EUROPEAN COMMISSION

DIGIT
Digital Europe Programme

Domibus

Upgrading and Downgrading using Backup and Restore

Version [1.10]

Status [Final]

Date: 14-06-2022

Document Approver(s):

Approver Name	Role
DUMITRIU Bogdan	Project Manager

Document Reviewers:

Reviewer Name	Role
BACIU Cosmin	Domibus Technical Leader

Summary of Changes:

Version	Date	Created by	Short Description of Changes
0.01	16/12/2021	Sebastian-Ion TINCU	Initial version
1.00	20/01/2022	Sebastian-Ion TINCU Chaouki BERRAH Caroline AEBY	Review (headings, links, content etc.)
1.10	14/06/2022	Caroline AEBY	No more CEF references and FMB email updated

Table of Contents

1. Introduction

A backup of the existing version of Domibus, including the configuration and the persistent data, must be performed before starting the upgrade of Domibus to allow for rollback to the previous version in case of problems.

The purpose of this document is to describe the details of making a backup and restore it, if it is needed.

The following steps in each category assume that Domibus is currently running on a WebLogic or a WildFly application server, or on a Tomcat servlet container. We will further use the term application server when referring to any of these setups.

2. Procedure based on the application server

There are two scenarios to upgrade to Domibus current version: reuse the same application server instance or install a new one. The steps specific to each scenario are described below.

2.1. Using the same Application Server instance

In this scenario, the user reuses the same application server instance to upgrade to Domibus current version. The backup and restore steps are complex.

2.1.1. Backup

1. Create a backup of ongoing JMS messages while Domibus previous version is deployed, depending on your specific application server, following the individual steps for Tomcat (see $5.1), WildFly (see $7.1) or WebLogic (see $6.1);
2. Shut down the application server on which the Domibus previous version application is deployed;
3. Create a backup of the application server configuration and logs;
4. Create a backup of the Domibus WAR application or make sure you still have access to the official distribution of the current version;
5. Create a backup of all of the application server startup arguments.

2.1.2. Upgrade

Remember to port the ongoing JMS messages from the previous Domibus version to the current Domibus version.

1. Run the upgrade process in the current application server instance (details not in the scope if this document);
2. Port the previous Domibus version ongoing JMS messages after Domibus current version is successfully upgraded, following the individual steps for Tomcat (see $5.2), WildFly (see $7.2) or WebLogic (see $6.2).

2.1.3. Restore

1. [Optional, assuming the JMS Server running] Create a backup of ongoing JMS messages while Domibus current version is deployed. Follow the individual steps corresponding to the application server you are using: Tomcat (see $5.1), WildFly (see $7.1) or WebLogic (see $6.1);
2. Shut down the application server on which the Domibus current version application is deployed;
3. Restore the previous application server configuration and logs;
4. Restore the Domibus previous version WAR application;
5. Restore the previous application server startup arguments;
6. Start the application server on which the Domibus previous version application is deployed;
7. Restore the previous Domibus version ongoing JMS messages after the previous Domibus is successfully restored, following the individual steps for Tomcat (see $5.2), WildFly (see $7.2) or WebLogic (see $6.2).

2.2. Using a new instance of the Application Server

In this scenario, the user installs a new application server instance to upgrade to Domibus current version. The backup and restore steps are less complex.

2.2.1. Backup

Create a backup of ongoing JMS messages while Domibus previous version is deployed: depending on your setup, follow the individual steps for Tomcat (see $5.1), WildFly (see $7.1) or WebLogic (see $6.1).

2.2.2. Upgrade

Remember to port the ongoing JMS messages from the old Domibus previous version application to the new Domibus current version one.

1. Run the upgrade process in the new application server instance (details are not in the scope if this document);
2. Port the Domibus previous version ongoing JMS messages once the Domibus current version is successfully upgraded, following the individual steps for Tomcat (see $5.2), WildFly (see $7.2) or WebLogic (see $6.2).

