Contents

Contents ............................................................................... 3
Preface .............................................................................. 9
Intended Audience ......................................................... 10
Release Notes .................................................................. 10
Document Conventions .................................................. 10
Getting Help .................................................................... 12
Comments ........................................................................ 12

1 Overview ....................................................................... 13
Introduction ...................................................................... 14
Real-World Activities .................................................... 15
How It Works .................................................................... 16

2 Hardware and Software Environments ....................... 19
Hardware Requirements ................................................ 20
Storage Models .................................................................. 20
Host Interface ................................................................... 20
Device Type .................................................................... 21
Volume Type ..................................................................... 21
Software Requirements ........................................................... 21
Oracle Grid Software ............................................................ 22
Oracle Database ................................................................ 22
Server OS ......................................................................... 22
Hitachi Replication Software and Supported Configurations .... 23
Product Licensing ................................................................ 23
Creating the Hitachi Adapter Product License File ............... 23
Rules for Entering a License Key ........................................... 23

3 Required Pre-installation .................................................. 25
Pre-installation Steps ........................................................... 26
Install Java ....................................................................... 26
Configure Any Required Proxy Settings ................................. 26
Packages Required ............................................................. 26
Pre-installation Setup Requirements ....................................... 27
HUS Environment Configurations ......................................... 27
Pre- and Post-Installation Script ............................................. 28
Checks Performed .............................................................. 28
Using the Script ................................................................. 28

4 Required Pre-operation .................................................... 31
Initial Configuration ............................................................. 32
Storage Subsystem Latency .................................................. 32
Adapter Prerequisites (Block Storage) ..................................... 32
Setting the Primary Database to ARCHIVELOG Mode .......... 33
Configuring RAID Manager .................................................... 34
Command Device Configuration ........................................... 34
RAID Manager Setup .......................................................... 35
Setting up the Database ....................................................... 36
Setting up the Source and Target Servers ............................. 36
Account Privileges ..............................................................37
For Command Control Interface Commands Using the In-band Method ........................................................................37
Configuring Oracle Storage Manager for the Primary Database .37

5 Installation and Setup ...................................................... 41
Installation Checklist ............................................................42
Installing the Hitachi Storage Adapter RPM ..............................42
Adapter User Account Configuration ......................................43
Post-installation Steps ..........................................................43
Uninstalling Hitachi Adapter RPM ............................................44
Configuring the Virtual Environment ........................................44
Summary ..........................................................................45
Inventory Hosts File ...........................................................45
Set up Connectivity to the Servers ........................................46
Database Installation ..........................................................47

6 Configuring the Adapter ................................................... 49
Configuring the Application ....................................................50
Configuring the Storage ........................................................51
Configuration File Example ..................................................52
Configuring Common Information ...........................................53
Checking Inventory Files .......................................................53
Configuring the Source Database System .................................54
ASM Layout .......................................................................56
Configuring the Target Database System .................................56
Inventory and Hosts File .....................................................59
Configuring the Adapter Examples ..........................................60
9 Troubleshooting ................................................................. 91

Troubleshooting Tips ................................................................ 92
Oracle Error........................................................................... 92
Adapter Installation.............................................................. 92
Cloning Tips .......................................................................... 92
Archive Backup Directory.................................................... 92
Cloning RAC Tips............................................................... 92
Naming Tips......................................................................... 92
Directory Structure............................................................ 93
Deleting Database created using True Copy ......................... 93
Preface

- Intended Audience
- Release Notes
- Document Conventions
- Getting Help
- Comments
Note  The use of Hitachi Storage Adapter for Oracle Enterprise Manager - Database Cloning and all other Hitachi Data Systems products are governed by the terms of your agreement(s) with Hitachi Data Systems.

Intended Audience

This document guides customers of the Hitachi Storage Adapter for Oracle Enterprise Manager - Database Cloning: Oracle database administrators, storage administrators, and architects implementing a backup, recovery, and cloning solution for Oracle databases. Ideally, readers should have a solid understanding of the architecture, administration, and backup and recovery concepts of Oracle databases.

Release Notes

You can find release notes on the documentation CD. Release notes contain requirements and more recent product information that this manual may not fully describe. Be sure to review the release notes before installation.

Document Conventions

This document uses the following typographic conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold</strong></td>
<td>Indicates text on a window, other than the window title, including menus, menu options, buttons, fields, and labels. Example: Click <strong>OK</strong>.</td>
</tr>
</tbody>
</table>
| *Italic*              | Indicates a variable, which is a placeholder for actual text provided by the user or system. Example: *copy* source-file target-file  
  **Note:** Angled brackets (<> ) also indicate variables. |
| screen/code           | Indicates text on the screen or that you enter. Example: # pairdisplay -g oradb |
| <> angled brackets     | Indicates a variable, which is a placeholder for actual text provided by the user or system. Example: # pairdisplay -g <group>  
  **Note:** Italic font also indicates variables. |
| [ ] square brackets    | Indicates optional values. Example: [a | b] indicates that you can select a, b, or nothing. |
### Convention

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ } braces</td>
<td>Indicates required or expected values. Example: { a</td>
</tr>
<tr>
<td>vertical bar</td>
<td>Indicates that you have a choice between two or more options or arguments. Examples: [a</td>
</tr>
</tbody>
</table>

This document uses the following icons to draw attention to information:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Note" /></td>
<td>Note</td>
<td>Calls attention to important and/or additional information.</td>
</tr>
<tr>
<td><img src="image" alt="Tip" /></td>
<td>Tip</td>
<td>Provides helpful information, guidelines, or suggestions for performing tasks more effectively.</td>
</tr>
<tr>
<td><img src="image" alt="Caution" /></td>
<td>Caution</td>
<td>Warns the user of adverse conditions and/or consequences (for example, disruptive operations).</td>
</tr>
<tr>
<td><img src="image" alt="WARNING" /></td>
<td>WARNING</td>
<td>Warns the user of severe conditions and/or consequences (for example, destructive operations).</td>
</tr>
</tbody>
</table>
Getting Help

The Hitachi Data Systems Support Center staff is available 24 hours a day, seven days a week. Provisions for patches and fixes are restricted to normal business hours, 8 a.m. to 5 p.m. PST.

To reach us, please visit the support Web site for current telephone numbers and other contact information: http://www.hds.com/services/support/. If you purchased this product from an authorized HDS reseller, contact that reseller for support.

Before calling the Hitachi Data Systems Support Center, please provide as much information about the problem as possible, including:

- The circumstances surrounding the error or failure.
- The exact content of any error message(s) displayed on the host system(s).

Comments

Please send us your comments on this document: doc.comments@hds.com. Include the document title, number, and revision, and refer to specific section(s) and paragraph(s) if possible. Thank you! (All comments become the property of Hitachi Data Systems Corporation.)
Overview

- Introduction
- How It Works
Introduction

This guide describes version 03.0.0 of the Hitachi Adapter for Oracle Enterprise Manager – Database Cloning, software that lets you clone an Oracle database, working with a Hitachi storage system and Hitachi replication features.

Without stopping I/O operations at the primary server, you can create a hot clone of an Oracle database on the clone system. Cold and hot database cloning are supported.

In a software development context, each developer or QA engineer controls his or her own personal copy of the database. Developers and QA engineers can make modifications to these copies and even destroy them, if needed, without affecting other users. This applies to both standalone databases and RAC database configurations. These configurations allow Oracle Automatic Storage Management (ASM) to manage the database storage.

Once the database is set up for cloning operations, redo services are suspended. Then the adapter clones the specified LUNs/disks, yielding a copy of the Oracle database at a given point in time. Redo services resume, and the adapter activates and converts the cloned Oracle database.

In order to create a writable cloned database, the adapter uses the Hitachi ShadowImage technology and Hitachi Copy-on-Write (CoW) snapshot technology, which creates a storage-based volume copy of the database. The ShadowImage replication function runs without stopping I/O operations on the primary server. Hitachi ShadowImage replication software is a non-disruptive, host-independent, data-replication solution for creating copies of any database accessible database within a single Hitachi storage system. The CoW/HTI function helps in taking periodic point-in-time disk copies by not using much space.

