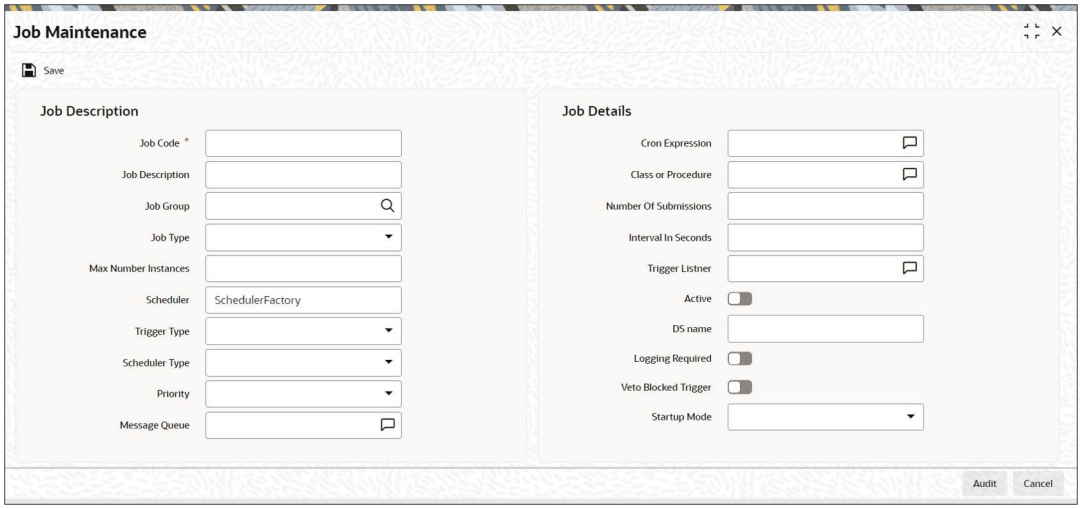
A job is a business activity which the system performs repeatedly on timely basis.

Oracle® FLEXCUBE Investor Servicing enables you to define a job and schedule it using **Job Maintenance** screen.

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

   The **Job Maintenance** screen is displayed.

Figure 1-1 Job Maintenance



2. On **Job Maintenance** screen, click **New** to enter the details.

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

Table 1-1 Job Maintenance - Field Description

Field	Description
Job Code	*Alphanumeric; 100 Characters; Mandatory* Specify the unique code to identify the Job.
Job Description	*Alphanumeric; 200 Characters; Optional* Specify a brief description of what the job is supposed to do.
Job Group	*Alphanumeric; 200 Characters; Optional* Specify the job group name to represent the same group of jobs for identification.
Job Type	*Optional* Select the type of job from the drop-down list. The following options are available for selection: • **PLSQL** • **JAVA**
Max Number Instances	*Numeric; 2 Characters; Mandatory* 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.

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

Field	Description
Scheduler	*Alphanumeric; 100 Characters; Optional* 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	*Optional* 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)
Scheduler Type	*Optional* Select the type of scheduler from the drop-down list. The following options are available: • **Quartz**
Priority	*Optional* 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	*Alphanumeric; 100 Characters; Optional* 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.
Job Details	It displays the following details.
Cron Expression	*Alphanumeric; 400 Characters; Optional* 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. For instance, 40 * * * * ? Indicates job needs to be executed every 40 sections. Example: For time based jobs, if Cron Expression is defined as 0 0/15 11-23 * * ?, then Parameter Definition will be as follows: • **TIMEZONE** Parameter to be defined: Asia/Hong Kong • **MODULEID**: Job to be executed in FMGHKGDISTE • **INTERFACEID**: Interface ID to be mentioned Above scenario indicates that the Job to be started at 11:00 am Hong Kong Timezone and stops at 11:00 pm Hong Kong Timezone. Frequency will be every 15 minutes.
Class or Procedure	*Alphanumeric; 400 Characters; Optional* 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	*Alphanumeric; 3 Characters; Optional* Specify the number of times a job can fire before it is unscheduled from scheduler. This applies only to trigger types maintained as **Simple**.

