

Oracle® Cloud

Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration Generation 2



F11444-35
June 2024



Oracle Cloud Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration Generation 2,
F11444-35

Copyright © 2019, 2024, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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 fail-safe, 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.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	v
Documentation Accessibility	v
Diversity and Inclusion	v
Related Resources	vi
Conventions	vi

1 Understand the Oracle Autonomous Transaction Processing Adapter

Oracle Autonomous Transaction Processing Adapter Capabilities	1-1
Cloud Database Connectivity Support	1-2
Oracle Autonomous Transaction Processing Adapter Restrictions	1-3
What Application Version Is Supported?	1-4
Workflow to Create and Add an Oracle Autonomous Transaction Processing Adapter Connection to an Integration	1-4

2 Create an Oracle Autonomous Transaction Processing Adapter Connection

Prerequisites for Creating a Connection	2-1
Download the Credentials Wallet and Obtain Database Details	2-1
Satisfy Bulk Data Import Operation Prerequisites	2-1
Migrate from an Oracle Database Classic Cloud Service Database Instance to an Oracle Autonomous Transaction Processing or Oracle Autonomous Data Warehouse Database Instance	2-3
Create a Connection	2-4
Configure Connection Properties	2-5
Configure Connection Security	2-6
Configure an Agent Group	2-7
Test the Connection	2-8

3	Add the Oracle Autonomous Transaction Processing Adapter Connection to an Integration	
		<hr/>
	Basic Information Page	3-1
	Invoke Stored Procedure Page	3-2
	Invoke SQL Statement Page	3-4
	Table Operation Page	3-5
	Import Tables Page	3-5
	Relationships Page	3-5
	Create Relationship Page	3-6
	Attribute Filtering Page	3-6
	Advanced Options Page	3-6
	Operations on Table Page	3-7
	Invoke Bulk Load from Object Storage Page	3-9
	Summary Page	3-11
4	Implement Common Patterns Using the Oracle Autonomous Transaction Processing Adapter	
		<hr/>
	Methods for Connecting Oracle Integration to an Oracle Autonomous Transaction Processing - Dedicated Database Instance with the Oracle Autonomous Transaction Processing Adapter	4-1
	Define a Select Operation on Database Tables	4-1
	Upload Data from a Flat File into a Database Table	4-4
5	Troubleshoot the Oracle Autonomous Transaction Processing Adapter	
		<hr/>
	Service Name Specified on the Connections Page is Too Long	5-1
	Special Characters are Not Supported in Schema Names	5-1
	Oracle Autonomous Transaction Processing Database is Unreachable During a Connection Test	5-2

Preface

This guide describes how to configure this adapter as a connection in an integration in Oracle Integration.



Note:

The use of this adapter may differ depending on the features you have, or whether your instance was provisioned using Standard or Enterprise edition. These differences are noted throughout this guide.

Topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Related Resources](#)
- [Conventions](#)

Audience

This guide is intended for developers who want to use this adapter in integrations in Oracle Integration.

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

See these Oracle resources:

- Oracle Cloud
<http://cloud.oracle.com>
- *Using Integrations in Oracle Integration Generation 2*
- *Using the Oracle Mapper with Oracle Integration Generation 2*

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.
<i>italic</i>	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

Understand the Oracle Autonomous Transaction Processing Adapter

Review the following conceptual topics to learn about the Oracle Autonomous Transaction Processing Adapter and how to use it as a connection in integrations in Oracle Integration. A typical workflow of adapter and integration tasks is also provided.

Topics:

- [Oracle Autonomous Transaction Processing Adapter Capabilities](#)
- [Oracle Autonomous Transaction Processing Adapter Restrictions](#)
- [What Application Version Is Supported?](#)
- [Workflow to Create and Add an Oracle Autonomous Transaction Processing Adapter Connection to an Integration](#)

Oracle Autonomous Transaction Processing Adapter Capabilities

The Oracle Autonomous Transaction Processing Adapter enables you to integrate the Oracle Autonomous Transaction Processing database with Oracle Integration through use of a wallet for direct connectivity. Use the Oracle Autonomous Transaction Processing Adapter to execute SQL queries or stored procedures in the Oracle Autonomous Transaction Processing database. For example, quotes in Oracle CPQ Cloud can be created as `Orders` in the Oracle Autonomous Transaction Processing database by executing SQL statements or stored procedures using the Oracle Autonomous Transaction Processing Adapter.

The Oracle Autonomous Transaction Processing Adapter provides the following capabilities:

- Support for using a wallet for direct connectivity to connect to the Oracle Autonomous Transaction Processing - Shared (ATP-S) database in place of using the on-premises connectivity agent.

WARNING:

For direct connections (configured without using the connectivity agent), the Oracle Autonomous Transaction Processing Adapter can only be used for making outbound invocations as an invoke connection. This type of Oracle Autonomous Transaction Processing Adapter-specific connection cannot be configured as a trigger connection in an integration. Use cases related to inbound polling must be implemented using the scheduled orchestration integration pattern.

- Support for accessing an Oracle Autonomous Transaction Processing – Dedicated (ATP-D) database with a wallet-based connection that uses the connectivity agent.
- Support for integrating an Autonomous Transaction Processing - Shared (ATP-S) database with a private endpoint. Integration is achieved with a wallet-based connection that uses the connectivity agent. See [Configure Connection Security](#).

- Support for the bulk data import operation. This feature is supported for the following scenarios:
 - When using direct connectivity (that is, without use of the connectivity agent)
 - When using the Oracle Autonomous Transaction Processing - Shared (ATP-S) database
 - With the comma-separated value (CSV) data format only.

A typical bulk data import use case with the Oracle Autonomous Transaction Processing Adapter consists of getting a data file from an FTP server or application (such as an HCM payroll system), placing the file in an object storage bucket, and placing the file contents into an Oracle Autonomous Transaction Processing - Shared (ATP-S) database. The file can be in a format expected by the database (for example, a CSV file) or in a format that the database is not expecting (for example, XML or JSON). In that case, Oracle Integration is used to translate the format into a CSV format that the database expects.

- Support for creating integrations with Oracle Autonomous Data Warehouse.
- Support for invocation of stored procedures in the Oracle Autonomous Database.
- Support for non-JDBC (PL/SQL record and PL/SQL table) datatypes in outbound invocations of stored procedures.
- Support for execution of DML statements and SQL queries: *Select*, *Insert*, *Update*, and *Delete*.

Select the **Run a SQL Statement** option on the Basic Info page of the Adapter Endpoint Configuration Wizard to execute simple SQL queries. For complex SQL queries, use stored procedures by selecting the **Invoke a Stored Procedure** option on the Basic Info page of the Adapter Endpoint Configuration Wizard. Stored procedures can reduce the complexity of a SQL query.

- Support for updating or inserting multiple records in a single request.
- Support for performing a `SELECT` operation against database tables.
- Support for the operations on a table feature, which enables you to model SQL statements with the Adapter Endpoint Configuration Wizard. The operation on a table feature also supports multiple records in a single request.

**Note:**

In Java, Unicode characters are represented as 2 bytes.

Oracle Autonomous Transaction Processing delivers a self-driving, self-securing, self-repairing database service that can instantly scale to meet demands of mission critical transaction processing and mixed workload applications. See [Autonomous Transaction Processing](#).

The Oracle Autonomous Transaction Processing Adapter is one of many predefined adapters included with Oracle Integration. You can configure the Oracle Autonomous Transaction Processing Adapter as a connection in an integration in Oracle Integration.

