# Oracle® Cloud

Configuring a Disaster Recovery Solution for Oracle Integration

February, 2021, Version 1.0

## Purpose statement

This document describes how to configure a disaster recovery solution for Oracle Integration. It is intended for personnel who are responsible for configuring a disaster recovery solution for Oracle Integration.

## Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

## Table of contents

**List of images**

## Introduction

Oracle Integration is highly available in an Oracle Cloud Infrastructure (OCI) region governed by service-level agreements (SLAs). This white paper details the procedure to build a cross-region, customer-managed disaster recovery solution for Oracle Integration, specifically for the Integrations component in Oracle Integration.

### Disaster Recovery Concepts

A disaster recovery (DR) solution enables you to recover quickly from natural or man-made disasters and continue to provide services to your users. In addition, you can use the DR set up for planned migrations and switch between different regions periodically.

Before you set up the DR solution, you must determine the recovery point objective (RPO) and recovery time objective (RTO) for your service. The RTO is the target time within which your service must be restored after a disaster occurs. The RPO is the period after a disaster occurs for which the service can tolerate lost data before the disaster begins to affect the business.

The DR solution described in this white paper replicates only the design-time artifacts; therefore, the RPO is not applicable in this scenario. You can measure the RTO based on how quickly you're able to complete the failover tasks in the secondary region.

> **Note:** A comprehensive, Oracle-managed DR solution for Oracle Integration is planned for the future.

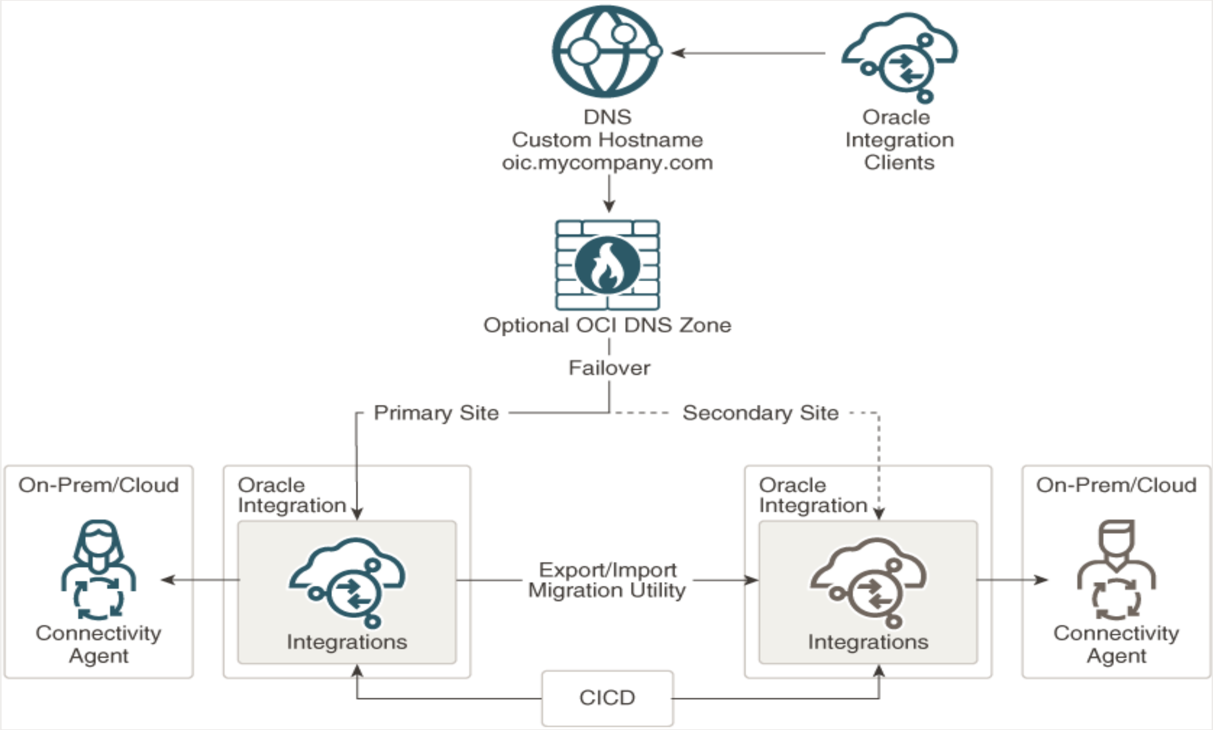
## Disaster Recovery Architecture

You can configure a high-availability DR solution only for the Integrations component in Oracle Integration.