2.2.3. Restore

1. (This first step is optional and has to be completed only if the JMS Server is running) Create a backup of ongoing JMS messages while Domibus current version is deployed. Depending on your specific application server, follow the individual steps for Tomcat (see $5.1), WildFly (see $7.1) or WebLogic (see $6.1);
2. Restore the previous Domibus version ongoing JMS messages once the previous Domibus is successfully restored, following the individual steps for Tomcat (see $5.2), WildFly (see $7.2) or WebLogic (see $6.2).

3. Database

Domibus current version comes with a new improved database schema, built around the use of dictionary data. Because of these changes, the database migration is a big step in the upgrade procedure. The user will create a snapshot of their database before attempting the upgrade procedure. This snapshot could be later used to revert the database to its previous working state.

3.1. Using the same Application Server instance

3.1.1. Backup

1. Shut down the application server on which the Domibus previous version application is deployed;
2. Create a snapshot of your current MySQL (see $8.1) or Oracle (see $9.1) database, including all the tenants in case of a multitenancy setup.

3.1.2. Upgrade

Run the upgrade process in the same database instance (details not in the scope if this document).

3.1.3. Restore

1. Shut down the application server on which the Domibus current version applications is deployed;
2. Restore the snapshot of your previous MySQL (see $8.2) or Oracle (see $9.2) database, including all the tenants in case of a multitenancy.

3.2. Using a new instance of the Application Server

3.2.1. Backup

1. Shut down the application server on which the Domibus previous version application is deployed;
2. Create a snapshot of your current MySQL or Oracle database, including all the tenants in case of a multitenancy setup;
3. Clone the current MySQL or Oracle database in the new database instance restoring the snapshot created in the previous step, including all the tenants in case of a multitenancy setup.

3.2.2. Upgrade

Run the upgrade process in the new database instance (details are not in the scope if this document).

3.2.3. Restore

No actions needed since the current database instance has not been altered by the upgrade process.

4. Domibus configuration

There are two scenarios to upgrade to Domibus current version: reusing the same Domibus configuration location or installing in a new one. The specific steps are described below.

4.1. Domibus is kept in the original location

In this scenario, the user reuses the same Domibus configuration location to upgrade to Domibus current version. The backup and restore steps are complex.

4.1.1. Backup

1. Shut down the application server on which the Domibus previous version application is deployed.
2. Create a backup of the Domibus configuration, referenced by the domibus.conf.location system property that is passed to the server startup arguments.
3. Create a backup of all the plugins or extensions used by Domibus.
    a. FS-Plugin
        i. Create a backup of the location for messages managed by the FS-Plugin (location defined in property fsplugin.messages.location);
        ii. Create a backup of ongoing JMS messages of the FS-Plugin send queue (property fsplugin.send.queue) while Domibus previous version is deployed, depending on your specific application server, following the individual steps for Tomcat (see $5.1), WildFly (see $7.1) or WebLogic (see $6.1).
    b. JMS Plugin
        i. Create a backup of ongoing JMS messages of the JMS-Plugin queues (properties jmsplugin.queue.in, jmsplugin.queue.out, jmsplugin.queue.reply, jmsplugin.queue.consumer.notification.error, jmsplugin.queue.producer.notification.error, jmsplugin.queue.*.queue, DOMAIN.jmsplugin.queue.*.queue) while Domibus previous version is deployed, depending on your specific application server, following the individual steps for Tomcat (see $5.1), WildFly (see $7.1) or WebLogic (see $6.1).
4. [Optional, assuming payloads are saved on disk] Create a backup of the location where payloads are saved on the file system (location defined in property domibus.attachment.storage.location).

4.1.2. Upgrade

Remember to port the ongoing JMS messages of the FS-Plugin and JMS Plugin from the old Domibus previous version application to the new Domibus current version one.

Run the upgrade process in the current Domibus configuration location (details not in the scope if this document).

