Scheduler JAVA JOB Creation

Oracle FLEXCUBE Universal Banking

Release 14.8.3.0.0

Part No. G61377-01

October 2026



Table of Contents

1. INTRODUCTION 1-2
2. BACKGROUND 2-1
3. PROCEDURE 3-1
4. EXAMPLE 4-1

1. Introduction

This document introduces the reader to the procedure to be followed for Scheduler JAVA JOB Creation for FLEXCUBE Universal Banking Solutions.

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. Procedure

This section describes job creation:

1. Launch the job maintenance screen STDJOBMT for creation of a Job.
2. Specify the field values. For information regarding individual fields please refer to Installation Docs.



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, 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.

4. Example

Suppose you have created a new FTP_COPY job, to copy ftp files from one location to another. It is a java job, where the class to be executed is FtpCopy.java.

You have provided the fully qualified class name as com.ofss.scheduler.quartz.ftp. FtpCopy.

The physical location of the java file is INFRA\FCJNeoWeb\Javasource\com\ofss\scheduler\quartz\ftp\FtpCopy.java.

In this class, you have to override the execute method of Job class to provide the implementation logic, which is executed when the quartz scheduler executes the class. The execute method format is as below.

public void execute (JobExecutionContext context) throws JobExecutionException {}

The sample java file is attached:



FtpCopy.java



Scheduler JAVA JOB Creation
October 2026
Version 14.8.3.0.0

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
https://www.oracle.com/industries/financial-services/index.html

Copyright © [2007], [2026], 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.