Oracle® Cloud Handle Throttling Using Parking Lot Pattern





Oracle Cloud Handle Throttling Using Parking Lot Pattern,

G18200-01

Copyright © 2024, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

Contents

Overview	1-3
System and Access Requirements	1-2
Recipe Schema	1-2
Before You Install the Recipe	
Access Your FTP Server	2-1
Configure Oracle ATP	2-2
Install and Configure the Recipe	
Configure the Oracle FTP Connection	3-2
Configure the Oracle ATP Connection	3-2
Configure the Lookup Table	3-3
Activate and Run the Recipe	



Preface

This document describes how to install, configure, and run this recipe in Oracle Integration 3.

Topics:

- Documentation Accessibility
- · Diversity and Inclusion
- Related Resources
- Conventions

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.

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.

Related Resources

For more information, see these Oracle resources:

- Oracle Integration documentation on the Oracle Help Center.
- Oracle Cloud at http://cloud.oracle.com.

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.



Convention	Meaning
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.



1

About This Recipe

Use this recipe to implement parking lot pattern in order to handle downstream throttling.



Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe demonstrates how to use the parking lot pattern for handling downstream throttling and errors with an example use case of importing employee bank details from an external system to a downstream payroll system. An Oracle ATP database table is used for implementing the parking lot pattern.

To use the recipe, you must install the recipe and configure the connections and other resources within the recipe. In this example, employee bank details are received from an external system such as Oracle HCM Cloud in a CSV file. The CSV file is then downloaded to a local directory on FTP Server or File Server. Subsequently, the employees bank details are registered in batches to an Oracle ATP database table and parked there for a specific time period before further processing. Thus, the integration flow has a chance to throttle the number of batches processed concurrently. The ready batches are picked up in a staged manner for processing and the processed data is finally updated to a downstream payroll system.

Overview of the recipe's integration flows:

- The Request Persister integration (SR_BulkDownload_RequestPersister_ATP) reads the CSV file received from external system and writes it to a local input directory in FTP. It reads the batch configurations (such as, group_id and group_type) and writes the metadata to the Oracle ATP database table. The registered batches are parked in the Parking Lot table in ATP for a specific period of time.
- The Scheduled Dispatcher integration (SR_ScheduledDispatcher_CSVBatch) is scheduled to run at required frequency. At every run, it fetches the batch requests from the Parking Lot table in ATP and dispatches them to the Async Batch Processor integration for processing. Thus, it triggers the Async Bach Processor integration to process the batches that are submitted for processing.
- The Async Processor integration (SR_OneWay_Processor_HCM_To_Payroll) processes all the submitted batch requests. Also, it updates the batch status (for example, if a batch is successfully processed, it updates the status to PROCESSED) in the Batch Statistics table in ATP. On successful processing, it invokes another integration (for example, SR_MOCK_EmpBankAccountProvider integration in this recipe) to send the processed information to a downstream application via REST API calls. Any errors received in the response from the downstream application are written into an Error folder in FTP. These payload errors are also written to a Payload Error Record table in ATP.
- The **Resubmission Processor** integration (**SR_ScheduledDispatcher_PayrollErrors**) is used to resubmit any batch requests failures/errors that were fixed. It fetches any record

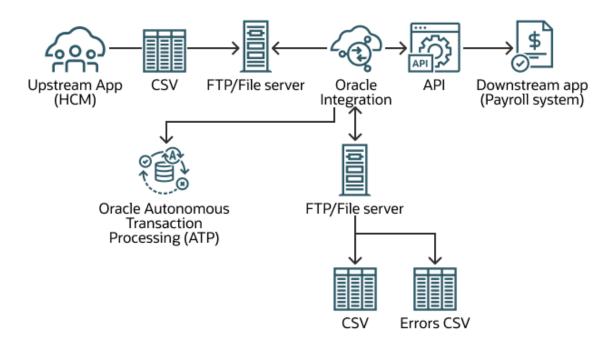
from the Payload Error Record table that are in READY state and re-submits them for processing.

System and Access Requirements

- Oracle Integration 3
- · A secure FTP (sFTP) server or File Server
- An FTP client to access the sFTP server
- Oracle ATP
- An account on Oracle ATP with the Administrator role

Recipe Schema

This section provides an architectural overview of the recipe.