4.1.3. Restore

1. [Optional, assuming the JMS Server running] Create a backup of all the plugins or extensions used by Domibus, while Domibus current version is deployed.
    a. FS-Plugin
        i. Create a backup of the location for messages managed by the FS-Plugin (location defined in property fsplugin.messages.location);
        ii. Create a backup of ongoing JMS messages of the FS-Plugin send queue (property fsplugin.send.queue) while Domibus current version is deployed, depending on your specific application server, following the individual steps for Tomcat (see $5.1), WildFly (see $7.1) or WebLogic (see $6.1).
    b. JMS Plugin
        i. Create a backup of ongoing JMS messages of the JMS-Plugin queues (properties jmsplugin.queue.in (http://jmsplugin.queue.in), jmsplugin.queue.out, jmsplugin.queue.reply, jmsplugin.queue.consumer.notification.error, jmsplugin.queue.producer.notification.error, jmsplugin.queue.*.queue, DOMAIN.jmsplugin.queue.*.queue) while Domibus current version is deployed, depending on your specific application server, following the individual steps for Tomcat (see $5.1), WildFly (see $7.1) or WebLogic (see $6.1).
2. Shut down the application server on which the Domibus current version application is deployed.
3. Restore the backup of the previous Domibus configuration.
4. Restore any backups of the previous Domibus plugins or extensions.
    a. FS-Plugin
        i. Restore the backup of the previous location for Domibus current version messages managed by the FS-Plugin (location defined in property fsplugin.messages.location).
5. Restore the previous Domibus JMS messages after Domibus previous version is successfully restored, following the individual steps for Tomcat (see $5.2), WildFly (see $7.2) or WebLogic (see $6.2).
6. [Optional, assuming payloads saved on disk] Restore any backup of the location where payloads are saved on the file system (location defined in property domibus.attachment.storage.location).

4.2. Domibus installed in a new location

In this scenario, the user installs a new Domibus configuration location when upgrading Domibus to current version.

4.2.1. Backup

1. Shut down the application server on which the Domibus previous version application is deployed.
2. Create a backup of all the plugins or extensions used by Domibus.
    a. FS-Plugin

i. Create a backup of the location for messages managed by the FS-Plugin (location defined in property fsplugin.messages.location);

ii. Create a backup of ongoing JMS messages of the FS-Plugin send queue (property fsplugin.send.queue) while Domibus previous version is deployed, depending on your specific application server, following the individual steps for Tomcat (see $5.1), WildFly (see $7.1) or WebLogic (see $6.1).

b. JMS Plugin

i. Create a backup of ongoing JMS messages of the JMS-Plugin queues (properties jmsplugin.queue.in (http://jmsplugin.queue.in), jmsplugin.queue.out, jmsplugin.queue.reply, jmsplugin.queue.consumer.notification.error, jmsplugin.queue.producer.notification.error, jmsplugin.queue.*.queue, DOMAIN.jmsplugin.queue.*.queue) while Domibus previous version is deployed, depending on your specific application server, following the individual steps for Tomcat (see $5.1), WildFly (see $7.1) or WebLogic (see $6.1).

3. [Optional, assuming payloads saved on disk] Create a backup of the location where payloads are saved on the file system (location defined in property domibus.attachment.storage.location).

4.2.2. Upgrade

Remember to port the ongoing JMS messages of the FS-Plugin and JMS Plugin from the old Domibus previous version application to the new Domibus current version one.

1. Run the upgrade process in the current Domibus configuration location (details not in the scope if this document).
2. Port the previous Domibus version ongoing JMS messages of the FS-Plugin and JMS Plugin after Domibus current version is successfully upgraded, following the individual steps for Tomcat (see $5.2), WildFly (see $7.2) or WebLogic (see $6.2).