Cloud Database Connectivity Support

This table describes cloud database connectivity support in Oracle Integration. Review these capabilities and use the adapter (Oracle Autonomous Transaction Processing Adapter, Oracle Autonomous Data Warehouse Adapter, or Oracle Database Cloud Service Adapter) appropriate to your environment.

Database	Direct Connectivity with Wallet	Connectivity Agent with Username Token Policy	Connectivity Agent with Wallet
Autonomous Database - Shared (Oracle Autonomous Transaction Processing - Shared, Oracle Autonomous Data Warehouse - Shared)	Supported	Not supported	Supported
Private Endpoint for Autonomous Database - Shared (Oracle Autonomous Transaction Processing - Shared Private Endpoint, Oracle Autonomous Data Warehouse - Shared Private Endpoint)	Not supported	Not supported because Oracle Autonomous Transaction Processing - Shared supports only wallets	Supported
Autonomous Database - Dedicated (Oracle Autonomous Transaction Processing - Dedicated, Oracle Autonomous Data Warehouse - Dedicated)	Not supported	Supported	Supported
Autonomous Database - Dedicated (Oracle Autonomous Transaction Processing - Dedicated Private Endpoint, Oracle Autonomous Data Warehouse - Dedicated Private Endpoint)	Not supported	Supported	Supported
Oracle Database Cloud Service	Supported	Supported	Supported
Oracle Cloud Infrastructure - Bare Metal and Virtual Machine Database Systems (Oracle Database Cloud Service Private Endpoint)	Not supported	Supported	Supported

See [Overview of Autonomous Databases](#) and [Bare Metal and Virtual Machine DB Systems](#).

Oracle Autonomous Transaction Processing Adapter Restrictions

Note the following Oracle Autonomous Transaction Processing Adapter restrictions in Oracle Integration.

- Using the bulk data import operation with the connectivity agent is not supported.
- The database password length cannot exceed 20 characters.
- Database schema names with hyphens (-) are not supported.
- All integrations that include stored procedure, PureSQL, or operation on table database operations must finish within 240 seconds. Otherwise, the query times out and a `Limit Exceeded` error occurs.
- Access to object storage is a requirement for using the bulk data import operation. Oracle Integration for SaaS consists of a subscription-based account dedicated to Oracle Integration, and does not include services such as object storage. You must obtain a separate trial account to provision object storage for use with the Oracle Autonomous Transaction Processing Adapter.
- For direct connections (configured without using the connectivity agent), the Oracle Autonomous Transaction Processing Adapter can only be used for making outbound invocations as an invoke connection. This type of Oracle Autonomous Transaction Processing Adapter-specific connection cannot be configured as a trigger connection in an integration. Use cases related to inbound polling must be implemented using the scheduled orchestration integration pattern.
- The PL/SQL boolean type is not supported as an IN/OUT parameter in a stored procedure. However, you can create a wrapper stored procedure that converts PL/SQL boolean to an integer and use those wrapper stored procedures in Oracle Integration.
- The Oracle Autonomous Transaction Processing Adapter uses JDBC drivers to interact with the database and is restricted by JDBC driver constraints. Therefore, nested PL/SQL types (for example, `RECORD` types inside a `TABLE` type) are not supported as IN/OUT parameters in a stored procedure. However, you can define `OBJECT` types inside the `TABLE` type.
- Cross schema stored procedures are not allowed in cases where Oracle Integration must generate the wrappers.

 **Note:**

There are overall service limits with Oracle Integration. A service limit is the quota or allowance set on a resource. See [Service Limits](#).

What Application Version Is Supported?

For information about which application version is supported by this adapter, see the [Connectivity Certification Matrix](#).

See [Connectivity Certification Matrix](#).

Workflow to Create and Add an Oracle Autonomous Transaction Processing Adapter Connection to an Integration

You follow a very simple workflow to create a connection with an adapter and include the connection in an integration.

Step	Description	More Information
1	Create the adapter connections for the applications you want to integrate. The connections can be reused in multiple integrations and are typically created by the administrator.	Create an Oracle Autonomous Transaction Processing Adapter Connection
2	Create the integration. When you do this, you add trigger and invoke connections to the integration.	Create Integrations and Add the Oracle Autonomous Transaction Processing Adapter Connection to an Integration
3	Map data between the trigger connection data structure and the invoke connection data structure.	Map Data in <i>Using Integrations in Oracle Integration Generation 2</i>
4	(Optional) Create lookups that map the different values used by those applications to identify the same type of object (such as gender codes or country codes).	Manage Lookups in <i>Using Integrations in Oracle Integration Generation 2</i>
5	Activate the integration.	Manage Integrations in <i>Using Integrations in Oracle Integration Generation 2</i>
6	Monitor the integration on the dashboard.	Monitor Integrations in <i>Using Integrations in Oracle Integration Generation 2</i>
7	Track payload fields in messages during runtime.	Assign Business Identifiers for Tracking Fields in Messages and Manage Business Identifiers for Tracking Fields in Messages in <i>Using Integrations in Oracle Integration Generation 2</i>
8	Manage errors at the integration level, connection level, or specific integration instance level.	Manage Errors in <i>Using Integrations in Oracle Integration Generation 2</i>

2

Create an Oracle Autonomous Transaction Processing Adapter Connection

A connection is based on an adapter. You define connections to the specific cloud applications that you want to integrate.

Topics:

- [Prerequisites for Creating a Connection](#)
- [Create a Connection](#)

Prerequisites for Creating a Connection

Satisfy the following prerequisites appropriate to your environment for creating a connection with Oracle Integration.

- [Download the Credentials Wallet and Obtain Database Details](#)
- [Satisfy Bulk Data Import Operation Prerequisites](#)
- [Migrate from an Oracle Database Classic Cloud Service Database Instance to an Oracle Autonomous Transaction Processing or Oracle Autonomous Data Warehouse Database Instance](#)

Download the Credentials Wallet and Obtain Database Details

You must download the credentials wallet and obtain database details.

1. Download the client credentials wallet from the Oracle Autonomous Transaction Processing instance. See [Download Client Credentials \(Wallet\)](#) of *Getting Started with Oracle Autonomous Database for Transaction Processing and Mixed Workloads*.
2. Ensure that the target database is accessible.
3. Ensure that you have write permissions to the database.
4. Ensure that you have the required permissions to run stored procedures and SQL statements.
5. Know the database service name.
6. Know the database service username and password for connecting to the database.

Satisfy Bulk Data Import Operation Prerequisites

If you want to use the bulk data import operation, you must configure the Oracle Autonomous Transaction Processing Adapter to use the JDBC With OCI Signature security policy on the Connections page. Before you can successfully configure this security policy on the Connections page, you must complete the following prerequisites.


1. Create an object storage bucket in Oracle Cloud Infrastructure and obtain the OCID value of the compartment in which you create the storage bucket. You need this value when configuring the security policy on the Connections page.

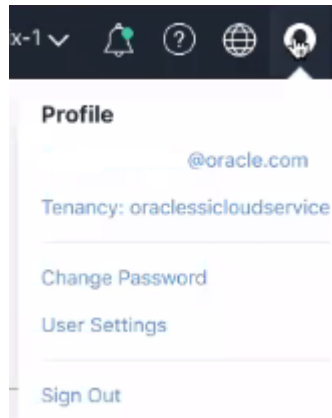
You specify the OCID of the object storage user on the Connections page when configuring the JDBC With OCI Signature security policy. This user should have the `manage bucket` and `manage object` privileges. Set the following policies to manage privileges on the bucket and object in the object storage:

```
Allow group group_name to manage buckets in compartment compartment_name
```

```
Allow group group_name to manage objects in compartment compartment_name
```

See [How Policies Work](#) and [Policy Reference](#).