A CSV file with employee bank details from an upstream application such as Oracle HCM Cloud is downloaded to an FTP Server or File Server. The first integration flow (**Request Persister**) in Oracle Integration registers the batches in the ATP database. Each batch request is parked in the ATP database parking lot table for a certain period of time so the integration flow has a chance to throttle the number of batches processed concurrently. The second scheduled integration flow (**Scheduled Dispatcher**) in Oracle Integration fetches the batches from the parking lot table in ATP at a date and time of your choice, and dispatches the batches for further processing. The second integration triggers the third integration flow (**Async Processor**) which processes the dispatched batches. The processed batches are then sent to a downstream application such as a Payroll system via a REST API call. Any errors that are received as response from the downstream application are rectified and then resubmitted for processing by the fourth integration flow (**Resubmission Processor**) in Oracle Integration.

Before You Install the Recipe

You must perform the following configuration tasks on your FTP Server and Oracle ATP instance in order to connect to these external systems using Oracle Integration and successfully handle throttling with the implementation of parking lot pattern.

Access Your FTP Server

Obtain a sFTP server and ensure that you're able to access it.

- 1. Log in to the FTP server using your user name and password through an FTP client; for example, FileZilla.
- Create the following directories:
 - An input directory to place the CSV file received from the upstream system. For example: HCM_To_Payroll_Input. This file should be in a particular format for the recipe to run successfully. Download the following sample input file to successfully test the recipe: Sample file.
 - A directory to place the files from the upstream system that were successfully processed. For example: HCM Archive
 - A directory to receive files from the downstream system that could not be processed successfully. For example: Payroll_To HCM_Input
 - A directory to archive files received from the downstream stream that were retried and processed successfully. For example: Payroll Archive
 - A local input directory. For example: Input
 - A local output directory. For example: Output
 - A local error directory. For example: Error
 - A local processing directory. For example: Processing

You can also use File Server, an embedded sFTP server within Oracle Integration to store and work with the files. To use File Server, you must enable it for your Oracle Integration instance. See Enable File Server in *Using File Server in Oracle Integration 3*.

Configure Oracle ATP

To configure Oracle ATP, perform the following configuration tasks:

- Complete the prerequisites for creating a connection from Oracle Integration to Oracle ATP. See Prerequisites for Creating a Connection.
- Download and run the following SQL script to create the required parking lot table: SQL Script.

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

- 1. On the Oracle Integration Home page, in the **Get started** section, click **Browse store**.
- 2. Find the recipe you want to install, then click **Get**.

A message confirms that the recipe was successfully installed, and the recipe card shows **In use**.

3. Click **Configure** on the recipe to configure its resources.

The project workspace opens, displaying all the resources of the recipe. Configure the following resources before you activate and run the recipe.

Note that you can delete and replace the recipe's REST Payroll API Connection with a REST connection created specifically to connect to the downstream application of your particular use case.

Configure the Oracle FTP Connection

- 1. In the Connections section, click the connection name.
- 2. In the Properties section, enter the following details:

Field	Information to Enter
FTP Server Host Name	Enter the host address of your sFTP server.
FTP Server Port	22
SFTP Connection	Select Yes from the list.

3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select FTP Server Access Policy.
Username	Enter the username to connect to your FTP server.
Password	Enter the password to connect to your FTP server.

- 4. Click Save. If prompted, click Save again.
- Click Test to ensure that your connection is successfully configured. In the resulting dialog, click Test again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back** <

Configure the Oracle ATP Connection

- 1. In the Connections section, click the connection name.
- 2. In the Properties section, specify the database service name in the **Service Name** field. See Configure Oracle ATP.
- 3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select JDBC With OCI Signature.
Wallet	Select the check box, then click Upload to upload the wallet file. See Configure Oracle ATP.
Wallet Password	Enter the wallet password.
Database Service Username	Enter the database service username to connect to the ATP database.
	The database service username is the schema username for the user to log in to the database. The database service username is not the same as the database service name that you specify in the connection Properties section.
Database Service Password	Enter the database service password to connect to the ATP database.
Object Storage Region	Specify the region in which your object storage is located. For example, us-ashburn-1.
Object Storage Tenancy OCID	Specify the value you copied from the Oracle Cloud Infrastructure Console. For example, ocid1.tenancy.oc1.alphanumeric.value
Object Storage Compartment OCID	Specify the value you copied from the Oracle Cloud Infrastructure Console. For example, ocid1.compartment.oc1.alphanumeric.value
Object Storage User OCID	Specify the value you copied from the Oracle Cloud Infrastructure Console. For example, ocid1.user.oc1.alphanumeric.value
Private Key	Click Upload to select the key you created. Ensure that the key is in RSA (PKCS1) format.
	Note: Only a private key without a pass phrase/ password is supported.
Finger Print	Enter the finger print that was generated when you created the key in the Oracle Cloud Infrastructure Console.