4.2.3. Restore

1. [Optional, assuming the JMS Server running] Create a backup of all the plugins or extensions used by Domibus, while Domibus current version is deployed.
   a. FS-Plugin

i. Create a backup of the location for messages managed by the FS-Plugin (location defined in property fsplugin.messages.location);

ii. Create a backup of ongoing JMS messages of the FS-Plugin send queue (property fsplugin.send.queue) while Domibus current version is deployed, depending on your specific application server, following the individual steps for Tomcat (see $5.1), WildFly (see $7.1) or WebLogic (see $6.1).

b. JMS Plugin

   i. Create a backup of ongoing JMS messages of the JMS-Plugin queues (properties jmsplugin.queue.in (http://jmsplugin.queue.in), jmsplugin.queue.out, jmsplugin.queue.reply, jmsplugin.queue.consumer.notification.error, jmsplugin.queue.producer.notification.error, jmsplugin.queue.*.queue, DOMAIN.jmsplugin.queue.*.queue) while Domibus current version is deployed, depending on your specific application server, following the individual steps for Tomcat (see $5.1), WildFly (see $7.1) or WebLogic (see $6.1).

2. Shut down the application server on which the Domibus current version application is deployed.
3. Restore any backups of the previous Domibus plugins or extensions.
   a. FS-Plugin

      i. Restore the backup of the previous location for Domibus current version messages managed by the FS-Plugin (location defined in property fsplugin.messages.location).

4. Restore the previous Domibus version ongoing JMS messages after the previous Domibus is successfully restored, following the individual steps for Tomcat (see $5.2), WildFly (see $7.2) or WebLogic (see $6.2).
5. [Optional, assuming payloads saved on disk] Restore any backup of the location where payloads are saved on the file system (location defined in property domibus.attachment.storage.location).

5. JMS CONFIGURATION WHEN USING TOMCAT

The following instructions assume Tomcat has been configured with ActiveMQ using KahaDB as its persistence adapter.

5.1. Export JMS messages

The user will create a backup of the KahaDB directory containing all the persisted files.

1. Stop the Tomcat server and wait for the ActiveMQ process to exit normally.
2. The ActiveMQ connector listens on port 1199 by default so wait for any processes to exit successfully before continuing.
3. Create a backup of the KahaDB directory.

5.2. Restore JMS messages

The user will use the backup created in the previous step to restore the KahaDB persisted files.

1. Stop the Tomcat server and wait for the ActiveMQ process to exit normally.
2. The ActiveMQ connector listens on port 1199 by default so wait for any processes to exit successfully before continuing.
3. Restore the content of the KahaDB directory.

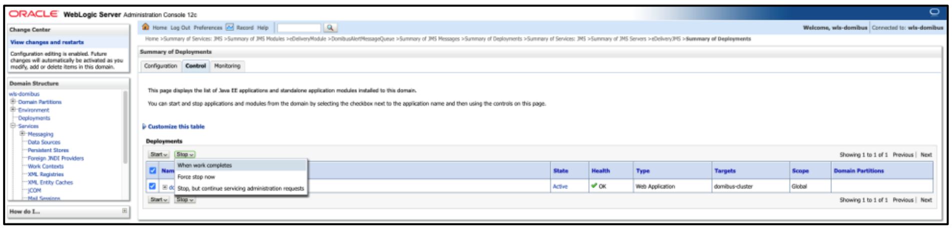
6. JMS CONFIGURATION WHEN USING WEBLOGIC

If the WebLogic server is running in production mode, please remember to lock your domain before doing any of the edit changes below.

6.1. Export JMS messages

The user will create backup files containing all of the ongoing JMS messages.

1. Start the WebLogic application server and log into the Admin Console.
2. Pause the consumption of the JMS Servers
    a. Go to the current_domain → Deployments → Control;
    b. Select the current Domibus deployment;
    c. Select Stop → When work completes as showed below:


ORACLE WebLogic Server Administration Console 12c
Change Center
View changes and restarts
Home Log Out Preferences Record Help
Welcome, wls-domibus | Connected to: wls-domibus
Home >Summary of Services: JMS >Summary of JMS Modules >eDeliveryModule >DomibusAlertMessageQueue >Summary of JMS Messages >Summary of Deployments >Summary of Services: JMS >Summary of JMS Servers >eDeliveryJMS >Summary of Deployments
Domain Structure
wls-domibus
Domain Partitions
Environment
Deployments
Services
Messaging
Data Sources
Persistent Stores
Foreign JNDI Providers
Work Contexts
XML Registries
XML Entity Caches
jCOM
Mail Sessions
How do I...
Summary of Deployments
Configuration
Control
Monitoring
Customize this table
Start
Stop
When work completes
Force stop now
Stop, but continue servicing administration requests
Showing 1 to 1 of 1 Previous | Next