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

Field	Description
Interval In Seconds	*Numeric; 2 Characters; Optional* Specify the time interval between jobs. This applies only to trigger types maintained as **Simple**.
Trigger Listener	*Alphanumeric; 200 Characters; Optional* 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	*Optional* Check this box to set the job as active. The scheduler does not pick the inactive jobs for scheduling.
Ds Name	*Alphanumeric; 100 Characters; Optional* 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 Investor Servicing** application.
Logging Required	*Optional* 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.
Veto Blocked Trigger	*Optional* Check this box to trigger veto block.
Startup Mode	*Optional* 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 Investor Servicing** 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	*Alphanumeric; 200 Characters; Optional* Specify the name of the job parameter. The parameter name you specify here is passed to job class or procedure at run time. For instance, **INTERFACEID** to be defined with Interface ID.
Data Type	*Optional* Select the data type of the parameter from the drop-down list. The list displays the following values: • **Varchar** • **Date** • **Number**
Parameter Value	*Alphanumeric; 200 Characters; Optional* Specify the value of the parameter.

3. On **Job Maintenance** screen, process EOD.

   On completion of Pre-EOD, the job will process EOD.

Table 1-2 Event Based - EOD

Job Code	EOD_FMGHKGDISTE
EVENTCODE	BEPREEOD
Event Description	Pre-EOD completed

4. On **Job Maintenance** screen, process BOD.

   On completion of EOD, the job will process BOD.

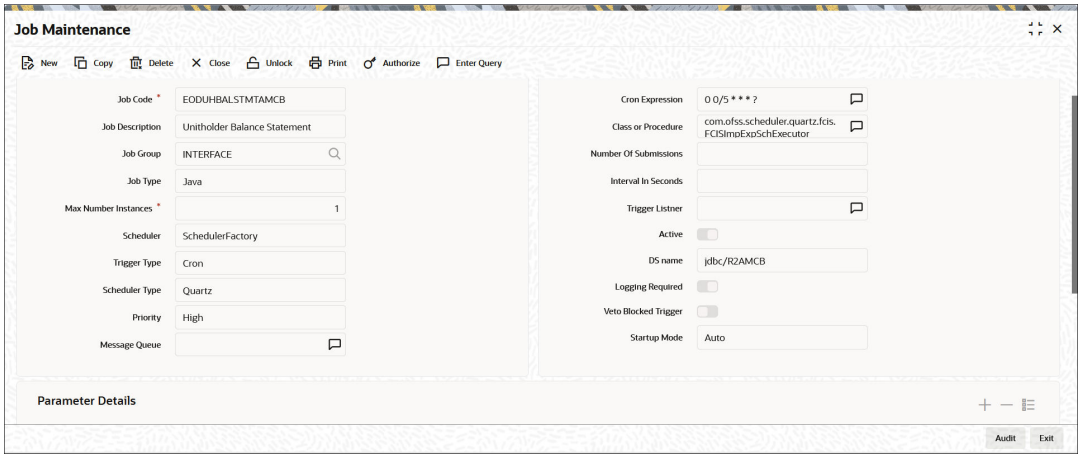
Table 1-3 Event Based - BOD

Job Code	EOD_FMGHKGDISTE
EVENTCODE	BE00003
Event Description	FMG EOD Completed

5. The system will perform the following jobs to execute reports on completion of EOD.
   - Parameter Name: TASKID
   - Parameter Value: Tasks defined
6. Below job will initiate a TASK – EODUHBALSTMT – Unitholder Balance Statement on completion of EOD.

   The TASK – EODUHBALSTMT is initiated.

Figure 1-2 TASK – EODUHBALSTMT



1.2 Job Maintenance Summary

This topic provides the systematic instructions to perform the basic operations on the selected records.