The DR architecture for Integrations consists of two Oracle Integration instances in two different cloud regions, which are accessed using a single custom endpoint (URL). Optionally, you can use an OCI Domain Name System (DNS) zone as a front-end to route traffic to the instances.

The two Oracle Integration instances in the architecture are designated as primary and secondary, and both the instances run concurrently; however, only one of the instances receives traffic. Initially, it's the primary instance that receives the traffic flow. When this instance becomes unavailable, the DNS record is updated to route the traffic to the secondary instance. The following image shows this architecture in detail:

Image 1. The disaster recovery architecture for the Integrations component of Oracle Integration.



You must update the DNS record at your DNS provider to switch between instances. Optionally, you can implement an OCI DNS zone to manage the sub-domain related to the Oracle Integration custom hostname. The OCI DNS zone can reflect changes to the CNAME or A-record much faster.

In this setup, you must synchronize the Oracle Integration metadata at both the sites using continuous integration and continuous deployment (CICD).

## About the Solution Scope

This sections details the scope and constraints of the DR solution.

### What's Supported?

- Active-Passive topology only.
- Integrations (replication of only the design-time artifacts is supported).
- Publish-subscribe integration patterns and integrations with scheduled parameters or polling triggers require special handling. See Additional Key Considerations.
- Visual Builder (requires special considerations; detailed instructions coming soon).

### What's Not Supported?

- Processes.
- Insight.
- File Server.
- B2B (may require special considerations).

### Additional Key Considerations

- You must frequently update integrations with scheduled parameters in the secondary Oracle Integration instance. See Automate Scheduled Parameters Updates.
- You must provision and configure a connectivity agent for both primary and secondary Oracle Integration instances.
- When the primary instance is active, you must stop or deactivate scheduled flows, integrations with polling triggers, and integrations subscribing to business events from SaaS applications in the secondary instance. After a failover, activate these artifacts in the secondary instance.
- Note that, after a failover, the integrations with scheduled parameters may not be current in the secondary instance. It depends on how often you update them.
- The monitoring data (message history) is not replicated. You can use APIs to extract the message history in your data lake or warehouse to have a global view.
- Ensure that your source applications and end users use the custom endpoint to access the Oracle Integration instance. Your administrators must use the respective original instance hostnames to manage the instances.

### Failover Management

When a failover to the secondary instance is required, you must manually change the routing by updating the DNS record at your DNS provider.

Here are some additional points to note about the failover process:

- It's customer initiated.
- It's quick and simple for non-scheduled and non-polling integration flows.
- Scheduled flows and integrations with polling triggers require special handling.

## Prerequisites to Set Up the Solution

Ensure that all the prerequisites for the DR solution are met before you begin the configuration process.

Before beginning the set up, you must:

- Provision another Oracle Integration instance in a different OCI region with at least one message pack. However, to ensure that the secondary instance handles the usual volume, provision it with the same number of message packs as your primary instance.
- Provision an OCI Object Storage bucket for metadata migration.

## Set Up the Disaster Recovery Solution for Oracle Integration

You'll perform a few one-time tasks to configure the DR system. After the system is set up, you'll regularly synchronize metadata between your instances, frequently update integrations with scheduled parameters in the secondary Oracle Integration instance, monitor the instances for a failover, and execute failover-handling tasks when necessary.

### Perform One-Time Configuration Tasks

After you've procured the necessary Oracle services, you can begin setting up the DR system. As part of the configuration, you can also set up an OCI DNS zone if necessary.

1. Create two Oracle Integration instances in two different OCI regions; for example, one in Ashburn and another in Phoenix.
2. Configure a common custom endpoint for both primary and secondary Oracle Integration instances. See Configure a Custom Endpoint.
3. Optionally, create an OCI DNS zone to manage the sub-domain related to the Oracle Integration custom hostname. See Use an OCI DNS Management Zone.
4. Provision a dedicated connectivity agent for each Oracle Integration instance and use the original instance hostnames in the agents' configurations.
5. If you have integration flows that subscribe to business events from Oracle Fusion Applications, configure credential keys. See Configure the Oracle Fusion Application Credential Keys.
6. Optionally, use REST APIs to migrate the metadata from the primary to secondary instance for the first time. See Migrate Metadata from the Primary Instance.
7. Verify your system end-to-end after configuring the DR environment. Access the custom endpoint and navigate through the Oracle Integration console.