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

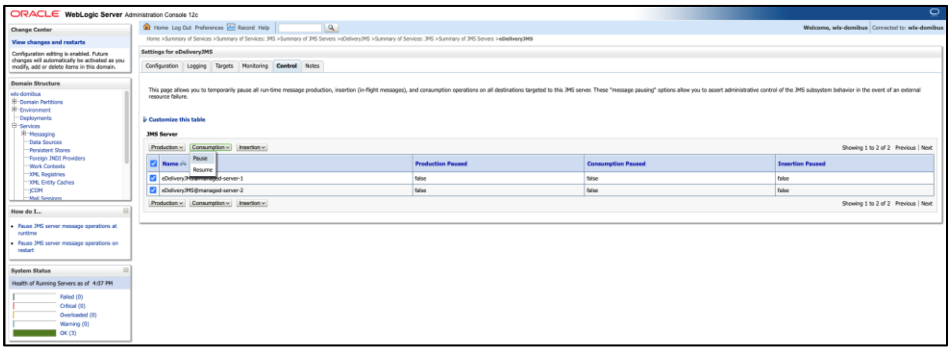
This page displays the list of Java EE applications and standalone application modules installed to this domain.

You can start and stop applications and modules from the domain by selecting the checkbox next to the application name and then using the controls on this page.

Deployments

☑	Nam	State	Health	Type	Targets	Scope	Domain Partitions
☑	⊞ do	Active	✓ OK	Web Application	domibus-cluster	Global	

    d. Go to the current_domain → Services → Messaging → JMS Servers → current_jms_server → Control;
    e. Select all the JMS Servers;
    f. Select Consumption → Pause from the menu as showed below:


ORACLE WebLogic Server Administration Console 12c
Change Center
View changes and restarts
Home Log Out Preferences Record Help
Welcome, wls-domibus | Connected to: wls-domibus
Home >Summary of Services >Summary of Services: JMS >Summary of JMS Servers >eDeliveryJMS >Summary of Services: JMS >Summary of JMS Servers >eDeliveryJMS
Domain Structure
wls-domibus
Domain Partitions
Environment
Deployments
Services
Messaging
Data Sources
Persistent Stores
Foreign JNDI Providers
Work Contexts
XML Registries
XML Entity Caches
jCOM
Mail Sessions
How do I...
Pause JMS server message operations at runtime
Pause JMS server message operations on restart
System Status
Health of Running Servers as of 4:07 PM
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (3)
Settings for eDeliveryJMS
Configuration
Logging
Targets
Monitoring
Control
Notes
Customize this table
Production
Consumption
Insertion
Pause
Resume
Showing 1 to 2 of 2 Previous | Next

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

This page allows you to temporarily pause all run-time message production, insertion (in-flight messages), and consumption operations on all destinations targeted to this JMS server. These "message pausing" options allow you to assert administrative control of the JMS subsystem behavior in the event of an external resource failure.

JMS Server

☑	Name	Production Paused	Consumption Paused	Insertion Paused
☑	eDeliveryJMS@managed-server-1	false	false	false
☑	eDeliveryJMS@managed-server-2	false	false	false

    g. Confirm by clicking on **Yes**;
    h. Go to the current_domain → Services → Messaging → JMS Modules → current_jms_module;
    i. Export all the messages for each JMS resource;

        i. Click on the JMS resource name inside the Summary of Resources table → Monitoring;

        ii. Select each destination in turn and click the Show Messages button

        iii. Select all ongoing message;

iv. Select Export → Export All;

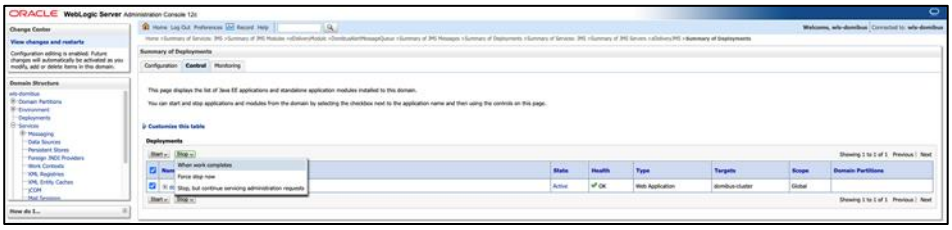
v. Finish the export wizard.