2. Obtain the tenancy OCID and user OCID details in the Oracle Cloud Infrastructure Console. You need these values when configuring the security policy on the Connections page. When you sign up for Oracle Cloud Infrastructure, Oracle creates a tenancy for your company, which is a secure and isolated partition within Oracle Cloud Infrastructure where you can create, organize, and administer your cloud resources.
 - a. Sign in to the Oracle Cloud Infrastructure Console.
 - b. In the upper left corner, select , then select **Home > Governance & Administration > Tenancy Details**.
 - c. In the **Tenancy Information** section, click **Show** to display the **OCID** tenancy value.
 - d. Copy the value. You need this value when configuring the security policy on the Connections page.
 - e. In the upper right corner, click the **Profile** icon and select **User Settings**.




 **Note:**

You can also select



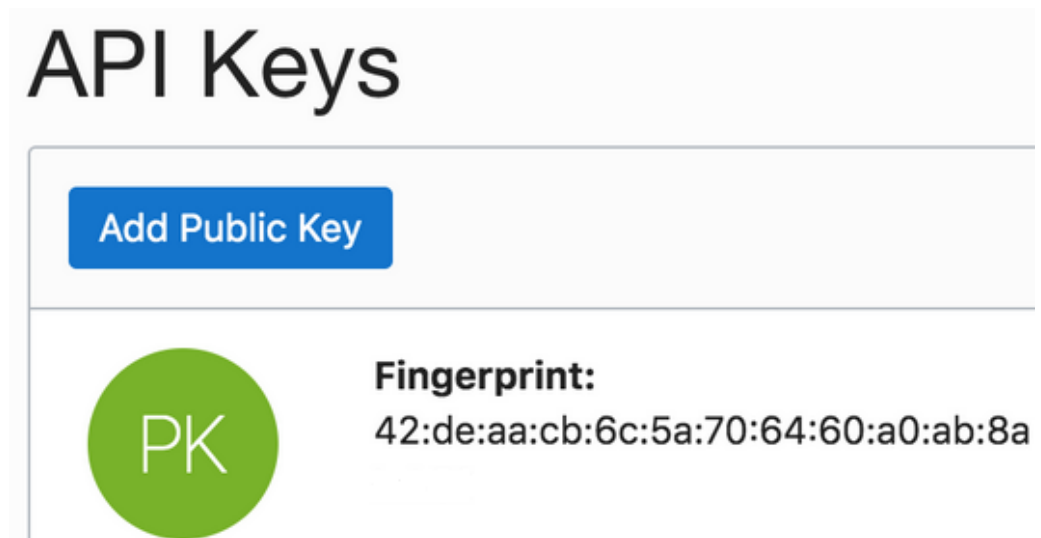
in the upper left corner, then select **Identity > Users** to access the user profile.

- f. Click **Show** to display the **OCID** user value.
- g. Copy the value.
3. Create an API signing key. You then specify the signing key in Oracle Cloud Infrastructure.
 - a. Sign in to the Oracle Cloud Infrastructure Console to upload the public key.
 - b. In the upper left corner, select , then select **Identity > Users**.
 - c. On the Users page, click the link of the user name to use.
 - d. Scroll down to the **API Keys** section, and click **Add Public Key**.

 **Note:**

Only a private key *without* a pass phrase/password is supported.

- e. In the Add Public Key dialog, enter the contents of the public key you created, and click **Add**.
- f. Copy the finger print value generated by Oracle Cloud Infrastructure. You need this value when configuring the JDBC With OCI Signature security policy on the Connections page.



Migrate from an Oracle Database Classic Cloud Service Database Instance to an Oracle Autonomous Transaction Processing or Oracle Autonomous Data Warehouse Database Instance

Perform the following steps if you want to migrate from an Oracle Database Classic Cloud Service database instance to an Oracle Autonomous Transaction Processing or Oracle Autonomous Data Warehouse database instance.

1. Migrate all the required database objects, stored procedures, wrapper procedures, and tables to the destination Oracle Autonomous Transaction Processing or Oracle Autonomous Data Warehouse database instance.
2. Change the Oracle Database Cloud Service Adapter connection details to point to an Oracle Autonomous Transaction Processing or Oracle Autonomous Data Warehouse database instance.
 - a. Go to the Connection page for the Oracle Database Cloud Service Adapter.
 - b. Click **Configure Connectivity**.
 - c. Specify the new host name.
 - d. Specify the new service name and click **OK**.
 - e. Click **Configure Security**.
 - f. Select the **Oracle Wallet** security policy.
 - g. Upload the wallet.
 - h. Specify the wallet password and reconfirm it.
 - i. Specify the database service username.
 - j. Specify the database service password, reconfirm it, and click **OK**.
 - k. Delete the agent from the connection.
3. Test the connection.
4. Once the test is successful, click **Save** to save the connection details.
5. Reactivate the integrations.

Create a Connection

Before you can build an integration, you have to create the connections to the applications with which you want to share data.


To create a connection in Oracle Integration:

1. In the left navigation pane, click **Home > Integrations > Connections**.
2. Click **Create**.

Note:

You can also create a connection in the integration canvas of:

- An orchestrated integration (See Define Inbound Triggers and Outbound Invokes.)
- A basic routing integration (See Add a Trigger (Source) Connection.)

3. In the Create Connection — Select Adapter dialog, select the adapter to use for this connection. To find the adapter, scroll through the list, or enter a partial or full name in the **Search** field and click  **Search**.
4. In the Create Connection dialog, enter the information that describes this connection.

- a. Enter a meaningful name to help others find your connection when they begin to create their own integrations. The name you enter is automatically added in capital letters to the **Identifier** field. If you modify the identifier name, don't include blank spaces (for example, SALES OPPORTUNITY).
 - b. Enter optional keywords (tags). You can search on the connection keywords on the Connections page.
 - c. Select the role (direction) in which to use this connection (trigger, invoke, or both). Only the roles supported by the adapter are displayed for selection. When you select a role, only the connection properties and security policies appropriate to that role are displayed on the Connections page. If you select an adapter that supports both invoke and trigger, but select only one of those roles, you'll get an error when you try to drag the adapter into the section you didn't select. For example, let's say you configure a connection for the Oracle Service Cloud (RightNow) Adapter as only an **invoke**. Dragging the adapter to a **trigger** section in the integration produces an error.
 - d. Enter an optional description of the connection.
5. Click **Create**.

Your connection is created. You're now ready to configure the connection details, such as connection properties, security policies, connection login credentials, and (for certain connections) agent group.

Configure Connection Properties

Enter connection information so your application can process requests.

1. Go to the **Connection Properties** section.
2. If configuring the connection to use direct connectivity (that is, you are *not* using the connectivity agent):
 - a. In the **Host** field, specify the host.
 - b. In the **Service Name** field, specify the database service name. It is recommended that you use the `low` profile as the database service name. The database service name is *not* the same as the database service username that you specify in the **Security** section.

 **Note:**

The service name must be the same as the one in the `tnsnames.ora` file in the wallet. However, if you receive an error when testing the connection, it may be because the name is too long. See [Service Name Specified on the Connections Page is Too Long](#).