- 4. Click Save. If prompted, click Save again.
- Click Test to ensure that your connection is successfully configured. In the resulting dialog, click Test again.

A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**



Configure the Lookup Table

The recipe contains the following lookup tables that you can edit and configure as per your requirement.

- Email Notification
- Group_Setup
- 1. In the Lookups section, click the lookup name.
- Edit the following lookup tables.
 - a. In the Email_Notification lookup table, edit the value of the BatchNotificationTo lookup key by entering an email address of your choice to which notification error emails have to be sent.
 - In the Group_Setup lookup table, use the default directories provided for GRPID:
 HCM_To_Payroll or edit them as required with the directories that you created in your FTP server. See Access Your FTP Server.
 - InputDir: Enter the directory path to receive and store the input files from the upstream system (for example: /pob-Workshop/HCM_To_Payroll_Input).
 - ArchiveDir: Enter the directory path to archive files from the upstream system
 which were successfully processed (for example: /pob-Workshop/HCM Archive.
 - ProcessingHomeDir: Enter the path of the home directory (for example: / batchHome).
- 3. Click Save. If prompted, click Save again.
- 4. To return to the project workspace, click **Go back \leftilde{S}**.



4

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. In the project workspace, click **Activate**. In the Activate project panel, with the default project deployment selected, choose an appropriate tracing option, then click **Activate**.

A message confirms that the integrations have been activated. Refresh the page to view the updated status of the integrations.

2. Run the recipe.

Ensure that a sample input file with employee bank details in the correct format is uploaded to the input folder (in this case, HCM To Payroll Input).

- a. Run the SR_BulkDownload_RequestPersister_ATP integration flow.
 - i. In the Integrations section of the project workspace, click **Actions** • on the integration flow, then select **Run**.
 - ii. On the Configure and run page, click Run.

You've now successfully submitted the integration for execution. The integration now writes the input file to a local input folder (/Input). The information from the file, that is, the file metadata is persisted to the parking lot table (PAYLOAD_PARKING_LOT_TAB) in Oracle ATP.



You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

- **b.** Run the **SR_ScheduledDispatcher_CSVBatch** integration flow.
 - i. In the Integrations section of the project workspace, click **Actions** • on the integration flow, then select **Run**.
 - ii. On the Configure and run page, enter a value in the throttling parameter MaxRecords_fromDB to specify the number of records that should be fetched per run from the parking lot table (PAYLOAD_PARKING_LOT_TAB) in Oracle ATP.
 - iii. On the Configure and run page, click Run.

You've now successfully submitted the integration for execution. The integration now reads the data from the parking lot table, and dispatches the data to the asynchronous integration flow SR_OneWay_Processor_HCM_To_Payroll, thus triggering the integration flow to process the batch records fetched from the parking lot table (PAYLOAD_PARKING_LOT_TAB). In the parking lot table, the status of the batch records that were submitted for processing changes from NEW to PROCESSED.

Note:

You can also schedule this integration to run at a date, time, and frequency of your choosing. See Define the Integration Schedule.

The **SR_OneWay_Processor_HCM_To_Payroll** integration flow processes the batch files and calls the downstream application via a REST API call. It updates the batch statistics (BATCH_STATISTICS) table in ATP with the status of each record that is being processed.

The records that the downstream application could not process are written to the payload error table (PAYLOAD_ERRORS_TAB) in ATP with the status as ERRORED. When the errors are rectified, the status of the records changes to READY.

The **SR_ScheduledDispatcher_PayrollErrors** integration flow can be run to fetch the READY records from the payload error table (PAYLOAD_ERRORS_TAB), process them, and send them to the downstream application.

- 3. Monitor the running of the integration flows in Oracle Integration.
 - a. In the project workspace, click **Observe**. You'll see the integration flow being triggered and running successfully.
 - b. To manage errors in your project, see Manage Errors in a Project.

Related Documentation

- Using the FTP Adapter with Oracle Integration 3
- Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3