### Configure a Custom Endpoint

Configure a common custom endpoint to your Oracle Integration instances, so that applications and users can access Oracle Integration with the same URL regardless of which instance is active in the background.

To configure a custom endpoint for your instances:

1. Choose a custom hostname for your instances and register it with a DNS provider.
2. Obtain an SSL certificate from a certificate authority (CA) for your hostname.
3. See Configure a Custom Endpoint for an Instance for the full list of tasks.

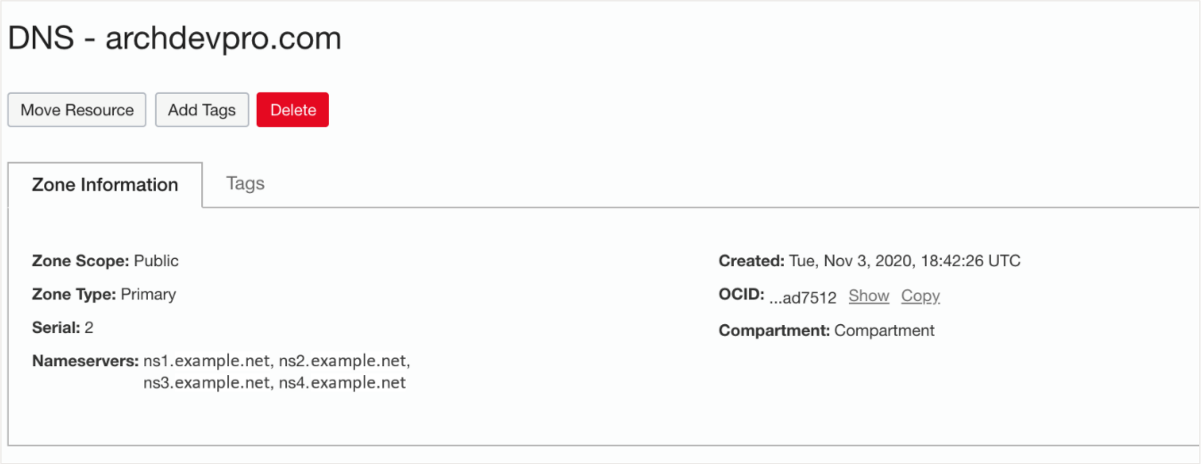
   **Note:** If you use a hostname certificate whose certificate authority (CA) is not in the Oracle Integration trust store, you must also upload the certificate to your Oracle Integration instance.

## Use an OCI DNS Management Zone

You can use an OCI DNS zone to manage DNS records and provide hostname resolution for your Oracle Integration instances.

1. After you've acquired a domain (or a sub-domain) for your Oracle Integration instance, add an OCI DNS zone through the OCI console or the API. See Managing DNS Service Zones for details on creating an OCI DNS zone and adding a record to it.
   The following image shows an example DNS zone created for the domain named *archdevpro.com*.

   Image 2. An OCI DNS zone named *archdevpro.com* added using the OCI console.