Each backup file should include the JMS resource and the destination in its file name for easy restoring later on. The messages must be restored in their corresponding JMS resources in the other application server.

6.2. Restore JMS messages

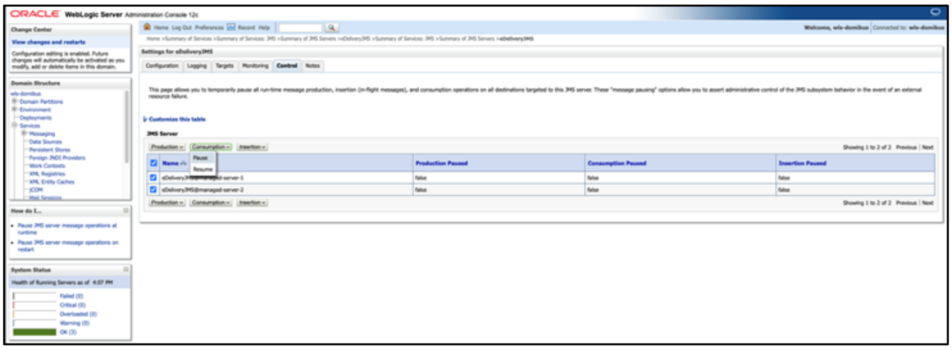
The user will use the backup created in the previous step to restore any ongoing JMS messages.

1. Start the WebLogic application server and log into the Admin Console;
2. Pause the consumption of the JMS Servers
    a. Go to the current_domain → Deployments → Control;
    b. Select the current Domibus deployment;
    c. Select Stop → When work completes as showed below:



Name	State	Health	Type	Targets	Scope	Domain Partitions
	Active	OK	Web Application	domibus-cluster	Global	

    d. Go to the current_domain → Services → Messaging → JMS Servers → current_jms_server → Control;
    e. Select all the JMS Servers;
    f. Select Consumption → Pause from the menu, as showed below:



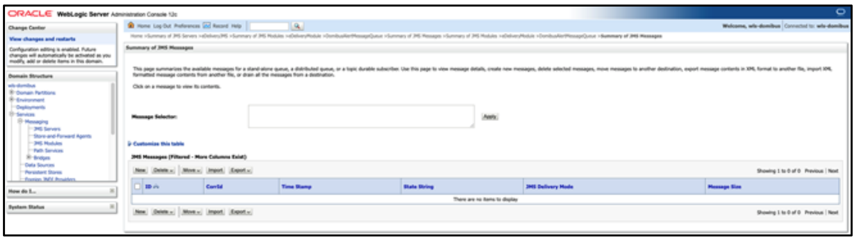
Name	Production Paused	Consumption Paused	Insertion Paused
eDeliveryJMS@managed-server-1	false	false	false
eDeliveryJMS@managed-server-2	false	false	false

    g. Confirm by choosing Yes;
    h. Go to the current_domain → Services → Messaging → JMS Modules → current_jms_module;
    i. Import all the messages into their corresponding JMS resource:

i. Click on the JMS resource name inside the Summary of Resources table → Monitoring;

ii. Select the corresponding destinations and click the Show Messages button

iii. Select Import:



Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Home >Summary of JMS Servers >eDeliveryJMS >Summary of JMS Modules >eDeliveryModule >DomibusAlertMessageQueue >Summary of JMS Messages >Summary of JMS Modules >eDeliveryModule >DomibusAlertMessageQueue >Summary of JMS Messages

Summary of JMS Messages

This page summarizes the available messages for a stand-alone queue, a distributed queue, or a topic durable subscriber. Use this page to view message details, create new messages, delete selected messages, move messages to another destination, export message contents in XML format to another file, import XML formatted message contents from another file, or drain all the messages from a destination.