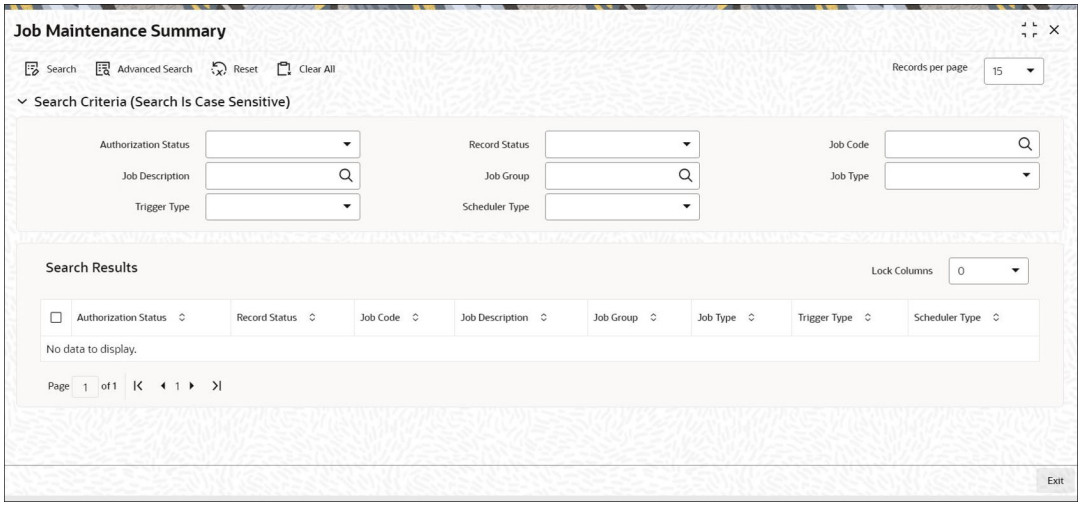
Specify **User ID** and **Password**, and log in to **Home Screen**.

Retrieve Job Maintenance Record

1. On **Home** screen, type **STSJOBMT** in the text box, and click **Next**.

   The **Job Maintenance Summary** screen is displayed.

Figure 1-3 Job Maintenance Summary



2. On **Fund Dividend Summary** screen, specify any or all of the following details in the corresponding fields:
   - **Authorization Status** - If you choose the status, then the records matching the specified status are retrieved. If you do not choose any option, then all the records are retrieved.
   - **Record Status**
   - **Job Code**
   - **Job Group**
   - **Trigger Type**
   - **Job Description**
   - **Job Type**
   - **Scheduler Type**
3. Click **Search** button to view the records.

   All the records with the specified details are retrieved and displayed in the screen.

> **Note:**
>
> You can also retrieve the individual record detail from the detail screen by querying in the following manner:
>
> - Press F7
> - Input the **Job Code**
> - Press F8

4. Perform **Edit**, **Delete**, **Amend**, and **Authorize** operations by selecting the desired operation from the Action list. You can also search a record by using a combination of % and alphanumeric value.

- Edit Job Maintenance Record
  This topic provides the systematic instructions to edit Job Maintenance record.

- View Job Maintenance Record
  This topic provides the systematic instructions to view Job Maintenance record.
- Delete Job Maintenance Record
  This topic provides the systematic instructions to delete Job Maintenance record.
- Authorize Job Maintenance Record
  This topic provides the systematic instructions to authorize Job Maintenance record.
- Amend Job Maintenance Record
  This topic provides the systematic instructions to amend Job Maintenance record.
- Authorize Amended Job Maintenance Record
  This topic provides the systematic instructions to authorize amended Job Maintenance record.

1.2.1 Edit Job Maintenance Record

This topic provides the systematic instructions to edit Job Maintenance record.

Modify the details of **Job Maintenance** that you have already entered into the system, provided it has not subsequently authorized. Perform this operation as follows:

1. Start the **Job Maintenance Summary** screen from the Browser.
2. Select the status of the record that you want to retrieve for modification in the **Authorization Status** field.

   You can only modify records that are unauthorized. Accordingly, choose the **Unauthorized** option.
3. Specify any or all of the details in the corresponding fields to retrieve the record that is to be modified.
4. Click **Search** button.

   All unauthorized records with the specified details are retrieved and displayed in the screen.
5. Double-click the record that you want to modify in the list of displayed records.

   The **Job Maintenance** screen is displayed.
6. Select **Unlock** operation from the Action list to modify the record. Modify the necessary information.
7. Click **Save** to save your changes.

   The **Job Maintenance** screen is closed and the changes made are reflected in the **Job Maintenance Summary** screen.