- Clone the standby Oracle database by putting the production database into backup mode. Then perform ShadowImage cloning on each of the LUNs that you use for hot cloning and cold cloning. This creates a new standby Oracle database, dedicated to supporting cloning operations, on the Hitachi storage system. Once established, redo services stop temporarily.
- The software creates a clone of the LUNs/Oracle disks managed by ASM, yielding a copy of the database at that point in time.
- When this standby Oracle database is complete, redo services resume, and a snapshot of the database LUNs is created on Hitachi storage and mapped to the target system. The adapter activates and converts the snapshot of the Oracle database, so that it is ready to use for testing, development, or QA.
- The Hitachi storage system can support the primary (source) Oracle database, standby database, and the target database. If you use this method, deploy the Oracle instance on a server that is separate from the existing source Oracle database(s).
## Real-World Activities

<table>
<thead>
<tr>
<th>Features in the scenario</th>
<th>Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cloned databases for QA or development</strong></td>
<td><strong>Feature:</strong> Create Oracle cloned databases for testing and development. <strong>Scenario.</strong> Merwan, the Oracle administrator, must fulfill two requests for clones of the current database that QA and development groups can use. Fortunately, he has just replicated the database with Hitachi ShadowImage. Now he can create two clones on two teams’ clone servers, one for QA and the other for development.</td>
</tr>
<tr>
<td><strong>Hot Cloning</strong></td>
<td><strong>Feature:</strong> Hot cloning of the Oracle database operation means making the backup when transactions are actively running on the database. <strong>Scenario:</strong> Hot backups are useful for Pierre’s three-shift warehouse business requirements, to allow cloning operations on a database even though it is actively accessible by workers and may be in a state so that active users’ transactions are under way.</td>
</tr>
<tr>
<td><strong>Cold Cloning</strong></td>
<td><strong>Feature:</strong> Cold cloning of the Oracle database occurs when there are no transactions on the production database. <strong>Scenario:</strong> Cold backups are useful, because Marilyn’s small pet business requirements allow for a shut-down window (Monday) to clone the database. A novice at database administration, Marilyn finds the cloning adapter easy to use.</td>
</tr>
<tr>
<td><strong>Discovery</strong></td>
<td><strong>Feature:</strong> The discovery script provides information about the primary database and disk. <strong>Scenario:</strong> Talleyrand wants to analyze the LUNs connected to the primary server. He needs to know the disk size, ASM info, SCSI ID, and so on. He takes advantage of the discovery script to collect all the information that he needs in order to make sure space use is optimal.</td>
</tr>
<tr>
<td><strong>Space-efficient clones</strong></td>
<td><strong>Feature:</strong> A snapshot of the database is available. <strong>Scenario:</strong> Samantha, the Oracle administrator, must fulfill four requests for clones of the current database that QA and development groups can use. The database size is 24TB. Normally a clone would require 96 TB of storage space By using this adapter Sam’s storage is only 34TB of space. Snapshots of the target database reduced the space used.</td>
</tr>
<tr>
<td><strong>Performance load on the production database</strong></td>
<td><strong>Feature:</strong> Standalone database storage between the source and target database. <strong>Scenario:</strong> Mick, the Oracle administrator, must fulfill several requests for clones of the current database that IT groups need every week. If he had created direct clones of the database, it would have taken a long time. By using the adapter, Mick now clones only once and then resynchronizes the standalone storage every time there is a new request for clones of the database.</td>
</tr>
</tbody>
</table>
The adapter creates a new clone of the production database on the secondary server. By using the Hitachi storage system, the adapter clones the LUNs/disks that ASM manages, yielding a copy of the Oracle database. To produce a database clone, the adapter first uses the Hitachi ShadowImage or Hitachi Thin Image feature, creating a storage-based volume copy of the database. The replication function runs without stopping I/O operations on the primary server.

During hot cloning, the adapter moves the production database to backup mode and performs cloning of the disks. Then the software returns the production database to normal mode. After this step, the adapter generates a snapshot of the cloned secondary LUNs and presents the snapshot LUNs to secondary system. Later, it adds services and brings up the cloned Oracle database on the secondary server.

During cold cloning, the adapter shuts down the Oracle production database and clones all ASM disks. It creates a ShadowImage clone called the staging database.
Once the staging database or ShadowImage clone is complete, and depending upon the type of clone required, you can select from these:

- **CoW/HTI clone.** The adapter creates a snapshot of the cloned secondary LUNs and presents these snapshot LUNs to the secondary system, later adding services and bringing up the cloned Oracle database on the secondary server.

- **ShadowImage clone.** See the Snapshot clone explanation above.

- **TrueCopy clone.** The adapter creates new LUNs on a remote storage system and maps them to target system. Then it creates a clone of the database LUNs using Hitachi TrueCopy technology.

Once the cloning is complete, the adapter discovers these newly created LUNs on the target system. Later, it adds service and brings up the cloned Oracle database on the secondary server. In this fashion, it provides a production clone that is helpful for development or QA.
Hardware and Software Environments

- Hardware Requirements
- Software Requirements
- Product Licensing
Note: This release supports only the software and hardware listed here. Please refer to the release notes for the latest information.

# Hardware Requirements

<table>
<thead>
<tr>
<th>Item</th>
<th>Minimum Requirements</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Servers</td>
<td>Two servers: production and backup</td>
<td></td>
</tr>
<tr>
<td>CPU (for Linux)</td>
<td>1.2 GHz or faster Intel/AMD processor</td>
<td>Recommended: Four CPUs at 500MHz each</td>
</tr>
<tr>
<td>Memory</td>
<td>4 GB or more</td>
<td></td>
</tr>
<tr>
<td>Graphics</td>
<td>1024 x 768, 24-bit color or more</td>
<td></td>
</tr>
<tr>
<td>HDD Capacity</td>
<td>1 GB</td>
<td>&gt;=400MB on the drive that has the ${TMP} directory</td>
</tr>
<tr>
<td>Network</td>
<td>Gigabit connection recommended</td>
<td></td>
</tr>
</tbody>
</table>

## Storage Models

<table>
<thead>
<tr>
<th>Model</th>
<th>MicroVersion</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hitachi Unified Storage VM</td>
<td>73-03-09-00/00</td>
<td>X</td>
</tr>
<tr>
<td>Virtual Storage Platform</td>
<td>70-06-21/00</td>
<td>X</td>
</tr>
<tr>
<td>Virtual Storage Platform-G1000</td>
<td>80-02-23(V02+1)/00</td>
<td>X</td>
</tr>
<tr>
<td>Hitachi Unified Storage</td>
<td>0980/A-S</td>
<td>X</td>
</tr>
<tr>
<td>Virtual Storage Platform-Gx00</td>
<td>83-01-01-40/00</td>
<td>X</td>
</tr>
<tr>
<td>VSP Fx00</td>
<td>83-03-01-40/00</td>
<td>X</td>
</tr>
<tr>
<td>VSP Gx00 Unified</td>
<td>83-03-20-40/01</td>
<td>X</td>
</tr>
</tbody>
</table>

## Host Interface

<table>
<thead>
<tr>
<th>Interface</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC</td>
<td>X</td>
</tr>
<tr>
<td>iSCSI</td>
<td>X</td>
</tr>
</tbody>
</table>
### Device Type

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk</td>
<td>X</td>
</tr>
</tbody>
</table>

### Volume Type

<table>
<thead>
<tr>
<th>Volume Type</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>X</td>
</tr>
<tr>
<td>HDP/HDT</td>
<td>X</td>
</tr>
<tr>
<td>HTI</td>
<td>X</td>
</tr>
<tr>
<td>HRT</td>
<td>X</td>
</tr>
</tbody>
</table>

### Software Requirements

<table>
<thead>
<tr>
<th>Category</th>
<th>Software Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Install this Oracle and grid software on primary and secondary servers. Make sure that the following packages are present on the machine where the adapter will reside. jdk-7u71-linux-x64.rpm, python-devel, python-setuptools, libstdc++, openssl, openssl-devel, libgcc, glibc, libxslt-devel, redhat-lsb-core-4.1-27.el7.x86_64, redhat-lsb-4.1-27.el7.x86_64 and /lib/ld-linux.so.2.</td>
</tr>
</tbody>
</table>
| Oracle Database 11g and 12c (Single Instance or RAC) | Oracle Database 11.2.0.3, 11.2.0.4, and 12cR1 Single Instance  
Oracle Real Application Cluster Support Version 11.2.0.3, 11.2.0.4, and 12cR1 |
Note

- We recommend that you install your Linux operating system with the default software packages (RPMs), unless you specifically intend to perform a minimal installation and follow the directions for performing such an installation to ensure that you have all required packages for adapter.
- We recommend that you do not customize RPMs during a default operating system installation. A default installation includes most required packages and helps you to limit manual verification of package dependencies.
- You must install the packages (or later versions) listed above, and ensure that the list of RPMs and all of the prerequisites for these RPMs are installed.

**Oracle Grid Software**

<table>
<thead>
<tr>
<th>Oracle Database Software</th>
<th>Supported in Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle 11g Release 2 (11.2.0.3)</td>
<td>X</td>
</tr>
<tr>
<td>Oracle 11g Release 2 (11.2.0.4)</td>
<td>X</td>
</tr>
<tr>
<td>Oracle 12c Release 1 (12.1.0.2)</td>
<td>X</td>
</tr>
</tbody>
</table>

**Oracle Database**

<table>
<thead>
<tr>
<th>Oracle Database Software</th>
<th>Supported in Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle 11g Release 2 (11.2.0.3)</td>
<td>X</td>
</tr>
<tr>
<td>Oracle 11g Release 2 (11.2.0.4)</td>
<td>X</td>
</tr>
<tr>
<td>Oracle 12c (non CDB/non PDB)</td>
<td>X</td>
</tr>
</tbody>
</table>

**Note**
The adapter only supports ASM disk group-based database for cloning. In Oracle 12c, container databases (CDBs) and pluggable databases (PDBs) are not supported.