3. If configuring the connection to use the connectivity agent:
 - a. In the **Host** field, specify the host.
 - b. In the **Port** field, specify the SQL*Net port.
 - c. In the **SID** field, specify the database SID.
 - d. In the **Service Name** field, specify the database service name. It is recommended that you use the `low` profile as the database service name. The database service name is *not* the same as the database service username that you specify in the **Security** section.

 **Note:**

You must specify either a SID or service name value. Do *not* specify values for both fields.

Configure Connection Security

Configure security for your database connection by selecting the security policy and setting login credentials. A database connection is only allowed for publicly accessible databases.

1. Go to the **Security** section.
2. Select the security policy.
3. If you select **JDBC Basic Authentication**:

 **Note:**

Using the Username Password Token security policy, you can create and successfully test the connection to the database without the connectivity agent if the database is present in the same private network as the Oracle Integration instance. If the database is not in the same private network, it is not recommended that you connect without the connectivity agent because the connection won't be secure.

- a. Enter the database service username and password to connect to the Oracle Autonomous Transaction Processing 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.
 - b. Re-enter the password a second time.
4. If you select **JDBC Over SSL**:

 **Note:**

The Oracle Autonomous Transaction Processing Adapter can connect through the connectivity agent when using the wallet. It can be used as a trigger connection only if the connectivity agent is used in the connection. However, all operations that you select on the Basic Info page such as **Run a SQL Statement**, **Invoke a Stored Procedure**, and **Perform an Operation On a table** are supported when configuring the adapter to use direct connectivity (without the connectivity agent).

- a. In the **Wallet** field, select the check box, then click **Upload** to upload the wallet file.
- b. Enter the wallet password, then re-enter it a second time to confirm.
- c. Enter the database service username and password to connect to the Oracle Autonomous Transaction Processing 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.

- d. Enter the database password a second time to confirm.
5. If you select **JDBC With OCI Signature**:

 **Note:**

This option is required to use the bulk data import operation and is only supported when using direct connectivity (that is, you are *not* using the connectivity agent).

- a. In the **Wallet** field, select the check box, then click **Upload** to upload the wallet file.
- b. Enter the wallet password, then re-enter it a second time to confirm.
- c. Enter the database service username and password to connect to the Oracle Autonomous Transaction Processing - Shared (ATP-S) 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.
- d. Enter the database password a second time to confirm.
- e. In the **Object Storage Region** field, specify the region in which your object storage is located (for example, `us-ashburn-1`).
- f. In the **Object Storage Tenancy OCID** field, specify the value you copied from the Oracle Cloud Infrastructure Console (for example, `ocid1.tenancy.oc1.alphanumeric.value`). See [Prerequisites for Creating a Connection](#).
- g. In the **Object Storage Compartment OCID** field, specify the value you copied from the Oracle Cloud Infrastructure Console (for example, `ocid1.compartment.oc1.alphanumeric.value`).
- h. In the **Object Storage User OCID** field, specify the value you copied from the Oracle Cloud Infrastructure Console (for example, `ocid1.user.oc1.alphanumeric.value`).
- i. In the **Private Key** field, 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.

- j. In the **Finger Print** field, enter the finger print that was generated when you created the key in the Oracle Cloud Infrastructure Console.

Configure an Agent Group

Configure an agent group for accessing the service hosted on your premises behind the fire wall.

1. Click **Configure Agents**.

The Select an Agent Group page appears.

2. Click the name of the agent group.
3. Click **Use**.

To configure an agent group, you must download and install the on-premises connectivity agent. See *Download and Run the Connectivity Agent Installer and About Connectivity Agents and Integrations Between On-Premises Applications and Oracle Integration in Using Integrations in Oracle Integration Generation 2*.

Test the Connection

Test your connection to ensure that it's configured successfully.

1. In the page title bar, click **Test**. What happens next depends on whether your connection uses a Web Services Description Language (WSDL) file.

If Your Connection...	Then...
Doesn't use a WSDL	The test starts automatically and validates the inputs you provided for the connection.
Uses a WSDL	A dialog prompts you to select the type of connection testing to perform: <ul style="list-style-type: none"> • Validate and Test: Performs a full validation of the WSDL, including processing of the imported schemas and WSDLs. Complete validation can take several minutes depending on the number of imported schemas and WSDLs. No requests are sent to the operations exposed in the WSDL. • Test: Connects to the WSDL URL and performs a syntax check on the WSDL. No requests are sent to the operations exposed in the WSDL.

2. Wait for a message about the results of the connection test.
 - If the test was successful, then the connection is configured properly.
 - If the test failed, then edit the configuration details you entered. Check for typos, verify URLs and credentials, and download the diagnostic logs for additional details. Continue to test until the connection is successful.
 - If using the JDBC With OCI Signature security policy, compartment ID verification doesn't occur during the connection test. The Adapter Endpoint Configuration Wizard handles verification and authorization issues with the compartment OCID.
3. When complete, click **Save**.

3

Add the Oracle Autonomous Transaction Processing Adapter Connection to an Integration

When you drag the Oracle Autonomous Transaction Processing Adapter into an integration, the Adapter Endpoint Configuration Wizard appears. This wizard guides you through configuration of Oracle Autonomous Transaction Processing Adapter endpoint properties.

The following sections describe the wizard pages that guide you through configuration of the Oracle Autonomous Transaction Processing Adapter as an invoke in an integration. The Oracle Autonomous Transaction Processing Adapter cannot be configured as a trigger connection in an integration.

Topics:

- [Basic Information Page](#)
- [Invoke Stored Procedure Page](#)
- [Invoke SQL Statement Page](#)
- [Table Operation Page](#)
- [Operations on Table Page](#)
- [Invoke Bulk Load from Object Storage Page](#)
- [Summary Page](#)

Basic Information Page

Specify a name, description, and operation type on the Basic Info page of each trigger and invoke connection in your integration.

Element	Description
What do you want to call your endpoint?	Identifies the connection with a meaningful name that defines the purpose of connection. For example, <code>CreateEmployeeInDB</code> for a database connection that adds new employee data. The name can include English alphabetic characters, numbers, underscores, and dashes. The name cannot include: <ul style="list-style-type: none">• Blank spaces (for example, <code>My DB Connection</code>)• Special characters (for example, <code>#;83&</code> or <code>right)now4</code>)• Multibyte characters

Element	Description
What operation do you want to perform?	<ul style="list-style-type: none"> • Invoke a Stored Procedure — Select to run a stored procedure on the database. • Run a SQL Statement — Select to run a SQL query on the database. • Perform an Operation On a Table — Select to perform one of the following operations on a table. You can update or insert multiple records in a single request. <ul style="list-style-type: none"> – Insert – Update – Insert or Update (Merge) – Select • Perform Bulk Data Import Operation — Select to perform one of the following bulk data import operations on a table. This option is only displayed if you configured the JDBC With OCI Signature security policy and are using direct connectivity (that is, the connection is <i>not</i> configured to use the connectivity agent). <ul style="list-style-type: none"> – Perform Bulk Load: Loads a file into the object storage bucket and puts it into the specified database table. – Fetch Load Status: Fetches the status of a bulk data import operation (either completed, failed, or in progress).
	<p>Notes</p> <ul style="list-style-type: none"> • When operations in a SQL statement such as Update, Concat, and Merge accept values for the inbound invocation of an integration, they do not work. For example, the following query does not work:
	<pre>select concat(empname, 'ss') from DB_AQ where empno=#empno</pre>
	<pre>select empno from DB_AQ where empname=concat(#empname, 'YY')</pre>
	<p>As a workaround, handle these scenarios during payload mapping. For example, perform a concatenation during mapping of the payload. The final output can then be passed as input to the SQL query.</p>
	<ul style="list-style-type: none"> • IN/BETWEEN operators are not supported with bind parameters. Use greater than (>) and less than (<) operators instead.