1.2.2 View Job Maintenance Record

This topic provides the systematic instructions to view Job Maintenance record.

View a record that you have previously input by retrieving the same in the Job Maintenance Summary screen. Perform this operation as follows:

1. Start the **Job Maintenance Summary** screen from the Browser.
2. Select the status of the record that you want to retrieve for viewing in the **Authorization Status** field.

   You can also view all records that are either unauthorized or authorized only, by choosing the Unauthorized/Authorized option.

3. Specify any or all of the details of the record in the corresponding fields on the screen and click **Search** button.

   All records with the specified fields are retrieved and displayed in the screen.

4. Double-click the record that you want to view in the list of displayed records.

   The **Job Maintenance** screen is displayed.

1.2.3 Delete Job Maintenance Record

This topic provides the systematic instructions to delete Job Maintenance record.

Delete a record that you have previously entered. You can delete only unauthorized records in the system as follows:

1. Start the **Job Maintenance Summary** screen from the Browser.
2. Select the status of the record that you want to retrieve for deletion.
3. Specify any or all of the details and click **Search** button.

   All records with the specified fields are retrieved and displayed in the screen.

4. Double-click the record that you want to delete in the list of displayed records.

   The **Job Maintenance** screen is displayed.

5. Select **Delete** operation from the Action list.

   The system prompts you to confirm the deletion and the record is physically deleted from the system database.

1.2.4 Authorize Job Maintenance Record

This topic provides the systematic instructions to authorize Job Maintenance record.

Authorize an unauthorized job maintenance record in the system for it to be processed as follows:

1. Start the **Job Maintenance Summary** screen from the Browser.
2. Select the status of the record that you want to retrieve for authorization. Typically, choose the **Unauthorized** option.
3. Specify any or all of the details and click **Search** button.

   All records with the specified details that are pending authorization are retrieved and displayed in the screen.

4. Double-click the record that you wish to authorize.

   The **Job Maintenance** screen is displayed.

5. Select **Authorize** operation from the Action List.

   When a checker authorizes a record, details of validation, if any, that were overridden by the maker of the record during the **Save** operation are displayed. If any of these overrides results in an error, the checker must reject the record.

1.2.5 Amend Job Maintenance Record

This topic provides the systematic instructions to amend Job Maintenance record.

Modify the details of an authorized record using the **Unlock** operation from the Action List. To make changes to a record after authorization:

1. Start the **Job Maintenance Summary** screen from the Browser.
2. Select the status of the record that you want to retrieve for amendment.

   You can only amend authorized records.
3. Specify any or all of the details and click **Search** button.

   All records with the specified details are retrieved and displayed in the screen.
4. Double-click the record that you wish to amend.

   The **Job Maintenance** screen is displayed.
5. Select **Unlock** operation from the Action List to amend the record.
6. Amend the necessary information and click **Save** to save the changes.

1.2.6 Authorize Amended Job Maintenance Record

This topic provides the systematic instructions to authorize amended Job Maintenance record.

Authorize an amended Job Maintenance Record for the amendment to be made effective in the system. The authorization of amended records can be done only from Fund Manager Module and Agency Branch module.

The subsequent process of authorization is the same as that for normal transactions.

1.3 Schedule Jobs

This topic provides instructions to schedule jobs that are stored in a static data store.

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 **Quartz** scheduler.

- When a contract is created in **Oracle® FLEXCUBE Investor Servicing**, 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 **Unprocessed (U)**).
- The scheduled job polls the notification log table for unprocessed records and validates whether notification is required.