**Server OS**

<table>
<thead>
<tr>
<th>OS version</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redhat 6.5 64 bit</td>
<td>X</td>
</tr>
<tr>
<td>Redhat 6.6 64 bit</td>
<td>X</td>
</tr>
<tr>
<td>Redhat 7.2 64 bit</td>
<td>X</td>
</tr>
<tr>
<td>OEL 6.6 64 bit</td>
<td>X</td>
</tr>
<tr>
<td>OEL 7.2 64 bit</td>
<td>X</td>
</tr>
</tbody>
</table>
Hitachi Replication Software and Supported Configurations

<table>
<thead>
<tr>
<th>Replication Software</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShadowImage</td>
<td>X</td>
</tr>
<tr>
<td>Copy-On-Write /HTI</td>
<td>X</td>
</tr>
<tr>
<td>TrueCopy</td>
<td>X</td>
</tr>
</tbody>
</table>

Product Licensing

Hitachi Storage Adapter for Oracle Enterprise Manager for Database Cloning requires the purchase of a license for use.

The licensing policy depends on the capacity (in TB) that is mapped to the production database that the Hitachi adapter manages.

The license file exists at this path in the adapter:

```
/opt/hitachi/storage/odc/config/license
```

Creating the Hitachi Adapter Product License File

Create a text file called `hilicense.txt` in this directory:

```
/opt/hitachi/storage/odc/config/license
```

Rules for Entering a License Key

- Enter each 75-character license key with a license start date, separated by one space.
- Enter each license key and license start date on a separate line.
- If you do not enter the license key line in this format, the license key does not work.
- The license start date must be in the format `MM-DD-YYYY`. Example: 04-29-2017.
- This is the license key format (no spaces except before the date):

  `ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCMM-DD-YYYY`

  `ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCMM-DD-YYYY`
**Tips about hilicense.txt File**

- Lines beginning with a pound sign symbol (#) are comment lines and are ignored. You can add comments to help keep track of your license keys or add other notes.
- Do not move or delete this file. You may make a copy of this file as a backup.
- Do not change the ownership of the file.
- Do not change the file's permissions.
Required Pre-installation

- Pre-installation Setup Requirements
- Pre- and Post-Installation Script
**Pre-installation Steps**

**Install Java**

Download JDK or Java from the Oracle site and install the package. Set the Java home (JAVA_HOME) to the new installation of Java.

```
yum localinstall jdk-7u71-linux-x64.rpm
```

Note that you need to install Oracle jdk 1.7 and not Openjdk.

**Configure Any Required Proxy Settings**

Configure proxy settings as needed.

**Packages Required**

Make sure that the following packages are present on the machine before installing the adapter. If any are missing, install them.

- python-devel,
- python-setuptools
- libstdc++
- openssl
- openssl-devel
- libgcc
- glibc
- libxml2-devel
- /lib/ld-linux.so.2
Pre-installation Setup Requirements

Configure the storage subsystem as shown below.

<table>
<thead>
<tr>
<th>Item</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Licenses for Program Product</td>
<td>The required licenses for Program Products are:</td>
</tr>
<tr>
<td></td>
<td>• LUN Manager</td>
</tr>
<tr>
<td></td>
<td>• ShadowImage</td>
</tr>
<tr>
<td></td>
<td>• Thin Image</td>
</tr>
<tr>
<td></td>
<td>• TrueCopy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Host Ports</th>
<th>When FC is used:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Host group Security is enabled on all the ports.</td>
</tr>
<tr>
<td></td>
<td>• LU Mapping mode is enabled</td>
</tr>
</tbody>
</table>

**HUS Environment Configurations**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Authentication</td>
<td>• If using the Account Authentication function, select <strong>Storage Administrator (View</strong></td>
</tr>
<tr>
<td></td>
<td>and <strong>Modify</strong>) from the role check box in the Storage Management Software account addition window.</td>
</tr>
<tr>
<td></td>
<td>• For permissible characters and lengths allowed for usernames and passwords, refer to the Account Authentication User’s Guide.</td>
</tr>
</tbody>
</table>
Pre- and Post-Installation Script

Checks Performed

The adapter installation zip file contains a script that performs pre- and post-installation checks.

The pre-installation script checks the following items.

1. Supported operating system and version
2. Checks required packages needed for the adapter to work

The post-installation scripts check the following items. Run these scripts on those systems where HORCM is configured.

1. Checks that RMLIB is installed and proper permissions are set
2. Lists all the command devices present in the systems
3. Checks that HORCM is installed and HORCM services are running

Using the Script

Script name: hitachi-odc-install-check.py

The script takes the following options

- pre
- post

For example:

- python hitachi-odc-install-check.py --pre [-p]
- python hitachi-odc-install-check.py --post [-o]

When the argument “pre” is passed, the script performs pre-installation checks. When “post” is passed, it does post-installation checks, script logs the output. Log entries are color coded. Log messages and levels are:

- **INFO**: for information purposes
- **INFO_2**: yum plugin log messages
- **DEBUG**: yum plugin debug messages
- **WARNING**: warning messages that show any configuration changes required
- **ERROR**: Error in pre- or post-installation, such as a missing required packages or HORCM is not running.
Required Pre-installation

Hitachi Storage Adapter for Oracle Enterprise Manager - Database Cloning v03.0.0 User’s Guide
Required Pre-operation

- Initial Configuration
- Storage Subsystem Latency
- Adapter Prerequisites (Block Storage)
- Setting up the Database
- Verifying and Validating the OEM-Database Cloning
Initial Configuration

Perform the following operations before using the Hitachi Storage Adapter for OEM-Database Cloning. The adapter uses block storage.

<table>
<thead>
<tr>
<th>Item</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>License P.P.</td>
<td>ShadowImage, ThinImage, TrueCopy</td>
</tr>
<tr>
<td>Set up RAID Manager</td>
<td>See RAID Manager documentation for installation and setting up of the RAID Manager 01-36-03/04 or higher.</td>
</tr>
<tr>
<td>Configure ASM disks (ASM on Linux)</td>
<td>Follow Oracle Grid Infrastructure Installation Guide and Oracle ASM Administrator’s Guide. These are available on the Oracle web site.</td>
</tr>
</tbody>
</table>

Storage Subsystem Latency

Any modification that changes the state of the subsystem requires 20 seconds or more after the API call returns. The storage subsystem microcode behavior causes the delay.

Adapter Prerequisites (Block Storage)

The following sections address the prerequisites for using block storage to store the database-related files.

- Set the primary database to ARCHIVELOG mode. The database should be up and running.

- Create a host group/iSCSI target for staging storage. The adapter creates secondary LUNs and maps them to a host group/iSCSI-target if one is available. Ensure that a valid available host group/iSCSI-target exists on the secondary machine. If a host group is not available, create one so that the cloning operation can be performed. Provide the host group/iSCSI-target information in the adapter storage configuration file. This host group/iSCSI-target is a dummy, so do not map it to any target system (odc.yaml).

- Create a host group/iSCSI target on the secondary machines. The adapter creates secondary LUNs and maps them to a host group if one is available. Ensure that a valid available host group exists on the secondary machine. If a host group is not available, create one, so that the connection to the storage and secondary server exists.
Make primary, secondary, and ansible machines network-reachable. The primary server, secondary server, and ansible server should be network-reachable and should have password-free authentication enabled.

Configure the ansible server. We recommend you have a dedicated ansible server for the adapter RPM and HORCM are installation and configuration (you also can use the primary or secondary server as an ansible machine). Adapter-specific scripts can be run on this server.

Configure the underlying Oracle Data Storage Manager as a secondary machine.

Set the primary database to ARCHIVELOG mode.

**Setting the Primary Database to ARCHIVELOG Mode**

The database must be in ARCHIVELOG mode for the Hitachi ODC Storage Adapter to perform the cloning. This assures the recoverability of the database as an exact duplicate when mounted and started on a different system. Follow these steps:

1. Verify that the database is in ARCHIVELOG mode and correct as needed using the following commands to enter the SQL shell:
   
   ```bash
   $ source /usr/local/bin/oraenv
   ```

2. Specify the database name (SID). The maximum database name length is 8 characters. For a unique DB name, the limit is 30 characters.
   
   ```bash
   $ sqlplus / as sysdba
   ```

3. Verify the ARCHIVELOG mode setting as follows:
   
   ```sql
   SQL> archive log list;
   ```

4. For Oracle RACs running Linux, perform the following on one of the cluster nodes.
   
   ```sql
   SQL> select name, log_mode from v$database;
   ```

5. If the database log mode is not in archive log mode, the following commands place the RAC database in archive log mode:
   
   When the database log mode is not set to ARCHIVELOG mode, the following commands place the database in ARCHIVELOG mode:
   
   ```sql
   SQL> shutdown immediate;
   SQL> startup mount;
   SQL> alter database archivelog;
   SQL> alter system set log_archive_dest_1='location=+ARCHDG/<SID>' scope=spfile;
   SQL> shutdown immediate;
   ```
SQL> startup;

SQL> alter system set cluster_database=false scope=spfile sid='<SID>,'
- $ srvctl stop database -d <SID>
- $ sqlplus / as sysdba
SQL> startup mount;
SQL> alter system set log_archive_dest_1='location=+ARCHDG/<SID>,'
SQL> alter database archivelog;
SQL> alter system set cluster_database=true scope=spfile sid='<SID>',
SQL> shutdown immediate;
- $ srvctl start database -d <SID>

**Configuring RAID Manager**

To use the Hitachi Adapter with any RAID subsystem, you must configure the Command Device (CMD) and RAID Manager. This applies to the VSP, VSP G1000, VSP Gx00, VSP Fx00, and HUS-VM platforms.