Invoke Stored Procedure Page

Enter the invoke stored procedure values. The Invoke a Stored Procedure page appears when you select **Invoke a Stored Procedure** as the operation to perform on the Basic Info page.

You can specify the following values on the Invoke a Stored Procedure page.

- Select the database schema that includes the data you want to query (for example, you want to query details about an employee based on their employee ID).

- Select a stored procedure or package from the list that is displayed after you select the database schema.

 **Note:**

- Stored procedures return binary large objects (for example, BLOB database data types) as base64Binary types in XML. Depending upon the use cases, these can be decoded during transformation using inbuilt functions such as decodeBase64 or can be passed as-is for downstream processing.
- Adapter input/output parameters are defined based on the stored procedure IN/OUT parameters. The IN parameter corresponds to the request and the OUT parameter is translated as the response. Procedures without parameters are not listed in the Adapter Endpoint Configuration Wizard for database versions 18c and above. You can pass a dummy parameter or create a wrapper procedure with a dummy parameter to list it in the wizard.
- Stored procedures that contain Boolean or %rowtype as IN/OUT parameters are not supported. You can create wrappers on top of these stored procedures to use them.

Element	Description
Select Schema	Select a database schema from the list. This action refreshes the page to display fields for selecting a package or procedure to invoke. Special characters (for example, #) are not supported in schema names. See Special Characters are Not Supported in Schema Names .
Select Package	Select the database package. This action refreshes the page to display the procedures available for the package. When importing a predefined integration package containing PLS or SQL stored procedures, the wrapper package is not recreated in the target database. To add the wrapper package, confirm JPublisher is installed on the target database and define the original stored procedure. After confirming JPublisher is installed and the stored procedure is defined, open the PL/SQL Wrapper utility and execute the add scripts command to add the scripts included in the exported inventory archives (IAR) file.
Select Procedure	Displays the in (inbound), out (outbound), and in/out (inbound/outbound) parameters for the selected package.
Arguments	Display the in, out, and in/out parameters that are passed with this procedure.

Invoke SQL Statement Page

Enter the SQL statement values. The Run a SQL Statement page appears when **Run a SQL Statement** is selected as the operation to perform on the Basic Info page. You can specify the following values on the Run a SQL Statement page.

Note:

- Do not use schema/database names in SQL queries. Configure the details in the connection. For example:

```
Update HR.employee set HR.employee.first_name = 'Name' where
HR.employee.employee_id='1'
```

can be changed to a simple query, such as:

```
Update employee set first_name = 'Name' where employee_id='1'
```

where `HR` is used in the connection details. This restricts a user with specific privileges to a particular schema/database.

- When configuring the adapter as an invoke connection, ensure that proper spaces are provided between key words for a pure SQL statement. For example, the following statement fails during integration activation because there is no blank space between `VALUES` and `(#`.

```
INSERT INTO table_name VALUES(#EMPNO, #EMPNAME)
```

Add a blank space between `VALUES` and `(#`, and the statement is successfully processed.

```
INSERT INTO table_name VALUES (#EMPNO, #EMPNAME)
```

- When configuring the adapter as an invoke connection, define all bind parameters in the same order and define the parameters that takes absolute values at the end.

```
INSERT INTO table_name (EMPNO, EMPNAME, EMPUUID, EMPPHONE,
EMPHIREDATE) VALUES (#EMPNO,
#EMPNAME, Sys_guid(), NULL, SYSDATE)
```

Element	Description
SQL Query	Identifies the SQL query.
Validate SQL Query	Validates the SQL query syntax.
Status	Displays the SQL query syntax validation status. When syntax validation is successful, the message <code>Success!</code> appears.

Table Operation Page

You can update or insert multiple records in a single request.

 **Note:**

When you change the structure of a table (for example, you add or delete a column), you must re-import the table by doing a re-edit in the Adapter Endpoint Configuration Wizard. Go to the Import Tables page and re-import the same table, then click **OK > Next > Done** to complete the wizard. Only then are the table changes reflected in the integration.

Topics:

- [Relationships Page](#)
- [Create Relationship Page](#)
- [Attribute Filtering Page](#)
- [Operations on Table Page](#)

Import Tables Page

Filter and select the tables to import based on the selected schema. These tables are used to generate a SQL statement based on the operation selected.

You can import the following number of tables:

- A maximum of three tables for insert, update, and insert or update actions
- A maximum of five tables for the select - operation on table feature
- A maximum of five tables for the polling feature

Element	Description
Schema	Select the schema to use. The page is refreshed to display the tables available for selection.
Name Filter	Filter the display of tables.
Available	Select the tables on which to insert or update records.
Selected	Displays the selected tables.

Relationships Page

Review the relationships between the selected tables and optionally create, remove, or rename relationships. These relationships are used in the insert or update SQL statements.

Element	Description
Relationships Table	Displays the relationships defined on the root database table and any related tables (one-to-one or one-to-many).
Create	Click to create new relationships.

Element	Description
Remove	Click to remove a selected relationship.
Rename	Click to rename a selected relationship.

Create Relationship Page

Specify the parent and child relationships to use in the SQL statement.

Element	Description
Parent Table	Select the parent table.
Child Table	Select the child table.
Mapping Type	Select the mapping type (one-to-many, one-to-one, or one-to-one with the foreign key on the child table). For example, if you selected Employees as the parent table and Departments as the child table, the following options are displayed: <ul style="list-style-type: none"> • Employees has a 1:1 Relationship with Departments • Employees has a 1:1 Relationship with Departments (Foreign Key on Child table) • Employees has a 1:M Relationship with Departments
Parent and Child Table	Associate the foreign key fields to the primary key fields.
Relationship Name	Optionally name the relationship (a default name is generated).

Attribute Filtering Page

Filter out the attributes to exclude.

Element	Description
Attributes Tree	Deselect any attributes to exclude from the database query. You cannot exclude primary key attributes.

Advanced Options Page

Provide additional advanced options such as sequencing. This is only valid for the insert and merge operations.

Element	Description
Table	Displays the selected table.
Sequence	Specify that the primary key is assigned from a sequence on any insert. Click Search and select a sequence from the list. Only the sequences of the user who owns the adapter on the Connections page are listed. The adapter generates sequence numbers in a batch of 50. Configure sequences in increments of 50. This issue only applies to the Oracle Database Adapter.

Operations on Table Page

Select the database tables. To use the bulk extract feature, you must choose the **SELECT** operation from the **Perform an Operation On a Table** list on the Basic Info page.