DNS - archdevpro.com

**Zone Scope:** Public

**Zone Type:** Primary

**Serial:** 2

**Nameservers:** ns1.example.net, ns2.example.net, ns3.example.net, ns4.example.net

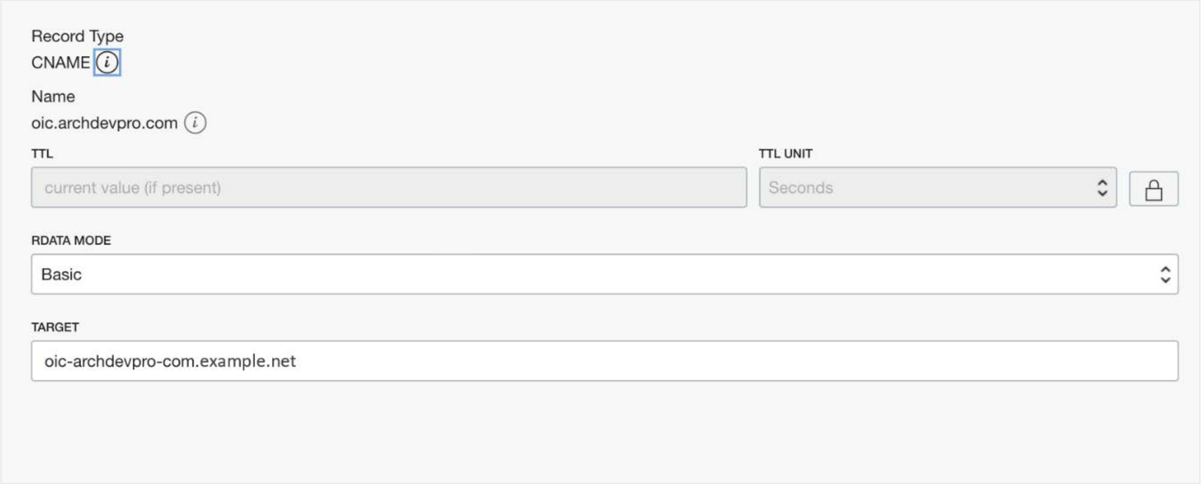
**Created:** Tue, Nov 3, 2020, 18:42:26 UTC

**OCID:** ...ad7512

**Compartment:** Compartment

2. In the zone, add the Oracle Integration custom hostname as a CNAME record.

   Image 3. The Oracle Integration hostname added as a CNAME record within an OCI DNS zone.



Record Type
CNAME

Name
oic.archdevpro.com

TTL: current value (if present)

TTL UNIT: Seconds

RDATA MODE: Basic

TARGET: oic-archdevpro-com.example.net

3. After you've successfully published the above changes, update your domain to use the OCI DNS nameservers.

## Configure the Oracle Fusion Application Credential Keys

If you have integration flows that subscribe to business events from Oracle Fusion Applications, configure the Credential Store Framework (CSF) keys in those applications so that both the primary and secondary Oracle Integration instances can successfully consume events.

1. Create separate CSF keys for the primary and secondary Oracle Integration instances in Oracle Fusion Applications (such as ERP, SCM or CRM).

2. Provide the same Oracle Integration username and password in the CSF keys of both the instances. In event of a password expiration, you must update the new password in both the keys.

When you activate a subscribe integration flow in the primary instance, the Oracle Integration custom endpoint is registered as a subscriber in the respective Oracle Fusion Application. And, in the event of a failover, the secondary instance starts consuming events after you activate the particular integration flow in the instance. The failed events (the events that aren't consumed until the failover tasks are successfully completed) will also be consumed by the secondary instance after you activate the integration flow.

## Migrate Metadata from the Primary Instance

The Oracle Integration metadata consists of connections, integrations, lookups, libraries, and packages.

After you've configured the primary instance with all your integration deployments, you can use REST APIs to export the metadata from the instance and import the same in the stand-by instance. You can do this as an initial, one-time task. Subsequently, you can employ continuous integration continuous deployment (CICD) to have the metadata synchronized between instances. See Automate Metadata Synchronization.

**Note:**

- You must use the original instance hostnames for all administrative tasks, including metadata migration.
- If you have already implemented CICD for both of your instances, you may skip this one-time migration task.
- You require an OCI Object Storage bucket to store the artifacts for the import or export APIs. It's recommended that you import the artifacts without activating them. As a result, the connections created in the secondary instance will not be in the *Configured* state. You can manually test the connections and activate them or use Connections REST Endpoints for the same.
- You can also use the Oracle Integration UI to export and import the metadata. See Export and Import Design-Time Metadata Between Instances.

Follow the steps provided here to synchronize the metadata between the instances using REST APIs.

1. Export the metadata from the primary instance.
   Invoke the REST API using the following `postman` or `curl` command. This action uploads the metadata to the OCI Object Storage Cloud Service bucket instance.