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

1.4 Process Job Details

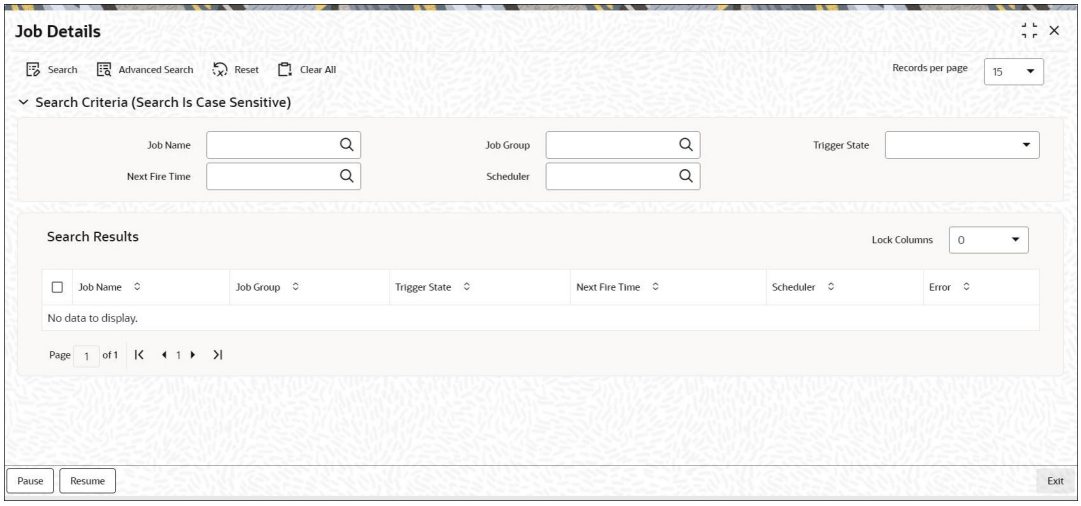
This topic provides the systematic instructions to control and view the details of jobs that are scheduled.

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.

1. On **Home** screen, type **SMSJOBBR** in the text box, and click **Next**.

   The **Job Details** screen is displayed.

Figure 1-4 Job Details



2. On **Job Details**, you can view the details related to the job.

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

Table 1-4 Job Details - Field Description

Field	Description
Job Name	*Alphanumeric; 100 Characters; Optional* Select the name of the job that you want to search for from the option list provided.
Trigger State	*Optional* Select the state of the job you want to search for from the option list provided. The following options are possible for Quartz schedulers: • **Not Scheduled** • **Scheduled** • **Paused** • **Complete** • **Error**

Table 1-4 (Cont.) Job Details - Field Description

Field	Description
Scheduler	*Alphanumeric; 100 Characters; Optional* Select the scheduler to which the job you want to search for has been assigned.
Job Group	*Alphanumeric; 200 Characters; Optional* Select the group to which the job you want to search for belongs, from the option list provided.
Next Fire Time	*Numeric; 22 Characters; Optional* Select the time when the job is scheduled to be run next.

3. Click **Search** to view the details related to the job.
4. Click the **Pause** button to pause a job by selecting it.
5. Click the **Resume** button to resume a paused job and the job is scheduled for its next fire time.

- States in Job Details
  This topic gives an overview on different status that a job can take in Job Details.

1.4.1 States in Job Details

This topic gives an overview on different status that a job can take in Job Details.

A job can take any of the following states:

1. **SCHEDULED** -This indicates that the message is processed.
2. **NOT SCHEDULED** -This indicates that the message processing is not scheduled.
3. **PAUSED** - This indicates that the job is manually paused from executing.
4. **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.

1.5 Notification Process

This topic provides information on notification process using the jobs scheduler.

The notification process is in two layers.