Operations on Table Page

Element	Description
Schema	Select the database schema that includes the tables to process.
Table Name	Enter a filter with which to search the schema (for example, %TAB to search for tables with TAB in the name).
Table Type	Specify the table type filter to get a subset of the appropriate database objects, then click Search . <ul style="list-style-type: none"> • ALL • TABLE • VIEW
Filter By	Enter the initial letters to filter the display of table names.
Table Names	Select the tables to import. Note: It is recommended that you to import the tables together for the adapter to automatically recognize the relationship. If you import the tables separately, you must explicitly create the table relationship.
Import Tables	Click to import the tables. The page is refreshed for you to select the parent database table.
Select the parent database table	Select the parent (root) table from the list. If using multiple related tables, this is the top-level parent table in the relationship. After making your selection, the page is refreshed for you to view and edit the table relationships.
Add Remove Tables	Click to add more tables or remove tables no longer in use.
Review and manage parent database table relationships	Click Edit to view and edit the table relationships. The relationships automatically identified by the adapter are displayed. See Review and manage parent database table relationships Option .
Review and filter columns from selected database tables	Click Edit to view and edit the table attributes. You can deselect any attributes to exclude from the database queries. Primary key attributes cannot be excluded. See Review and filter columns from selected database tables Option .
Review and edit SQL query	Click Edit to view and edit the default SQL query. See Review and edit SQL query Option . Note: This field is available for a <code>Select</code> operation on the table.

Review and manage parent database table relationships Option

Table 3-1 - Review and manage parent database table relationships Option

Element	Description
Create New	Click to create a new relationship.

Table 3-1 (Cont.) - Review and manage parent database table relationships Option

Element	Description
Relations	View the existing parent and child table relations automatically created by the adapter.

Review and filter columns from selected database tables Option**Table 3-2 - Review and filter columns from selected database tables Option**

Element	Description
Attributes Tree	View and deselect attributes automatically created by the adapter.

Review and edit SQL query Option**Note:**

This is only applicable for a `Select` operation on a table.

Table 3-3 - Review and edit SQL query Option

Element	Description
SQL Edit	Click to manually edit the query in the SQL Query field.

Table 3-3 (Cont.) - Review and edit SQL query Option

Element	Description
Edit using Expression Builder	<p>Click to edit the query in the Expression Builder.</p> <ul style="list-style-type: none"> • Add New: Click to add new criteria to the SQL query. <ol style="list-style-type: none"> 1. Click Add New. 2. In the First Argument field, click Edit, and select the argument to add (for example, deptno). 3. In the Operator field, select the operator to use for the comparison from the dropdown list (for example, =). 4. In the Second Argument field, select the option to use: <ul style="list-style-type: none"> – Literal: Click to specify a value. If selected, you are prompted to select the data type (for example, integer) and specify the value. – Parameter: Click to specify a bind parameter. – Query Key: Click to run the comparison against another column in the table. <p>New criteria is appended to the SQL query with a WHERE clause. If you add subsequent SQL queries, they are appended to the SQL query with an AND clause</p> • Add Nested: Click to add nested criteria to the SQL query. • Edit: Click the edit the SQL criteria you specified. • Remove: Click the edit the SQL criteria you specified. <p>Click to edit the query with the Expression Builder.</p>
Maximum Number of Records to be fetched	<p>Select the number of records to fetch with this SQL query.</p>

Invoke Bulk Load from Object Storage Page

Specify the details for performing a bulk data import operation.

Element	Description
Select Bucket	<p>Select the object storage bucket in which to place or delete the file. The object storage buckets available for selection are located in the compartment that you specified on the Connections page when creating the connection.</p> <p>Note: If you don't have the privilege to access the buckets, a <code>User is Not authorized</code> message is shown on the screen and no other options are listed.</p>

Element	Description
Delete file from object storage after operation completion	Select to delete the file from the object storage bucket. Otherwise, the file remains in the bucket.
Review and specify the copy_data format options	<p>Click Edit to <i>optionally</i> specify the file format options. If not specified here or sent in the mapper, all default options are used. The following options are displayed by default.</p> <ul style="list-style-type: none"> • Delimiter: Select the field delimiter used in the file. <ul style="list-style-type: none"> – Pipe – Single space – Comma – Semicolon – Tab – Other (Select to display a field for entering a custom delimiter.) • Record Delimiter: Select the record delimiter used in the file. A new line value is the default value. • Skip Headers: Specify the number of rows to skip at the top of the file (if any). • Show Advanced Options: Click to optionally specify advanced formatting options for the file. If no formatting options are specified, the default settings are used. If you want to specify an option that is not available in the wizard, you can send the format options as a JSON string in the mapper as follows. <pre style="margin-left: 20px;">'{"delimiter" : "\" \"", "quote" : "\"^\"", "ignoremissingcolumns" : "true", "dateformat" : "YYYY-MM-DD-HH24-MI-SS", "blankasnull" : "true"}'</pre> <p>Note: For special character usage in the JSON string, when the format options are sent in the mapper, the options provided in the Adapter Endpoint Configuration Wizard are overridden. In this case, the format options sent in the mapper are only considered. It is recommended that you select only one way of providing the format options (either from the wizard or through the mapper). If both are provided, the options in the mapper are only considered.</p> <p>When complete, click OK to return to the Bulk Load from Object Storage page.</p>
Select Schema	Select the database table schema.
Select Table	Select the database table in which to write the file data. Based on your selection, all the fields in that table are displayed for selection.

Element	Description
Table Columns	<p>Displays the columns (fields) in the table.</p> <p>Note: When selecting the fields in the table, select all the fields in the order of the incoming data. If none of the fields are selected, a default value is considered that takes into account that the data fields are in the same order as the table fields. Table column names with special characters such as spaces or a # are not supported.</p> <p>For example, assume the CSV file was formatted as follows:</p> <pre>booker12 123 Rachel Johnson gray07 234 Laura Johnson1 johnson81 345 Craig Johnson2 jenkins46 456 Mary Johnson</pre> <p>As an example, you may select to move the following table columns to the Re-order columns as per input data list to match the order of the CSV file:</p> <ul style="list-style-type: none">• USERNAME• IDENTIFIER• FIRSTNAME• LASTNAME

Summary Page

You can review the specified adapter configuration values on the Summary page.

Element	Description
Summary	<p>Displays a summary of the configuration values you defined on previous pages of the wizard.</p> <p>The information that is displayed can vary by adapter. For some adapters, the selected business objects and operation name are displayed. For adapters for which a generated XSD file is provided, click the XSD link to view a read-only version of the file.</p> <p>To return to a previous page to update any values, click the appropriate tab in the left panel or click Back.</p> <p>To cancel your configuration details, click Cancel.</p>

4

Implement Common Patterns Using the Oracle Autonomous Transaction Processing Adapter

You can use the Oracle Autonomous Transaction Processing Adapter to implement the following common patterns.

Topics:

- [Methods for Connecting Oracle Integration to an Oracle Autonomous Transaction Processing - Dedicated Database Instance with the Oracle Autonomous Transaction Processing Adapter](#)
- [Define a Select Operation on Database Tables](#)
- [Upload Data from a Flat File into a Database Table](#)

Methods for Connecting Oracle Integration to an Oracle Autonomous Transaction Processing - Dedicated Database Instance with the Oracle Autonomous Transaction Processing Adapter

Connectivity from Oracle Integration to an Oracle Autonomous Transaction Processing-Dedicated database instance using the Oracle Autonomous Transaction Processing Adapter can be accomplished through the following methods.

- **Connectivity agent:** Use this approach when the Oracle Autonomous Transaction Processing - Dedicated database instance is not publicly accessible and resides within a private subnet in your Virtual Cloud Network (VCN). With this topology, the Oracle Autonomous Transaction Processing Adapter cannot directly access the Oracle Autonomous Transaction Processing- Dedicated database instance. The connectivity agent must be deployed in your network to allow the connectivity agent access to the Oracle Autonomous Transaction Processing- Dedicated database instance.
- **Direct connectivity:** Use this approach when the Oracle Autonomous Transaction Processing - Dedicated database instance is directly accessible from Oracle Integration. In this case, the connectivity agent is not necessary. However, Oracle strongly recommends using Transport Level Security (TLS) for the communication between Oracle Integration and the Oracle Autonomous Transaction Processing- Dedicated database instance to ensure the data being transmitted is encrypted in motion.