Software requirements include the following:

RAID Manager: version RAID Manager: version 01-36-03/04

**Command Device Configuration**

Use the Hitachi Storage Navigator to configure the Command Device (CMD). The screenshot below shows how to set the CMD attributes.

*Note* Set user Authentication to **Enable** in order to use the Hitachi adapter.
For TrueCopy, configure the Command Device screen as follows:

**RAID Manager Setup**

For TrueCopy, set up the Command Device screen as follows:

Confirm the RAID Manager version:

```
# raidqry -h

Model : RAID-Manager/Linux

Ver&Rev: 01-36-03/04
```

In the folder `/etc`, create or link to an instance configuration file for a subsystem. Name the configuration file with `horcm<instance number>.conf`. For example, the configuration file `/etc/horcm0.conf` is for the instance 0.

Edit the configuration file as shown below:

```
HORCM_CMD

#dev_name dev_name dev_name
/dev/sdf #VSP 66033
/dev/sde #VSP 66034
```

Start an instance by executing `horcmstart.sh <horcm instance number>`:
>horcmstart.sh 0

Confirm that the instance is working:

```
#raidcom get command_status -I0 -s 66033 -login <uid> <password>
HANDLE SSB1 SSB2 ERR_CNT Serial# Description 00c9 - - 0 66033 -
#raidcom get command_status -I0 -s 66034 -login <uid> <password>
HANDLE SSB1 SSB2 ERR_CNT Serial# Description 00c4 - - 0 66034
```

On the ansible server (explained in the note below), configure the RAID Manager HORCM instance. Keep in mind that Hitachi Adapter for Oracle Database cloning requires a HORCM instance for all subsystems on which the adapter operates.

---

**Note**

ansible is *open-source software* designed to configure and manage computers. It combines multi-node *software deployment*, ad hoc *task* execution, and *configuration management*. It manages *nodes* over SSH or *PowerShell* and requires Python (2.4 or later) to be installed on those nodes. The system uses *YAML* to express reusable descriptions of systems. Ansible is commercially supported and sponsored by Ansible, Inc.

---

For brief instructions on configuring RAID Manager on a Linux server, see the *Hitachi Command Control Interface Installation and Configuration Guide*.

**Setting up the Database**

**Setting up the Source and Target Servers**

This is a one-time setup process, and the procedure is the same for both servers:

1. Install a *supported OS* on the host server. Update any packages that are out of date.

2. Prepare the operating system for the Oracle Automatic Storage Management (ASM) and database installation. Consult:

   - *Oracle Grid Infrastructure Installation Guide*
3. Create a user/group for the Oracle and ansible user.
4. Install all the patches for the 11g and RAC databases.

**Account Privileges**

- To execute the `sqlplus` commands, Oracle sysdba privileges are required.
- To execute ASM commands, Oracle sysasm privileges are required.
- CCI commands must be set up to run with operating system “root” privileges.

**For Command Control Interface Commands Using the In-band Method**

Hitachi Command Control Interface commands are usually run in-band, which refers to the method for transferring commands from the host server to the command device using the same data path as the disk I/O (FC).

The VSP and later family of storage arrays also support an “out-of-band” command transfer method which uses LAN. The Hitachi Storage Adapter for Oracle Enterprise Manager – Database Cloning scripts always use the in-band approach for best performance. For more information about in-band vs out-of-band operation, refer to the *Hitachi Command Control Interface User and Reference Guide* for your storage system model.

**Configuring Oracle Storage Manager for the Primary Database**

The adapter can clone only if the underlying Oracle data storage is configured as specified in any of the diagrams below.

<table>
<thead>
<tr>
<th><strong>Note</strong></th>
<th>Configure the listeners. They should be up and running for the source system.</th>
</tr>
</thead>
</table>

---

*Oracle Database Installation Guide*
Three Disk Groups Database Configuration
Two Disk Group Database Configuration
Installation and Setup

- Installation Checklist
- Installing the Hitachi Storage ODC Adapter RPM
- Post-installation Steps
- Uninstalling Hitachi Adapter RPM
- Configuring the Virtual Environment
- Database Installation
**Installation Checklist**

Here is a checklist to start the installation process.

1. See earlier chapters and make sure all the prerequisite software and libraries are installed.

2. Ensure that Python 2.6 or above is installed.

3. Make sure that the Python virtual environment is installed.

4. Make a backup copy of the installation package.

5. Create the user ansible on all the systems and add the ansible user to sudoers list (no password required) in the following locations:
   - Where the ODC adapter will be installed
   - On all the source database systems
   - On all the target database systems

6. The adapter does not manage passwords, so configure SSH keyless authentication on source and target systems.

**Installing the Hitachi Storage Adapter RPM**

We distribute the Oracle database cloning storage adapter as a single Linux RPM package. Install the Hitachi Adapter RPM on a single Linux server where Hitachi CCI is configured, that is, on the ansible server. The top-level installation directory for the adapter is:

```
/opt/hitachi/storage
```

To install the Hitachi storage adapter, follow the steps below. You must be the root user.

1. From a supported Linux 6 server, run:
   ```
rpm -ivh odc-hitachi-storage-03.0.0-0.el6.x86_64.rpm
   ```
   From a supported Linux 7 server, run
   ```
rpm -ivh odc-hitachi-storage-03.0.0-0.el7.x86_64.rpm
   ```

2. From a Linux 6 server, run:

3. `rpm -qpl odc-hitachi-storage-03.0.0-0.el6.x86_64.rpm`
   
   From a Linux 7 server, run:
   ```
rpm -qpl odc-hitachi-storage-03.0.0-0.el7.x86_64.rpm
   ```

View the content of the installed folder. The adapter is installed by default to `/opt/hitachi/storage` path, which is the installation path for the ODC adapter (`PROJECT_ROOT=/opt/hitachi/storage`).
To view ODC adapters installed on the physical host, run:
```
rpm - qa | grep odc
```

**Adapter User Account Configuration**

1. Install the adapter as root user.
2. Create the user ansible and group ansible on all the systems:
   - Where the adapter is installed (ansible server)
   - Source database server (Primary server)
   - Target database server (Secondary Server)
3. Give “ansible” and “oracle” users sudo permission.
4. Open the `/etc/sudoers`, file and add the below lines:
   ```
   oracle   ALL=(ALL)   NOPASSWD: ALL
   ansible  ALL=(ALL)   NOPASSWD: ALL
   ```
5. The adapter uses SSH no-password authentication to connect to different systems, so the adapter is designed for the user to configure the SSH no-password authentication from the adapter-installed system to the source and target systems.

**Post-installation Steps**

A post-installation script installs the required RPMs and Python packages. This file resides in the `/opt/hitachi/storage/script` directory. Run the post-installation script as the root user, and adjust proxy settings as needed.

```
source script/postinstall.sh
```

This script installs the following required packages:

- python-devel
- libxslt-devel
- libstdc++
- openssl
- glibc

Python packages include:

- pycrypto
- python-crypto
- paramiko=1.10
- fabric
- virtualenv
If the script fails to run, install the packages manually.

The script also sets environment variables for the ansible user in the `<ansible home directory>/.bashrc` file:

- `PROJECT_ROOT`
- `ANSIBLE_LOG_PATH`

## Uninstalling Hitachi Adapter RPM

To uninstall the Hitachi adapter, use the following procedure:

1. Run `rpm -e odc-hitachi-storage-03.0.0-0.el6.x86_64` to uninstall the adapter from a 6.6 server.
2. Run `rpm -e odc-hitachi-storage-03.0.0-0.el7.x86_64` to uninstall the adapter from a 7.2 server.

## Configuring the Virtual Environment

The adapter uses the Python virtual environment, so any new packages required for the adapter to run are installed in the virtual environment, rather than on the base Python installation directory. This makes uninstallation simple.

Steps to install the virtual environment are as follows.