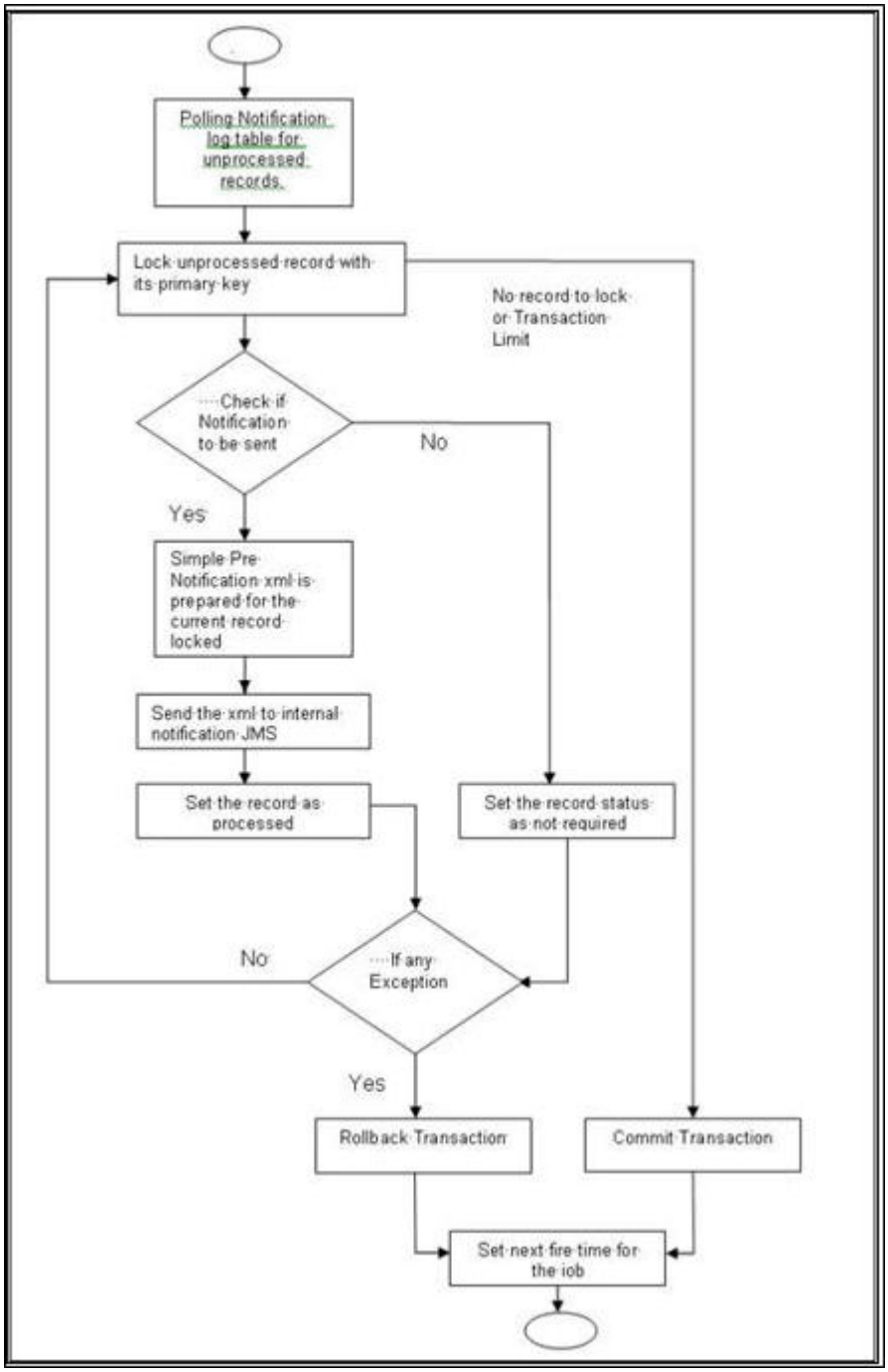
- In the first layer, the notification process as part of jobs in **FCJ Scheduler** sends minimal data required for notification to an internal JMS queue.
- In the second layer, the notification process as part of an **MDB** that listens on internal JMS queue builds final notifications and sends them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

The notification process using the jobs **FCJ 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 flow chart of notification process in **FCJ Scheduler**:

Figure 1-5 JobScheduling_NotificationProcess_SchedulerFlowchart



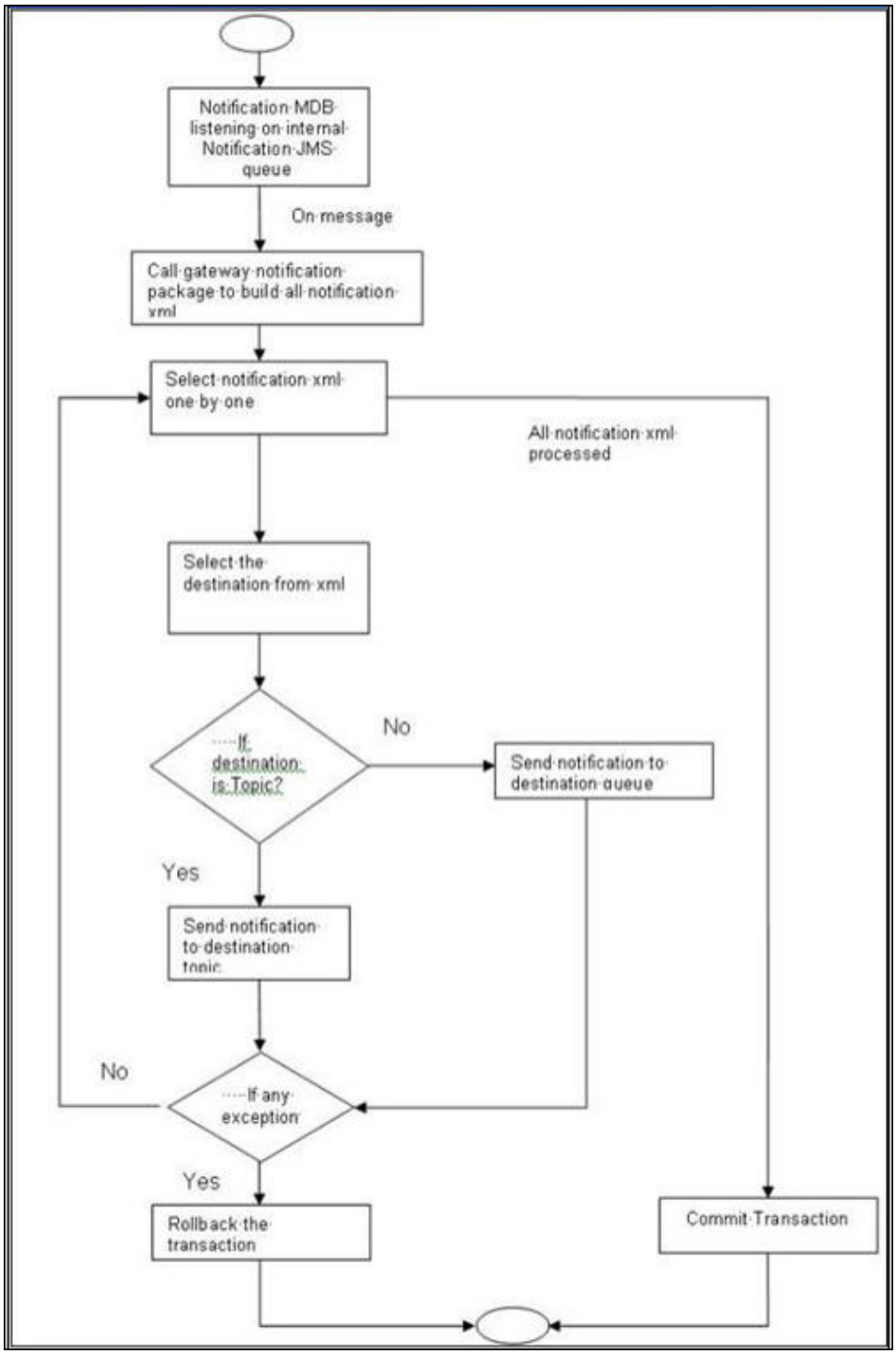
The notification process in **MDB** is as follows:

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 for notification process in **MDB**:

Figure 1-6 JobScheduling_NotificationProcess_MDBFlowchart



1.6 EMS Process with Scheduling Architecture

This topic provides information on EMS Process with Scheduling Architecture.

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).

1.7 Approach

This topic provides information on approach using the jobs scheduler.

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® FLEXCUBE Investor Servicing** for generated and unsent 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 **FCJ scheduler** is as follows:

1. Once job is triggered, it polls for messages in a folder (Configured for incoming messages).
2. Each message is then sent to an internal JMS queue.
3. The job is then rescheduled to fire next time.

The Incoming EMS processes in **MDB** are as follows:

1. An MDB that listens on the internal EMS incoming queue will receive the message.
2. The media details are identified and incoming message processing package in backend is called to process the message.
3. In case of any exception while processing, message will be sent to a deferred queue.
4. In case of messages directly arrive to JMS queue instead of a folder; the same MDB will be configured to listen on specific queue.

Index

S