Define a Select Operation on Database Tables

You can define a `SELECT` operation to perform against database tables. This section provides a high level overview of creating an integration in which an Oracle Autonomous Transaction

Processing Adapter is configured as an invoke connection to retrieve table records from the Oracle Database.

To define a `SELECT` operation on database tables:

1. Configure SOAP Adapter and Oracle Autonomous Transaction Processing Adapter connections.
2. Select **App Driven Orchestration** in the Create Integration - Select a Style dialog.
3. Add and configure the SOAP Adapter as a trigger connection in the integration.

The SOAP Adapter is configured to accept an input and return the response received from the invoke connection.

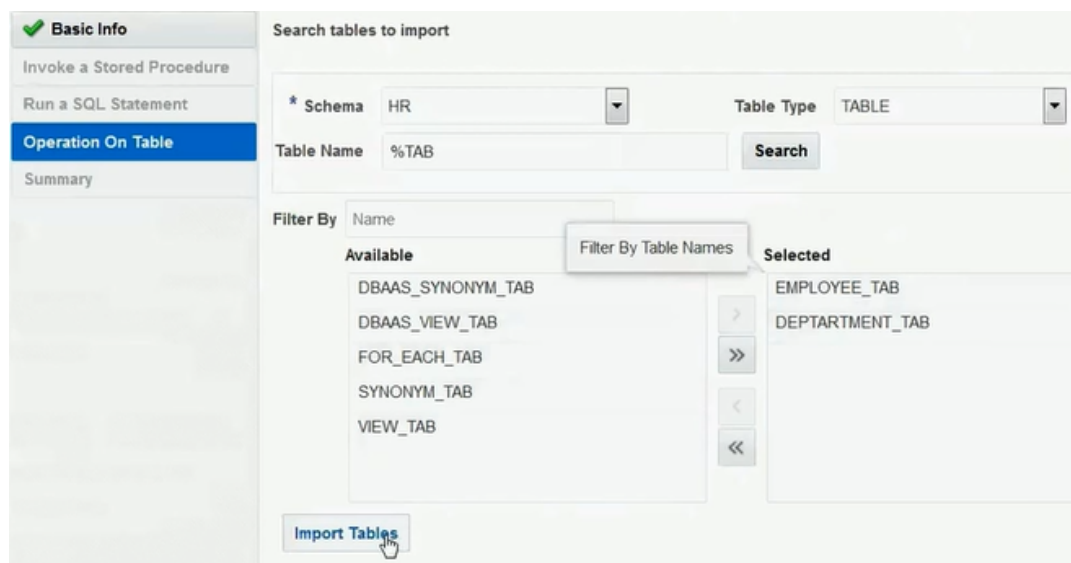
4. Add the Oracle Autonomous Transaction Processing Adapter as an invoke connection in the integration.

This invokes the Adapter Endpoint Configuration Wizard.

5. On the Basic Info page, select **Perform an Operation On a Table** as the type of operation to perform and **Select** as the operation to perform on the table.

6. On the Operate On Table page, specify the schema and tables to import, and click **Import Tables**. For this example, the following values are specified.

- **Schema:** HR
- **Table Type:** TABLE
- **Table Name:** %TAB
- **Selected Tables:** **EMPLOYEE_TAB** and **DEPARTMENT_TAB**. The tables are imported together for the Oracle Autonomous Transaction Processing Adapter to recognize the relation between the tables.



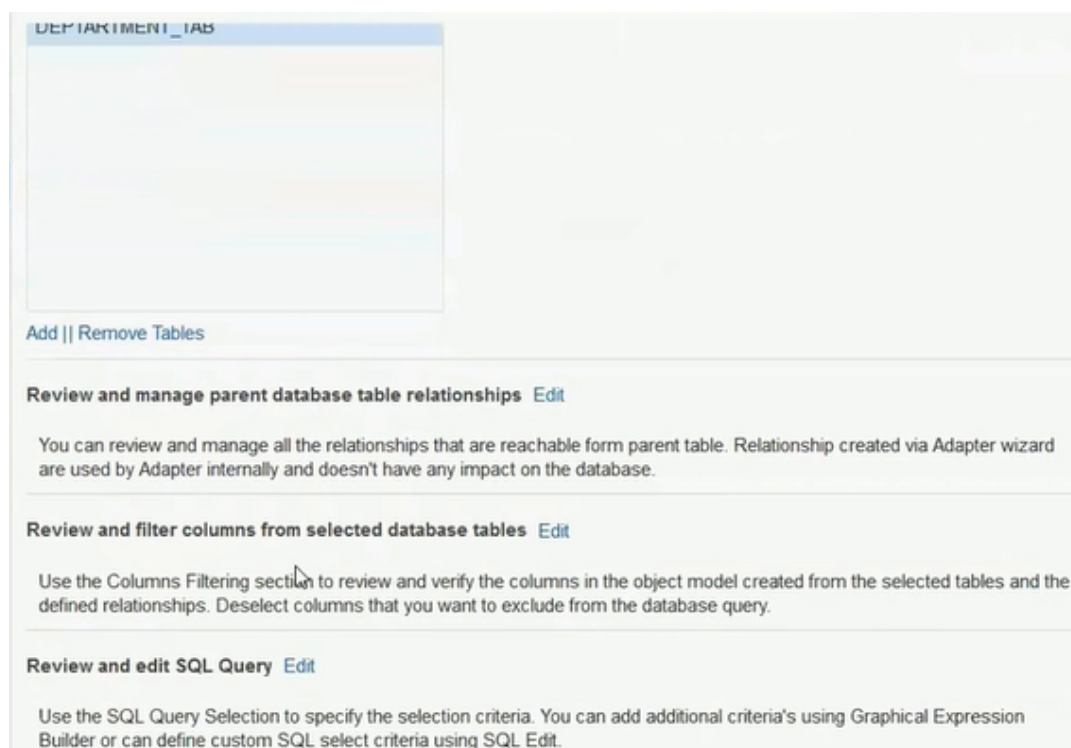
The page is refreshed for you to select the parent (root) database table.

7. Select the parent table (for this example, **DEPARTMENT_TAB** is selected).

This page enables you to:

- View the automatically created table relationships and create new ones.
- View and deselect attributes.

- View and edit the automatically created SQL query.



8. If you want to edit the automatically created SQL query, click **Edit** to the right of **Review and edit SQL Query**.
 - a. Click **Edit using Expression Builder**. You can also manually edit the SQL query by clicking **SQL Edit**.
 - b. Click **Add New** to add new criteria to the SQL query. The automatically created SQL query is displayed below the link.



- c. Specify values for the following fields, and click **OK**.
 - **First Argument**
 - **Operator**
 - **Second Argument**

For example:

The criteria you specify are appended to the existing SQL query as part of a **WHERE** clause. Any additional SQL query criteria you specify are appended as part of an **AND** clause. For example:

```
SELECT DISTINCT t0.DEPTNO, t0.DEPTNAME, t0.LOC FROM DEPARTMENT_TAB t0, EMPLOYEE_TAB t1
WHERE (((t0.DEPTNO = #deptno) AND (t1.EMPID > 999)) AND (t1.DEPTNO = t0.DEPTNO))
```

- d. Click **OK**.
9. Click **Next**.
10. View your selections on the Summary page. Links to the tables you selected to import and SQL query you specified are provided.