1. Switch user to ansible:
   ```sh
   su - ansible
   ```
2. Go to the installation directory:
   ```sh
   cd /opt/hitachi/storage
   ```
3. Set up HTTP proxy if required and then run the following command, which installs all the required Python packages:
   ```sh
   fab setup
   ```
4. The following command activates the virtual environment of the ansible machine.
source <installation_path>/env/bin/activate

Summary

1. Install ODC RPM:
   \[\text{rpm -ivh odc-hitachi-storage-03.0.0-0.el6.x86_64.rpm}\]
   Preparing...
   ########################################################### [100%]
   1:odc-hitachi-storage
   ########################################################### [100%]

2. Go to /opt/hitachi/storage/script.
   \[\text{[root@rh63-239-97 ~]# cd /opt/hitachi/storage/script}\]

3. Set the proxy settings if required and run the postinstall.sh script as a root user.
   \[\text{[root@rh63-239-97 ~]# cd /opt/hitachi/storage/}\]

4. Switch to ansible user: su - ansible

5. Go to the folder /opt/hitachi/storage
   \[\text{[ansible@rh63-239-97 ~]# cd /opt/hitachi/storage/}\]

6. Set the proxy settings if required.

7. Run fab setup:
   \[\text{[ansible@rh63-239-97 storage]$ fab setup}\]

8. Edit the config files odc.yaml, <target-system-ip>, <source-system-ip>, hosts, and all config files.

9. Enable the virtual environment.
   \[\text{[ansible@rh63-239-97 ~]# cd /opt/hitachi/storage/}\]
   \[\text{[ansible@rh63-239-97 storage]$ source env/bin/activate}\]
   \[\text{(env)[ansible@rh63-239-97 storage]$}\]

10. Run the ODC script.
    \[\text{(env)[ansible@rh63-239-97 storage]$ cd odc/ansible_oracle/}\]
    \[\text{(env)[ansible@rh63-239-97 ansible_oracle]$ ansible-playbook odc-clone-database.yml -i inventory/hosts -vvvv}\]

Inventory Hosts File

After completing adapter installation, set up authentication without a password to source and target systems.
The ansible inventory hosts file lists and groups your servers. Its default location is /opt/hitachi/storage/odc/oracle-ansible/inventory/hosts.

If you want to have your ansible hosts file in another location, then you can set this environment variable:

```bash
> export ANSIBLE_HOSTS=/etc/ansible/hosts
```

Alternatively, you can specify the ansible host location when running commands with the --inventory-file= (or -i) flag:

```bash
> ansible all --inventory-file=/root/ansible_hosts -m ping
```

For more on the inventory hosts file, see: http://docs.ansible.com/intro_inventory.html

---

**Set up Connectivity to the Servers**

For this example, let us assume that you have servers with the hostnames 192.168.0.1 and 192.168.0.2. When doing your own install, replace those hostnames with your own.

Your /opt/hitachi/storage/odc/oracle-ansible/inventory/hosts file looks like this:

192.168.0.1

192.168.0.2

To avoid entering a password to access a server, start by setting up SSH key authentication to the children nodes. Then follow the steps below:

1. Generate the SSH key on the master node:
   ```bash
   ansible@ansihost:~# ssh-keygen -t rsa -C "ansible@ansihost"
   ```

2. Then copy your public key to the servers with ssh-copy-id:
   ```bash
   ansible@ansihost:~# ssh-copy-id ansible@192.168.0.1
   ansible@ansihost:~# ssh-copy-id ansible@192.168.0.2
   ```

3. Now, test the connectivity:
   ```bash
   ansible@ansihost:~# ansible all -m ping
   192.168.0.1 | success >> {
"changed": false,
"ping": "pong"
```
Database Installation

The adapter supports Redhat guidelines in Oracle database installation for single-instance storage and RAC (real application clusters). For RHEL installation, follow the steps provided by Redhat. For OEL, follow Oracle instructions for database installation.
Configuring the Adapter

- Configuring the Application
- Configuring Common Information
- Checking Inventory Files
- Configuring the Source Database System
- Configuring the Target Database System
- Configuring Adapter Examples
Configuring the Application

Modify these configuration files:

- `/opt/hitachi/storage/odc/config/odc.yaml` – Configuration file that contains the storage-specific information
- `/opt/hitachi/storage/odc/ansible-oracle/group_vars/all` – Configuration file that contains the hot/cold cloning parameters.
- `/opt/hitachi/storage/odc/ansible-oracle/inventory/hosts` – Configuration file that gives different server detail, such as primary, target, and ansible servers.

The `hilicense.txt` file is here: `/opt/hitachi/storage/odc/config/license`

Source and target database configuration files are separate.

The inventory/host file contains two groups, clonedb and sourcedb.

For each group, add a host IP address corresponding to the source or clone database. See the example below:

```
[clonedb]
172.17.239.193

[sourcedb]
172.17.58.95
```

Configuration information for each host belongs in the `host_vars` directory. Using the examples above, suppose your sourcedb group contains 172.17.58.95 as a source database hostname. Add a configuration file called “172.17.58.95” in `/opt/hitachi/storage/odc/ansible-oracle/host_vars/`. This file contains source database configuration information, in the same way that the configuration file “172.17.239.193” in `/opt/hitachi/storage/odc/ansible-oracle/host_vars/` does. In this way, we ensure that each host’s configuration data is maintained separately.
# Configuring the Storage

The storage configuration file is at: `<installationPath>/odc/config`, and the filename is `odc.yaml`.

Modifying `odc.yaml` is a one-time operation. The parameters are listed below.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Storage serial number</td>
</tr>
<tr>
<td>management_ips</td>
<td>For DF, this is the controller 2 IP address. For RAID systems, this is the management IP address</td>
</tr>
<tr>
<td>hrcmid</td>
<td>HORCM ID. Refer to RAID Manager documentation for HORCM settings.</td>
</tr>
<tr>
<td>username</td>
<td>Storage system username</td>
</tr>
<tr>
<td>password</td>
<td>Storage system password</td>
</tr>
<tr>
<td>storage_type</td>
<td>Set this to san.</td>
</tr>
<tr>
<td>san_type</td>
<td>Either DF or RAID</td>
</tr>
<tr>
<td>storage_pool_type</td>
<td>Storage pool type used for the staging database</td>
</tr>
<tr>
<td>storage_pool_id</td>
<td>Storage pool IDs used for staging database.</td>
</tr>
<tr>
<td>target_pool_id</td>
<td>Snapshot pool ID used for target database</td>
</tr>
<tr>
<td>target_pool_type</td>
<td>Storage pool type for snapshots:</td>
</tr>
<tr>
<td></td>
<td>• HTI</td>
</tr>
<tr>
<td></td>
<td>ShadowImage database option:</td>
</tr>
<tr>
<td></td>
<td>• HDP</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>access_group</td>
<td><strong>Mandatory.</strong> Host group or ISCSI target information for presenting cloned LUNs for the staging database. Create the host group or ISCSI target before you start cloning activity. This access group cannot be mapped to any server, so there is no need for host group mapping. This is a dummy access group.</td>
</tr>
<tr>
<td>access_type</td>
<td>iscsi or fc</td>
</tr>
<tr>
<td>port_name</td>
<td>storage port name</td>
</tr>
<tr>
<td>group_name</td>
<td>hostgroup/iscsi-target name</td>
</tr>
</tbody>
</table>

**Configuration File Example**

```
- storage_id: 10076
  management_ips:
    - 172.17.45.64
  hrcmid: 12
  type: raid
  username: user1
  password: hitachi1
  storage_type: san
  san_type: raid
  storage_pool_id: 12
  storage_pool_type: hdp
  target_pool_id: 14
  targetlistener
    _pool_type: HTI
    access_grp:
      grp_name: HITACHI_ODC_AUTOGEN
      port_name: CL1-A
      access_type: fc
    journal_record:
      id: 12
```
Configuring Common Information

This information is common across the source database system, the target database system, and the adapter location.

Path: `<installationPath>/odc/ansible_oracle/groupvars`

Filename: `all`

Please check the `all` file for configuration details. For example, the `all` file includes the parameters listed in the table below.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ansible_config_dir</td>
<td>string</td>
<td>/tmp/odc</td>
<td>Temporary path where all temporary files are created.</td>
</tr>
<tr>
<td>ansible_backup_dir</td>
<td>string</td>
<td>/tmp/odc_backup</td>
<td>Place where the adapter takes a backup of the device_target database configuration information, so that it can be used for deleting the database.</td>
</tr>
</tbody>
</table>

Checking Inventory Files

The adapter works against multiple systems in your infrastructure at the same time. It does this by selecting portions of systems listed in the ansible inventory file, which defaults this location `/opt/hitachi/storage/odc/ansible_oracle/hosts`.

Example

[clonedb]
172.17.239.193

[sourcedb]
172.17.58.95

The group names appear in brackets and are used in classifying systems and deciding what systems you are controlling at what times and for what purpose.
Configuring the Source Database System

You specify the configurations for the data production layout:

- List of ASM disk groups for the source system
- `asm_storage-layout`, which determines the asmlabel, device name, and logical unit number of the disk associated with each disk group.
- Configuration of the database home, grid home paths, RAC, and single-instance node specifications, along with the database name and database instance.