   **POST:** `http:/host:port/ic/api/common/v1/exportServiceInstanceArchive`

   - **Request Headers:**
     `Content-Type →application/json`
   - **Request Payload:**

```
{
    "jobName":"Pod1_Metadata"   -  If jobName is omitted filename will
default to "archive_Local_Suite_Instance-<jobId>.zip",
    "overwrite":false,      - defaults to false, will return error if
archive file already exists
    "exportSecurityArtifacts":true,
    "exportAppRoleMembers":true,
    "description":"Export description",
    "storageName",   - name of storage configuration, this can be used
instead of storageInfo, if both are defined storageInfo will take
precedence
    "storageInfo":{
        "storageUrl":"https://swiftobjectstorage.us-ashburn-
1.oraclecloud.com/v1/paasdevoic/<bucket name>",
        "storageUser":"<OCI user name>",
        "storagePassword":"<Auth Password>”
    }
}
```

- **Response Headers:**

  ```
  Location →http://host:port/ic/api/common/v1/exportServiceInstanceArchive
  /483
  ```

- **Response Payload:**

  ```
  {
      “jobId”: “483”,
      “location”: “https://swiftobjectstorage.us-ashburn-
  1.oraclecloud.com/v1/paasdevoic/<bucket name>”,
      “status”: “NOT_STARTED”
  }
  ```

- **Response Status:**
  - 202 Accepted **–** Export job was accepted.
  - 409 Conflict – Import or Export job is already running or Storage details are incorrect/missing, or file already exists (if overwrite is set to *false*).
  - 500 Internal Server Error – Error communicating to the Registry or Storage.

2. Check the status of the export operation using the following command:

   **GET:** `http://host:port/ic/api/common/v1/exportServiceInstanceArchive/{jobId}`

   If the status is *Completed*, the metadata has been successfully exported to the object storage bucket.

3. Now, import the metadata into the stand-by instance.
   Invoke the REST API using the following `postman` or `curl` command. This action retrieves the archive from the OCI Object Storage Cloud Service bucket instance where the archive was initially created.

   **POST:** `http://host:port/ic/api/common/v1/importServiceInstanceArchive`

   - **Request Headers:**

     ```
     Content-Type →application/json
     ```

   - **Request Payload:**

     ```
     {
         "archiveFile":"archive_Local_Suite_Instance-483.zip",
         "importActivateMode":"importOnly",      // ImportOnly |
     ImportActivate | ActivateOnly | StartSchedulesOnly
         "importSecurityArtifacts":true,
         "importAppRoleMembers":true,
         "importScheduleParams":true,
         "startSchedules":false,
         "description":"Import to standby",
         "storageName",   - name of storage configuration, this can be used
     instead of storageInfo, if both are defined storageInfo will take
     precedence
         "storageInfo":{
             "storageUrl":"https://swiftobjectstorage.us-ashburn-
     1.oraclecloud.com/v1/paasdevoic/<bucket name>",
             "storageUser":"OCI cloud user name",
             "storagePassword":”Auth password"
         }
     }
     ```

**Note:** Set the `importActivateMode` variable to `ImportOnly`, so that the integration flows are imported but aren't activated.

- **Response Payload:**

```
{
"jobId": "457",
"status": "NOT_STARTED"
}
```

- **Response Status:**
  - 202 Accepted **–** Export job was accepted.
  - 409 Conflict – Import or Export job is already running or Storage details are incorrect/missing, or file already exists (if overwrite is set to *false*).
  - 500 Internal Server Error – Error communicating to the Registry or Storage.

4. Verify the import status.

   **GET:** https://host:port/ic/api/common/v1/importServiceInstanceArchive/457

   Where *457* is the Job ID from the import response payload.

   **Note:** In this example, the integrations imported are not activated, conforming to the best practice. However, if you have many integrations, you can activate the stateless integrations while importing, but do not activate scheduled, publish-subscribe, polling, or business-events integrations.

