

Scheduler JAVA JOB Creation  
Oracle FLEXCUBE Investor Servicing  
Release 14.7.5.0.0  
[June][2024]



---

# Table of Contents

1.	INTRODUCTION .....	1-3
2.	BACKGROUND.....	2-1
3.	EXAMPLE .....	3-1

---

## 1. Introduction

This document introduces the reader to the procedure to be followed for Scheduler JAVA JOB Creation for Oracle FLEXCUBE Investor Servicing.

---

## 2. Background

This document is based on the requirement of creating JAVA Jobs to be executed by Quartz Scheduler.

This document describes the procedure to be followed to accomplish the above.

## 3. Example

This section describes job creation:

1. Launch the job maintenance screen STDJOBMT for creation of a Job. User can maintain Interface, Reports and PL SQL Jobs.

### Interface Job definition

#### Event Based:

For Interface Job maintenance, user has to maintain `com.ofss.scheduler.quartz.fcis.FCISImpExpSchExecutor` class.

For Event Based Job Scheduling, EVENTCODE need to be maintained in Parameters. Valid Event Codes are available in Eventstbl.

The screenshot displays the 'Job Maintenance' window with the following sections:

- Job Description:** Includes fields for Job Code (marked with a red asterisk), Job Description, Job Group, Job Type (dropdown), Max Number Instances (marked with a red asterisk), Scheduler (SchedulerFactory), Trigger Type (dropdown), Scheduler Type (dropdown), Priority (dropdown), and Message Queue. A 'Validate' button is located below these fields.
- Job Details:** Includes fields for Cron Expression, Class Or Procedure, No of Submissions, Interval In Seconds, Trigger Listener, Active checkbox, Ds Name, Logging Required checkbox, Veto Blocked Trigger checkbox, and Startup Mode (dropdown).
- Parameter Details:** A table with columns for Parameter Name, Data Type, and Parameter Value. The table is currently empty.
- Footer:** Contains fields for Maker, Checker, Date Time, Mod No, Record Status, Authorization Status, and a 'Cancel' button.

## Time Based:

To maintain time based job, we have to maintain cron expression.

The screenshot shows the 'Job Maintenance' form with the 'Job Details' tab selected. The 'Job Description' section includes fields for Job Code, Job Description, Job Group, Job Type, Max Number Instances, Scheduler, SchedulerFactory, Trigger Type, Scheduler Type, Priority, and Message Queue. The 'Job Details' section includes Cron Expression, Class Or Procedure, No of Submissions, Interval In Seconds, Trigger Listener, Active checkbox, Ds Name, Logging Required checkbox, Veto Blocked Trigger checkbox, and Startup Mode. A 'Validate' button is located below the Message Queue field. The 'Parameter Details' section at the bottom shows a table with columns for Parameter Name, Data Type, and Parameter Value. The bottom status bar includes fields for Maker, Date Time, Mod No, Record Status, Authorization Status, and a 'Cancel' button.

## TimeZone Based

For Timezone based job scheduling, valid timezone need to be maintained in parameters. Timezone parameter should have standard Quartz supported timezone value.

This screenshot is identical to the one above, showing the 'Job Maintenance' form with the 'Job Details' tab selected. It displays the same fields for job description, details, and parameter details, including the 'Validate' button and the bottom status bar with 'Maker', 'Date Time', 'Mod No', 'Record Status', 'Authorization Status', and a 'Cancel' button.

## **REPORT Job definition**

To schedule BIP report generation, user has to maintain  
`com.ofss.scheduler.quartz.fcis.FCISReportSpollExecutor` class

The screenshot shows a 'Job Maintenance' window with a 'Save' button at the top left. The form is divided into two main sections: 'Job Description' and 'Job Details'.

**Job Description:**

- Job Code \***: A text input field.
- Job Description**: A text input field.
- Job Group**: A text input field.
- Job Type**: A dropdown menu.
- Max Number Instances \***: A text input field.
- Scheduler**: A dropdown menu with 'SchedulerFactory' selected.
- Trigger Type**: A dropdown menu.
- Scheduler Type**: A dropdown menu.
- Priority**: A dropdown menu.
- Message Queue**: A text input field.
- Validate**: A blue button.

**Job Details:**

- Cron Expression**: A text input field.
- Class Or Procedure**: A text input field.
- No of Submissions**: A text input field.
- Interval In Seconds**: A text input field.
- Trigger Listener**: A text input field.
- Active**: A checkbox.
- Ds Name**: A text input field.
- Logging Required**: A checkbox.
- Veto Blocked Trigger**: A checkbox.
- Startup Mode**: A dropdown menu.

**Parameter Details:**

At the bottom of the window, there is a 'Parameter Details' section with a table. The table has three columns: 'Parameter Name', 'Data Type', and 'Parameter Value'. The table is currently empty. Above the table, there are navigation controls: '< 1 Of 1 >' and a 'Go' button.

At the very bottom of the window, there is a footer area with the following fields:

- Maker**: A text input field.
- Checker**: A text input field.
- Date Time:**: A text input field.
- Mod No**: A text input field.
- Record Status**: A text input field.
- Authorization**: A text input field.
- Status**: A text input field.
- Cancel**: A blue button.

User has to maintain TASKID parameter to specify the Report Parameters. UTDTASK is the screen to maintain Report Parameters.

A Java job can be stateful or stateless.

### Stateful Job

If the job is configured to be STATEFUL then missed instances will be queued up so that scheduler would start executing once the long running job ends. It means, there will be no parallel execution of the java class.

To make a class as Stateful, please add two annotations to the job class.

- @PersistJobDataAfterExecution
- @DisallowConcurrentExecution

### Stateless Job

If the job is configured as STATELESS, threads can be run in parallel and can execute the same java class.

If the job is configured as stateless, the value of field **Max Number Instances** will denote the number of threads that can be executed in parallel .If the job is configured as stateful, then the above value will denote the number of missed instances that will be queued up so that Quartz Scheduler would start executing them once a long running job ends. This field specifies the number of such job instances that needs to be queued up.

The class defined in the class field has to be provided a fully qualified name.



Job params can be added to a particular java job in this screen as shown above.





Scheduler JAVA JOB Creation  
[June] [2024]  
Version 14.7.5.0.0

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  
[www.oracle.com/financialservices/](http://www.oracle.com/financialservices/)

Copyright © [2007], [2024], Oracle and/or its affiliates.

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.