- Path: `<installationPath>/odc/ansible_oracle/ host_vars`
- Filename: `<source database host ip>`
- Check the sourcedb file for configuration details. For example, the `<source database host ip>` file includes the parameters listed in the table below.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ss_record_id</td>
<td>Integer</td>
<td>10076</td>
<td>Storage system serial number</td>
</tr>
<tr>
<td>asm_diskgroup</td>
<td>List of String</td>
<td>- REDODG</td>
<td>List of all the ASM disk group headers of the production database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DATADG</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- FRADG</td>
<td></td>
</tr>
<tr>
<td>asm_storage_layout</td>
<td>List of Objects</td>
<td>REDODG:</td>
<td>Disk group header:</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>- asmlabel: redo1p1</td>
<td><code>asmlabel</code>: The device name at source.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>device:</td>
<td><code>device</code>: The ASM disk path at source</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/mapper/redo1</td>
<td><code>se_record_id</code>: The scsi_id of the LUN used, in decimal form.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>se_record_id: 518-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DATADG:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- asmlabel: data1p1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>device:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/mapper/data1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>se_record_id: 520</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FRADG:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- asmlabel:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configuring the</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Adapterfra1p1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>device:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/dev/mapper/fra1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>se_record_id: 522</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- asmlabel: fra2p1</td>
<td></td>
</tr>
<tr>
<td>Field name</td>
<td>Type</td>
<td>Example</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------</td>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>device</td>
<td>String</td>
<td>/dev/mapper</td>
<td>Path where the ASM disk resides</td>
</tr>
<tr>
<td>role_separation</td>
<td>Boolean, (true, false)</td>
<td>True</td>
<td>Are the Oracle and grid users different?</td>
</tr>
<tr>
<td>oracle_asm_disk_path</td>
<td>String</td>
<td>/dev/mapper</td>
<td>Path where the ASM disk resides</td>
</tr>
<tr>
<td>oracle_home_db</td>
<td>String</td>
<td>/u01/app/oracle/product/11.2.0/dbhome_1</td>
<td>Oracle home path</td>
</tr>
<tr>
<td>oracle_home_grid</td>
<td>String</td>
<td>/u01/app/11.2.0/grid/</td>
<td>Oracle grid path</td>
</tr>
<tr>
<td>oracle_db_name</td>
<td>String</td>
<td>source</td>
<td>Oracle database name. The maximum database name</td>
</tr>
<tr>
<td>device_persistence</td>
<td>String</td>
<td>udev</td>
<td>If the database uses a multipath disk, it is set to <code>multipath</code>. Otherwise, the database is set to <code>udev</code>. You can also choose <code>asmlib</code>, if you use ASM.</td>
</tr>
<tr>
<td>oracle_instance_name</td>
<td>String</td>
<td>source</td>
<td>Oracle database instance name on the server.</td>
</tr>
<tr>
<td>master_node</td>
<td>Boolean, (true, false)</td>
<td>true</td>
<td>Is this a master node? For a single database instance, this is always true. For a RAC database, this should be set to true if the RAC node is 1.</td>
</tr>
<tr>
<td>is_racone</td>
<td>Boolean, (true, false)</td>
<td>false</td>
<td>True if it is RAC</td>
</tr>
<tr>
<td>oracle_dbf_dir_asm</td>
<td>String</td>
<td>DATADG</td>
<td>Source DB data disk group header</td>
</tr>
<tr>
<td>oracle_reco_dir_asm</td>
<td>String</td>
<td>FRADG</td>
<td>Source DB redo disk group header</td>
</tr>
<tr>
<td>multipath_enabled</td>
<td>Boolean, (true, false)</td>
<td>true</td>
<td>Set to true if multipath is enabled; otherwise, set to false.</td>
</tr>
<tr>
<td>configure_cluster</td>
<td>Boolean, (true, false)</td>
<td>true</td>
<td>Default true (do not change this)</td>
</tr>
</tbody>
</table>
### Configuring the Adapter

Configuration information related to target database system should be given in this file.

Path: `<installationPath>/odc/ansible_oracle/host_vars`

Filename: `<target database host ip>` (as mentioned in inventory/host system.

Check the `<target database host ip>` file for configuration details. For example, the `clonedb` file includes the parameters listed in the table below.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot_clone</td>
<td>Boolean (true, false)</td>
<td>true</td>
<td>True if hot cloning set to true, otherwise false.</td>
</tr>
<tr>
<td>access_grp</td>
<td>String</td>
<td>access_grp: grp_name: ODC-RAC2-239-193 port_name: CL4-A access_type: fc</td>
<td>Host group name for the clone system and port number Access type is iSCSI or fc.</td>
</tr>
<tr>
<td>oracle_asm_disk_path</td>
<td>String</td>
<td>/devmapper</td>
<td>Path for the ASM disk</td>
</tr>
<tr>
<td>role_separation</td>
<td>Boolean (true, false)</td>
<td>true</td>
<td>True if grid and Oracle users are different.</td>
</tr>
<tr>
<td>oracle_home_db</td>
<td>String</td>
<td>/u01/app/oracle/product/11.2.0/dbhome_1</td>
<td>Oracle home path</td>
</tr>
</tbody>
</table>

### ASM Layout

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asmlabel</td>
<td>String</td>
<td>REDODG</td>
<td>ASM Disk Header label</td>
</tr>
<tr>
<td>device</td>
<td>String</td>
<td>/devmapper/redol</td>
<td>Disk path in Server</td>
</tr>
<tr>
<td>se_record</td>
<td>Integer</td>
<td>518</td>
<td>Logical Unit Number</td>
</tr>
</tbody>
</table>

### Configuring the Target Database System

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot_clone</td>
<td>Boolean (true, false)</td>
<td>true</td>
<td>True if hot cloning set to true, otherwise false.</td>
</tr>
<tr>
<td>access_grp</td>
<td>String</td>
<td>access_grp: grp_name: ODC-RAC2-239-193 port_name: CL4-A access_type: fc</td>
<td>Host group name for the clone system and port number Access type is iSCSI or fc.</td>
</tr>
<tr>
<td>oracle_asm_disk_path</td>
<td>String</td>
<td>/devmapper</td>
<td>Path for the ASM disk</td>
</tr>
<tr>
<td>role_separation</td>
<td>Boolean (true, false)</td>
<td>true</td>
<td>True if grid and Oracle users are different.</td>
</tr>
<tr>
<td>oracle_home_db</td>
<td>String</td>
<td>/u01/app/oracle/product/11.2.0/dbhome_1</td>
<td>Oracle home path</td>
</tr>
<tr>
<td>Field name</td>
<td>Type</td>
<td>Example</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------</td>
<td>-----------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>oracle_home_gi</td>
<td>String</td>
<td>/u01/app/11.2.0</td>
<td>Oracle grid path</td>
</tr>
<tr>
<td>oracle_base</td>
<td>String</td>
<td>/u01/app/oracle</td>
<td>Oracle base path</td>
</tr>
<tr>
<td>oracle_db_name</td>
<td>String</td>
<td>clone</td>
<td>Oracle cloned database name. Cannot be the same as the source database name.</td>
</tr>
<tr>
<td>device_persistence</td>
<td>String</td>
<td>udev</td>
<td>If the database uses a multipath disk, set to \texttt{multipath}. Otherwise, it is \texttt{udev}.</td>
</tr>
<tr>
<td>oracle_instance_name</td>
<td>String</td>
<td>clone</td>
<td>Oracle database instance name on the server. Currently we do not support the same instance name as source database instance. An instance name can be up to 16 characters. The SID should not contain any underscores.</td>
</tr>
<tr>
<td>is_racone</td>
<td>Boolean</td>
<td>false</td>
<td>True if this is a RAC server (only for RAC database)</td>
</tr>
<tr>
<td>oracle_dbf_dir_asm</td>
<td>String</td>
<td>DATADG</td>
<td>Source data disk group header</td>
</tr>
<tr>
<td>oracle_reco_dir_asm</td>
<td>String</td>
<td>FRADG</td>
<td>Source redo disk group header</td>
</tr>
<tr>
<td>multipath_enabled</td>
<td>Boolean</td>
<td>true</td>
<td>Yes if multipath is enabled.</td>
</tr>
<tr>
<td>oracle_install_version_gi</td>
<td>String</td>
<td>11.2.0.4</td>
<td>Oracle database version</td>
</tr>
<tr>
<td>oracle_asm_init_dg</td>
<td>String</td>
<td>Crs</td>
<td>Oracle ASM init disk group</td>
</tr>
</tbody>
</table>
| target_asmdg_key      | String    | CL_             | Unique key (up to three characters) for creating ASM disk groups, so that they are unique across the system. When the same storage is cloned multiple times, this ensures that the ASM disk labels are unique on the server. \textbf{Note:} The \texttt{target_asmdg_key} has one
<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>underscore, which must be added at the end. No other underscore can appear. Example of a 3-character target ASM disk group key: <code>CL_</code> If the production database is cloned multiple times on the target system, change this key each time.</td>
<td></td>
</tr>
<tr>
<td>copy_type</td>
<td>String</td>
<td><code>ss : snapshots</code></td>
<td>Specifies the type of replication technology to use for cloning.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>si: shadow</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>image</code></td>
<td></td>
</tr>
<tr>
<td>enable_rdm</td>
<td>Boolean</td>
<td>RDM is not supported in the current release. Always configure this setting as false.</td>
<td></td>
</tr>
<tr>
<td>esx:</td>
<td>String</td>
<td><code>esx-ip</code>: IP address for the ESX <code>esx-user</code>: user name to login to ESX <code>esx-password</code>: password to login to ESX <code>vm-name</code>: VM name, to which the raw devices have to be mapped.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RDM is not supported in the current release. You do not need to update any of these fields.</td>
<td></td>
</tr>
<tr>
<td>ss_record_id</td>
<td>Integer</td>
<td><code>ss_record_id: 16454</code></td>
<td>This optional field is only for True Copy cloning (remote cloning). In this case, the source and target database are on different storage servers. This field represent target storage number. Example: Below host_vars/&lt;target system ip&gt; configuration file image contains storage_id</td>
</tr>
<tr>
<td>is_part</td>
<td>Boolean</td>
<td>true</td>
<td>True if partitions exist on</td>
</tr>
</tbody>
</table>
### Inventory and Hosts File