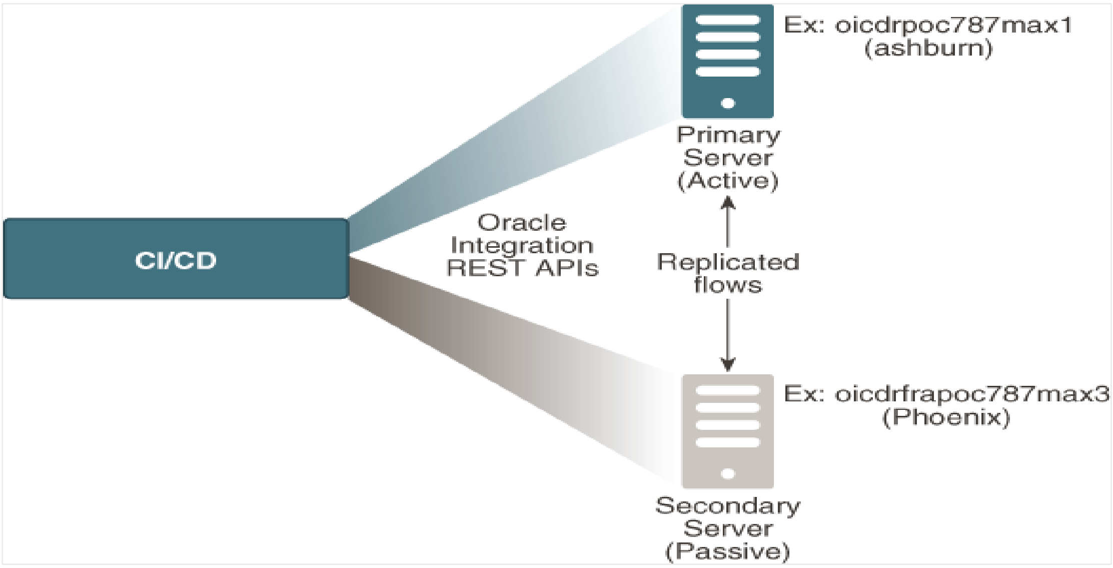
## Automate Metadata Synchronization

After the initial, one-time migration of the metadata is complete, you must keep the metadata synchronized between the instances using CICD.

You can use Jenkins or a similar tool to apply CICD to your instances and have the metadata synchronized. You can also use an OCI Compute instance as the Jenkins CI server and CD hub. See Integrations REST Endpoints and Connections REST Endpoints for the REST APIs to use.

The following figure shows the CICD forking to both the instances:

Image 4. CICD forking between primary and secondary Oracle Integration instances.



### Automate Scheduled Parameters Updates

Use REST APIs to update the integrations with scheduled parameters in the secondary Oracle Integration instance. You must frequently retrieve the metadata of integrations with scheduled parameters from the primary instance; for example, you can execute this every fifteen or thirty minutes. Subsequently, you can choose to update the corresponding integrations in the secondary instance either periodically or during a failover.

To get the details of each scheduled integration from the primary instance, see Retrieve an Integration. To update an integration in the secondary instance, see Update Scheduled Integration Parameters. (After a failover, you may have to manually update the parameter values retrieved earlier using this API to avoid reprocessing of old data.)

> **Note:** Optionally, you can use the export-import APIs discussed earlier, see Migrate Metadata from the Primary Instance. Ensure that both the instances are synchronized through CICD prior to executing the export-import operations.

### Monitor Your Instances

Regularly monitor the health of your active Oracle Integration instance. You can use the OCI health-check service or a third-party monitoring service.

Additionally, define a process to identify outages and, subsequently, trigger failovers.

### Execute Failover Tasks

To successfully switch from your primary instance to the stand-by instance during outages, perform the tasks listed here.

1. Stop the primary Oracle Integration instance if you're able to access the OCI Console.
2. Prepare your secondary instance.
    a. Use the Import API and import the latest snapshot extracted.
    b. Activate all relevant integrations.
    c. Update the scheduled parameters with the latest values to avoid reprocessing of old data.
3. Update the DNS record at your DNS provider or in the OCI DNS zone to route the traffic to the secondary instance.

After the failover process, the stand-by instance becomes your primary instance, and the instance previously designated as primary becomes the new stand-by instance.

> **Note:**
>
> - If your original primary instance restarts itself after the failover process, deactivate or shut down scheduled and polling-based integrations.
> - If there are backlogs (of asynchronous transactions) in the original primary instance, these may be triggered when the instance restarts, resulting in duplicate transactions. The backlogs belong to the faulted instances. You can choose how you handle them. See Set Data Retention for Runtime Instances.

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com     facebook.com/oracle     twitter.com/oracle

Disclaimer: If you are unsure whether your data sheet needs a disclaimer, read the revenue recognition policy. If you have further questions about your content and the disclaimer requirements, e-mail REVREC_US@oracle.com.