Click on a message to view its contents.

Message Selector:

JMS Messages (Filtered - More Columns Exist)

□	ID	CorrId	Time Stamp	State String	JMS Delivery Mode	Message Size
	There are no items to display					

iv. Finish the import wizard.

7. JMS CONFIGURATION WHEN USING WILDFLY

7.1. Export JMS messages

The user will export all the message data using the Artemis CLI tool.

1. Stop the WildFly server;
2. Export the message data using the Artemis CLI tool.

7.2. Restore JMS messages

The user will import all the message data created during the previous step using the Artemis CLI tool:

1. Stop the WildFly server;
2. Export the message data using the Artemis CLI tool

8. MySQL Database snapshots

The following instructions are not optimised for the specific setup of the user. For performance or speed of execution, other approaches should be explored or favoured when available (e.g. using cloud backup capabilities, relying on DBA, infrastructure experts etc.).

8.1. Create a snapshot

The user must create a backup of all the databases used by the current Domibus installation: these databases also include the general user database and the domain databases in case of a multitenancy setup.

Create a database dump file for each database:

```
mysqldump -u root -p database_name > database_name.sql
```

For more options, please check the mysqldump documentation on https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html#mysqldump-option-summary.

8.2. Restore a snapshot

The user will use the snapshots created in the previous step to restore the specific databases to their previous states.

Restore every database from all existing database dump files tool:

```
mysql database_name < database_name.sql
```

For more options, please check the mysql documentation on https://dev.mysql.com/doc/refman/8.0/en/mysql-command-options.html .

9. Oracle Database snapshots

The following instructions are not optimised for the specific setup of the user. For performance or speed of execution, other approaches should be explored or favoured when available (e.g. using cloud backup capabilities, relying on DBA, infrastructure experts, etc.).

9.1. Create a snapshot

Create a database dump file for each schema:

```
expdp username/password@instance DIRECTORY=DATA_PUMP_DIR DUMPFILE=username.dmp
```

For more options, please check the Data Pump Export documentation on https://docs.oracle.com/en/database/oracle/oracle-database/19/sutil/oracle-data-pump-export-utility.html.

ORA-39087 / ORA-31631

If the import user does not have access to read from and write to the directory, the impdp utility will throw one of the following errors:

```
ORA-39087: directory name DATA_PUMP_DIR is
invalid
```

or

```
ORA-31631: privileges are required
```

This can be fixed by granting - as the SYS user - the user correct permissions to the DATA_PUMP_DIR directory:

```
-- CONNECT SYS/...@instance AS SYSDBA

 -- grant permissions to the user to export to the DATA_PUMP_DIR directory

GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR TO username

/
```

9.2. Restore a snapshot

The user will use snapshots created in the previous step to restore the specific schemas to their previous states.

Restore every schema from all existing database dump files:

```
impdp username/password@instance DIRECTORY=DATA_PUMP_DIR DUMPFILE=username.dmp
```

For more options, please check the Data Pump Import documentation on https://docs.oracle.com/en/database/oracle/oracle-database/19/sutil/datapump-import-utility.html.

ORA-39087 / ORA-31631

If the import user does not have access to read from and write to the directory, the impdp utility will throw one of the following errors:

```
ORA-39087: directory name DATA_PUMP_DIR is
invalid
```

or

```
ORA-31631: privileges are required
```

This can be fixed by granting - as the SYS user - the user correct permissions to the DATA_PUMP_DIR directory:

```
-- CONNECT SYS/...@instance AS SYSDBA

-- grant permissions to the user to import from the DATA_PUMP_DIR directory

GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR TO username

/
```

10. Contact Information

eDelivery Support Team

By email: EC-EDELIVERY-SUPPORT@ec.europa.eu

SUPPORT Service: 8am to 6pm (Normal EC working Days)