Path: `<installationPath>/odc/ansible_oracle/inventory`

Check the hosts file for configuration details. For example, the hosts file includes the parameters listed in the table below.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[cloned]</td>
<td>String</td>
<td>172.17.239.193</td>
<td>IP address of the clone server</td>
</tr>
<tr>
<td>[sourcedb]</td>
<td>String</td>
<td>172.17.58.95</td>
<td>IP address of the source server</td>
</tr>
<tr>
<td>[ansible]</td>
<td>String</td>
<td>172.17.37.219</td>
<td>IP address of the server where the application/adapter is installed</td>
</tr>
</tbody>
</table>

**Note**  For log4cplus configurations, please refer to the Logging section.

---

**For log4cplus configurations, please refer to the Logging section.**
Configuring the Adapter Examples

The user needs to update the following four configuration files for database cloning:

- `/opt/hitachi/storage/odc/config/odc.yaml` – Config file which contains the storage specific information.
- `/opt/hitachi/storage/odc/ansible-oracle/host_vars/<source system ip>` – Config file which contains the source system information.
- `/opt/hitachi/storage/odc/ansible-oracle/host_vars/<target system ip>` – Config file which contains the clone system info.
- `/opt/hitachi/storage/odc/ansible-oracle/inventory/hosts` – Config file which gives the different sever details such as primary, target and ansible server.

### inventory/ hosts

```
[clonedb] 172.17.239.55  # target system where the database need to be cloned
[sourcedb] 172.17.239.53  # source system from which database need to be installed
[ansible] 172.17.239.55  # ansible system where adapter is installed
```

```
host_vars/ <source system ip>

ss_record_id: 10075
asm_diskgroups:
  - REDO DG
  - DATADG
  - ARCHDG
asm_storage_layout:
  REDO DG:
    - asmlabel: REDO G
      device: /dev/sdk
      se_record_id: 2163
  DATADG:
    - asmlabel: DATAG
      device: /dev/sdi
      se_record_id: 2036
  ARCHDG:
    - asmlabel: ARCHG
      device: /dev/sdj
      se_record_id: 2045
oracle_asm_disk_path: /dev/oracleasm/disks
is_part: true
role_separation: true
oracle_home_db: /u01/app/oracle/product/12.1.0/dbhome_1
oracle_home_gi: /u01/app/12.1.0/grid
oracle_db_name: cdb
device_persistence: asmlib
oracle_instance_name: cdb1
master_node: true
is_racone: true
oracle_dbf_dir_asm: DATADG
oracle_reco_dir_asm: ARCHDG
multpath_enabled: false
configure_cluster: false
hot_clone: true
host_vars/ <target system ip>

host_clone: true
access_grp:
  grp_name: iscsi-oel72-239-58
  port_name: CL1-G
  access_type: iscsi
enable_rdm: false
esx:
  esx_ip: 172.17.239.40
  esx_user: root
  esx_password: Hitachi1
  vm_name: OEL72_node3
oracle_home_db: /u01/app/oracle/product/12.1.0/dbhome_1/
oracle_home_gi: /u01/app/12.1.0/grid
oracle_base: /u01/app/oracle
oracle_db_password: Hitachi1
oracle_asm_disk_path: /dev/oracleasm/disks
oracle_install_version_gi: 12.1.0.2
oracle_db_name: orcr
copy_type: ss
role_separation: true
device_persistence: asmlib
master_node: true
is_raceOne: true
oracle_instance_name: orcr1
oracle_dir_asm: DATADG
oracle_rec_dir_asm: REDODG
oracle_name_node1: oe172-239-58.sie.hds.com
configure_cluster: false
oracle_asm_init_dg: ARCHDG
multipath_enabled: false
target_asmdg_key: CL9_
# Storage information

storage:
  - id: 10075
    management_ips:
      - 172.17.47.55
    hrcm_instance: 1
    type: san
    username: hapham
    password: Hitachi1
    san_type: raid
    storage_pool_id: 52
    storage_pool_type: HDP
    storage_type: VSP G1000
    target_pool_id: 53
    target_pool_type: HTI
    access_grp:
      grp_name: HITACHI_ODC_AUTOCEN
      port_name: CL1-A
      access_type: fc
Database Cloning

- Discovery
- Cloning
- Cloning for Single Instances and RAC Databases
- Verifying and Validating the OEM-Database Cloning
- Importing and Deploying Adapter-OEM Software
- Cloning Tips
**Discovery**

Hitachi Storage Adapter for OEM database cloning provides a feature called *discovery* where you can find storage and ASM group information about the primary server. Run the playbook that you use for ODC cloning, without any extra configuration. The discovery operation fetches all the information related to storage and prints it on the screen.

```
ansible-playbook odc-discover-database.yml -I inventory/hosts
```

The above script analyzes the storage connected to the source machine and reads all the LUN information, iSCSI ID, device name, and ASM disk group for the particular database. Then the script prints the information on the screen.

The same data is in `ansible.log` so that you can consult it if needed.

If you want to run only the discovery operation, configure the `sourceb` object only and not the clone file.

After reviewing the checklist above, run the discovery script.