11. Click **Done** to exit the Adapter Endpoint Configuration Wizard.
12. Complete the integration by performing mapping and tracking tasks.
13. Activate the integration.
14. Copy the link to invoke the integration from under the **How to Run** link.
15. Invoke the integration from a tool such as the SOAP UI.
16. Review the values returned by the Oracle Autonomous Transaction Processing Adapter.

Upload Data from a Flat File into a Database Table

The Oracle Autonomous Transaction Processing Adapter can read rows of data in chunks from a comma-separated value (CSV) or XML file and insert them into a table in the Oracle Autonomous Transaction Processing database. With support for cloud libraries in the Oracle Autonomous Transaction Processing database, this operation can be completed much faster. This section provides details about how to use the Oracle Autonomous Transaction Processing

Adapter to insert data from a large, flat CSV file into an Oracle Autonomous Transaction Processing database.

This use case is first demonstrated with a standard integration that:

- Retrieves a CSV file from the embedded File Server included in Oracle Integration.
- Uses the Oracle Autonomous Transaction Processing Adapter to process the file in chunks and perform a standard insert operation into an Oracle Autonomous Transaction Processing database table.

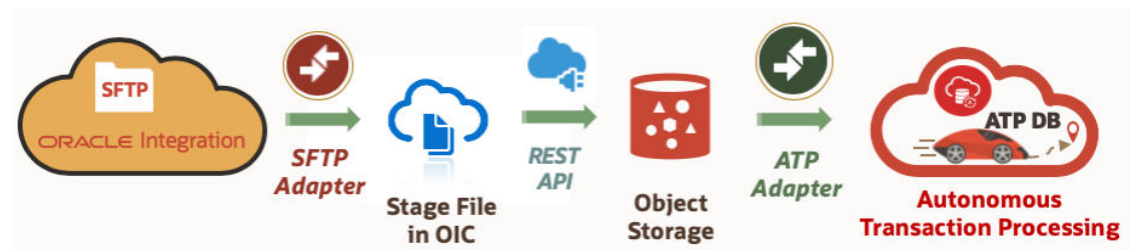
This standard integration is shown in [Figure 1 - Standard Integration](#).



To demonstrate the enhanced integration use case:

- The same file is read from the same SFTP server location and uploaded to an object storage location within Oracle Cloud Infrastructure through a REST Adapter invoke connection.
- The Oracle Autonomous Transaction Processing Adapter invokes the COPY_DATA PL/SQL procedure from the DBMS_CLOUD standard package library, which reads the file from the object storage location and inserts the data into the target table in the Oracle Autonomous Transaction Processing database. PL/SQL package library details are available in the Oracle Autonomous Transaction Processing documentation.

This enhanced integration is shown in [Figure 2 - Enhanced Integration](#).



The configuration process is described in two sections:

- The standard integration that is built uses basic components.
- The enhanced integration that is built uses PL/SQL libraries in the Oracle Autonomous Transaction Processing database.

Standard Integration

An overview of the steps to build the standard integration is provided below.

1. Configure the File Server to enable SFTP server functionality. See Administer File Server in *Using File Server in Oracle Integration Generation 2*.

2. Create an FTP Adapter connection to retrieve the file from the File Server.
3. Create an Oracle Autonomous Transaction Processing Adapter connection to point to the target Oracle Autonomous Transaction Processing database.
4. Build the integration shown in [Figure 1 - Standard Integration](#).
5. In the Adapter Endpoint Configuration Wizard, select the insert operation with the target table in the Oracle Autonomous Transaction Processing database.
When complete, the integration looks as follows:



Enhanced Integration

An overview of the steps to build the enhanced integration is provided below.

1. Use the same SFTP connection built in Step 2 of the previous section.
2. Use the same Oracle Autonomous Transaction Processing Adapter connection from Step 3 of the previous section.
3. Create a REST Adapter connection to upload a CSV file to an Oracle Cloud Infrastructure object storage location.
4. Create PL/SQL credentials in the database to access the object store using the `DBMS_CLOUD.CREATE_CREDENTIAL` procedure.
5. Create a PL/SQL wrapper package and procedure to call `DBMS_CLOUD.COPY_DATA` using the credentials from Step 4 and the target table in the Oracle Autonomous Transaction Processing database to insert the data.
6. Build the enhanced integration shown in [Figure 2 - Enhanced Integration](#).
7. In the Adapter Endpoint Configuration Wizard, instead of selecting the insert operation, select the PL/SQL procedure option.
8. Select the wrapper procedure created in Step 5.
9. When complete, the integration looks as follows.



XML files can also be processed by `DBMS_CLOUD`. See this [blog](#).

5

Troubleshoot the Oracle Autonomous Transaction Processing Adapter

Review the following topics to learn about troubleshooting issues with the Oracle Autonomous Transaction Processing Adapter.

Topics:

- [Service Name Specified on the Connections Page is Too Long](#)
- [Special Characters are Not Supported in Schema Names](#)
- [Oracle Autonomous Transaction Processing Database is Unreachable During a Connection Test](#)

Additional integration troubleshooting information is provided. See [Troubleshoot Oracle Integration in *Using Integrations in Oracle Integration Generation 2*](#) and the [Oracle Integration Troubleshooting page](#) in the Oracle Help Center.

Service Name Specified on the Connections Page is Too Long

The service name that you specify in the **Service Name** field on the Connections page is provided in the `tnsnames.ora` file included in the wallet ZIP file downloaded from the Oracle Autonomous Transaction Processing database. If the service name is too long, you receive the following error when testing the connection:

```
Unable to test connection "MYATP". [Cause: CLOUD-0005]
CLOUD-0005 : Unable to establish connection. Please check connection
parameters
Network Adapter could not establish the connection. Please check the wallet
credentials
and ensure database is reachable.
```

Perform the following steps to resolve this error.

1. Ensure that the correct service name is provided.
2. Pick the name from the name-value pair. The `tnsnames.ora` file includes a list of service names.
3. Ensure that the database is reachable. Try connecting from any other SQL client to confirm.
4. Ensure that the correct wallet credentials are provided.

Special Characters are Not Supported in Schema Names

If you use schema names with special characters such as #, integration activation fails. For stored procedures, the schema derives the names of the types in the XSD. If the type name contains #, the XSD has problems with the name. Use a schema name that does not contain any special characters.

Oracle Autonomous Transaction Processing Database is Unreachable During a Connection Test

If you receive the following error during a connection test and are using the connectivity agent to connect to an Oracle Autonomous Transaction Processing database running on a private subnet, you must verify that the Oracle Autonomous Transaction Processing database IP address can be resolved from the agent host.

Network Adapter could not establish the connection. Please check the wallet credentials and ensure database is reachable

1. From the Oracle Autonomous Transaction Processing wallet file, extract the `tnsnames.ora` file.
2. Find the service name being used (for example, `test_atp_low`) and retrieve the host name (for example, `test_atp.adb.us-phoenix-1.oraclecloud.com`).
3. From the agent host, run the following command:

```
nslookup ATP_hostname
```

4. If the hostname does not resolve to the Oracle Autonomous Transaction Processing database IP address, create an entry in the `/etc/hosts` file on the agent host:

```
ATP_IP_Address ATP_Service_Hostname_from_tnsnames.ora Hostname
```

For example:

```
10.0.0.64 test_atp.adb.us-phoenix-1.oraclecloud.com test_atp
```

5. Test the connection again.