# Table of Contents

1. INTRODUCTION .................................................................................................................. 1-1
2. BACKGROUND ..................................................................................................................... 2-1
3. PROCEDURE .......................................................................................................................... 3-1
1. Introduction

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

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

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

Follow the below steps to create PLSQL job.

1. Launch the job maintenance screen STDJOBMT for creation of a PLSQL Job.

2. Please provide the values for the fields. For information regarding individual fields please refer to Installation Docs.

A PLSQL 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 procedure.

To make a executing 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 procedure.

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.

**Note the following:**

- For every PLSQL job, the executing class is FCPLSQLJobInvoker.java, which is configured as stateless.
- Job params can be added to a particular PLSQL job in this screen as shown above, which are passed to the procedure maintained.

**Example**

Suppose we have created a new ITSPNDARC PLSQL job for Spend Batch Archival. Here it is a PLSQL job, where the procedure to be executed is itpks_spend_batch.pr_spend_archival.
### Job Maintenance

#### Job Description
- **Job Code**: tspndarc
- **Job Description**: Spend Batch Archival
- **Job Group**: PLSQL
- **Job Type**: PI 3.3
- **Max Number Instances**: 1
- **Scheduler**: SchedulerFactory
- **Trigger Type**: Simple
- **Scheduler Type**: Guard
- **Priority**: Normal
- **Message Queue**: Validate

#### Job Details
- **Cron Expression**: `<job_name> <day-of-month> <month> <day-of-week> <hour> <minute>`
- **Class Or Procedure**: <job_name>
- **No of Submissions**: 1
- **Interval In Seconds**: 86400
- **Trigger Listener**: 
- **Ds Name**: Activa
- **Logging Required**: 
- **Veto Blocked Trigger**: 
- **Startup Mode**: Manual

#### Parameter Details

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>Varchar</td>
<td>OFSS</td>
</tr>
</tbody>
</table>

---

*ORACLE*