```
ansible-playbook odc-discover-database.yml -i inventory/hosts
```

**Cloning**

Two types of cloning are supported, hot and cold cloning.

**Cold Cloning**

Hitachi Storage Adapter for OEM database cloning supports cold cloning of the Oracle database. Use cold cloning when there are no transactions in progress. If your schedule is flexible enough to sustain a blackout window, cold cloning is a good choice.

In hot cloning, the database is put in backup mode during cloning.

In order to perform the cold cloning, change one parameter in config files: Change the `hot_clone` parameter to `false` in all target system config files and run the ODC ansible playbook.

**Hot Cloning**

Hitachi Storage Adapter for OEM database cloning supports hot cloning of the Oracle database. Use hot cloning when transactions are in progress.

In cold cloning, database is shutdown during cloning operation.

In order to perform the hot cloning, change one parameter in config files: Change the `hot_clone` parameter to `true` in all target system config files and run the ODC ansible playbook.
Types of Cloning with Replication Technology

There are three types of Cloning Supported

a. Shadow Image Clones  
b. CoW/HTI Clones  
c. True Copy Clones

Below are the scripts supporting the cloning using the replication technology.

- odc-clone-database-SI.yml. This script can complete cloning in one step.
- odc-clone-database.yml. This script also can complete cloning in one step.
- odc-clone-source-database-to-staging.yml. This script performs the first part of a two-step process.
  - odc-clone-staging-to-target-database.yml. This script completes the process above.
- odc-clone-source-database-to-staging-SI.yml. This script performs the first part of a two-step process.
  - odc-clone-staging-to-target-database-SI.yml. This script completes the process above.
- odc-clone-database-truecopy.yml
  
  This script can complete the cloning in one step.
Cloning using Snapshot Replication Technology

Example of a HTI based Oracle DB clones

There are two ways to execute the CoW/HTI Replication technology.

- Cloning in one run, where user can execute the entire cloning operation in one run.
- Cloning in two separate runs, user executes the clone operation using two separate scripts.

Cloning in Two Steps via Staging Server with Snapshots

Run the scripts in the order shown.

The `odc-clone-source-database-to-staging.yml` script clones the source database to the staging storage server. This is what the script does:

1. Checks the source database to see if there are already ShadowImage clones. If there are no clones, it creates new LUNs, presents LUNs, and clones the existing database LUNs to newly created LUNs.
2. Moves the database to backup mode to split the database cloned LUNs, and restores the database to normal mode.
3. Creates and splits snapshot images from secondary LUNs.
4. Presents the LUNs to the target system.
5. Copies the output to a specific place, so that the database script can use those output files.

The `odc-clone-staging-to-target-database.yml` script clones the staging database to the target database. This is what the script does:

1. Checks the target database and makes sure the LUNs are present on the target system.
2. Updates `multipath` and `udev` rules.
3. Partitions the devices.
4. Creates ASM disk groups.
5. Creates directories needed for the database.
6. Starts the database.
7. Renames the database.

**Cloning in one Steps with Snapshots**

The `odc-clone-database.yml` script also can complete cloning in one step.

**Cloning using Shadow Image Replication Technology**

Example of an SI based full Oracle DB clones
There are two ways to execute the ShadowImage replication technology.

- Cloning in one run, where user can execute the entire cloning operation in one run.
- Cloning in two separate runs, user executes the clone operation using two separate scripts.

**Cloning in One Step with ShadowImage**

The `odc-clone-database-SI.yml` script can complete database cloning in one step, creating new clones via ShadowImage replication.

When you run the script, the adapter creates a ShadowImage replica from the staging server and later uses it in the form of database disks to bring up the database. The script later deletes the ShadowImage pairs so that they can serve as independent clones.

The script follows this sequence:

1. Checks the source database.
2. Checks for any ShadowImage clones of the database.
3. If there are none, creates new LUs, presents them, and clones the database LUs to newly created LUs.
4. Sets the database to backup mode, splits the database cloned LUs, and returns the database to normal mode.
5. Creates ShadowImages from secondary LUs, splits the ShadowImages, and deletes the pairs.
6. Presents the LUs to the target system.
7. Copies the output to a specific location, so that the database script can use those output files.

**Cloning in Two Steps with ShadowImage**

Before you start the first script, `odc-clone-source-database-to-staging-SI.yml`, make sure that there is an available pool with enough space to create disks for a clone database.

Plan to run the two scripts one after another.

`odc-clone-source-database-to-staging-SI.yml` follows this path:

1. Checks the source database.
2. Sees if any ShadowImage clones exist on the source system.
3. If there is no clone, it creates new LUs, presents the LUs, and close the database LUs to newly created LUs.

4. Moves the database into backup mode, splits the cloned LUs, and returns the database to normal mode.

5. Creates ShadowImage instances from secondary LUs, splits the ShadowImage clones and deletes the pairs.

6. Presents the LUs to the target system

7. Copies the output to a specific location, so that the database script can use the output files.

Next, run the `odc-clone-staging-to-target-database-SI.yml` script to complete the second portion of the process. The script:

1. Checks the target database.

2. Sees if LUs are present on the target system.

3. Updates multipath and udev rules.

4. Partitions the devices.

5. Creates ASM disk groups.

6. Creates directories needed for the database.

7. Starts the database.

8. Renames the database.

This script can be used to do a cleanup of the disks, when the script fails at or after the storage cloning is complete and before creation of disk groups.
Cloning using TrueCopy Replication Technology

Example of a True Copy Remote based full Oracle DB clones

Note From a given database, you can create TrueCopy clone. You cannot create both types of clones from a single database.

In order to clone using True Copy replication, user need to execute the script odc-clone-database-truecopy.xml.

odc-clone-database-truecopy.yml performs the replication with TrueCopy.

odc-validate-source-and-target.xml validates the source and target validation for both of the above scripts, which is useful for post-check validations of the source and target databases.

Deleting a Database Clone

odc-delete-clone-database.yml allows you to delete a database and its underlying storage. Specify the cloned database name and the copy type as ss or si.

This script first erases the cloned database and ASM disk groups, then deletes the replication pairs created and the LUs created for the cloned database.
Note
This script does not remove the udev rules/multipath entries, so make sure you remove the udev rules/multipath entries after running the script and reload the rules. This Script does not work for True copy, user need to manually delete database, and storage entities.

This script can help to clean up of the disks, when the script fails at or after the storage cloning is complete and before the creation of disk groups.

If unmount/delete of the cloned disk groups fails, be sure to unmount/delete the disk groups manually. The Delete Clone script does not perform a force delete of the disk groups. This is applicable for any failure.

Cloning for Single Instances and RAC Databases

Review the Installation Checklist and then proceed with the following steps.

1. Update all the configuration information:
   <installation_path>/odc/config/odc.yaml

2. Activate the source environment:
   source $(installation_path)/env/bin/activate

3. Update the clonedb and sourcedb files in
   group_vars$PROJECT_ROOT/odc/ansible_oracle/host_vars/<source system ip>
   $PROJECT_ROOT/odc/ansible_oracle/host_vars/<target system ip>

4. Update the hosts file in the inventory directory:
   $PROJECT_ROOT/odc/ansible_oracle/inventory/hosts

Verifying and Validating the OEM-Database Clone

In RAC systems, the adapter clones the database to only one node. You must add the instance to the rest of the nodes. Please follow the steps provided by Oracle for this purpose.

After making sure that the production databases are discovered, check the /etc/sudoers file in case the security setting is on. Turn it off if necessary.

Follow these steps to verify and validate cloning.

1. Create a tablespace and data files at the source database.
2. Perform the database cloning operation.
3. On the production machine, shut down the database:
4. On the secondary machine, start up the cloned database.
   ```sql
   export ORACLE_HOME=$ORACLE_HOME
   export ORACLE_SID=${CloneDB_SID}
   ./sqlplus / as sysdba
   SQL> Startup;
   ```

5. Make sure the earlier table space and data files exist in the target server as well.

6. Also try to create new table spaces at the cloned database.
   ```sql
   SQL> create tablespace TS_Test1 datafile
   '+SIAMSDATA/ODCAMSSI/test1.dbf' size 50M extent management
   local autoallocate;
   ```
   Tablespace created.

7. Create a table called `my_test` on tablespace `TS_Test1`.
   ```sql
   SQL> create table my_test(x number) tablespace TS_Test1;
   Table created.
   ```

8. Shut down the database.

---

**Importing and Deploying the Adapter - OEM Software**

The OEM archive file, `12.1.0.4.0_Hitachi.Adapter.ODC_2000_0.opar`, comes packaged in the
Hitachi_Adapter_for_Oracle_Enterprise_Manager_Database_Cloning-03.0.0.Linux.zip distribution file.

**System Prerequisites**

Complete OEM Cloud Control system prerequisites and installation procedures before following the steps in this chapter.

**Hardware Requirements**

Refer to the Oracle Enterprise Manager documentation for system hardware requirements.
## Software Requirements

### Base OS

The adapter is installed in the Oracle Enterprise Manager (OEM) Oracle Management System (OMS) that is supported under Red Hat Linux 6.6 with all the latest updates required for Oracle compatibility. Refer to Oracle for 6.6 and Red Hat for RHEL 6.6 installation procedures.

Prior to the Oracle installation, the user “oracle” must be created and belong to “oinstall” as a primary group and “oasm” and “disk” as secondary groups. Follow the Linux documentation for creating users and groups on the base OS.

### Oracle 11g Release 2 Database

Oracle Enterprise Manager (OEM) requires an Oracle 11g Release 2 instance created to store the OMS repository, also known as the OMR (Oracle Management Repository). Refer to Oracle for Oracle 11g Release 2 database installation procedures.

Ensure that an instance of OMA (Oracle Management Agent) is running on the same device that has the Oracle Database 11g Release 2 running.

### Oracle 11g Release 2 Grid Control

On Linux with Oracle ASM, Oracle Grid Control is required to provide the Oracle ASM. Refer to Oracle for Oracle 11g R2 Grid Control installation procedures.

### Oracle OEM 12c Cloud Control

Oracle Cloud Control is required to provide the management web front-end for OEM adapters. Refer to Oracle for Oracle OEM 12c Cloud Control installation procedures.

### WebLogic 10.3.2 Web Server

The OEM OMS requires an instance of WebLogic 10.3.2 web server to run the web components for the Grid Control web client. Refer to WebLogic web server installation procedures.

### Linux Packages

The Hitachi Storage Adapter for Oracle Recovery Manager is written in UNIX bash shell, and so requires the bash package that is standard with OEL and RHEL 6.6 OS installations.

Additional information is available at the following websites:
- Oracle: [https://edelivery.oracle.com/linux](https://edelivery.oracle.com/linux)
- Red Hat: [http://www.redhat.com/support/](http://www.redhat.com/support/)
**OEM Jobs Integration**

In the Hitachi Storage Adapter for Oracle Database Cloning folder is an OEM adapter archive file called 12.1.0.4.0_Hitachi.Adapter.ODC_2000_0. This archive file contains one OEM adapter targets.

The adapter contain job types that invoke the Hitachi Storage Adapter for Oracle Database Cloning scripts. You can create and schedule jobs built from these job types.

**Summary**

Using the Hitachi Storage Adapter for Oracle Recovery Manager OEM adapters involves the following steps:

- Prerequisites
- Importing the Adapter Archive
- Deploying the Adapters
- Creating a Target Instance
- Creating and Scheduling Jobs

**Prerequisites**

Complete the following prerequisites before importing the adapter archive.

**Setting up the Software Library**

This is a one-time setup procedure on the host.

1. Create a folder in the system where Enterprise Manager is installed. For example, `/home/oracle/swlib1`.
2. From the console, select `Enterprise> Provisioning and Patching> Software Library`.
3. Click **Actions**, then **Administration**.
4. Click **Add**.
5. In the pop up window, enter a name and location for the folder you created in step 1. For example, you might enter `swlib1` and `/home/oracle/swlib1`
   - Wait for the processing to finish.
6. Synchronize the EM CLI client with an Oracle Management Service (OMS). Issue the following command:
Run: `emcli sync`

If the synchronization fails due to a Session Expired error, log in again as `emcli login -username=sysman`. Now run the `emcli sync` again.

After synchronization, all verbs and associated command-line help available to this OMS become available at the EM CLI client.

**Importing the Adapter Archive**

Import the adapter archive into Enterprise Manager Cloud Control using the `emcli import_update` command as follows:

```
emcli import_update -file="path to the opar file" -omslocal
```

The `-omslocal` flag indicates that the adapter archive is on the same system where you are running this command and the path exists on this system.

For example:

```
emcli import_update -file=/tmp/sample_plugin.opar -omslocal
```

Operation completed successfully. Update has been uploaded to Enterprise Manager. Please use the Self Update Home to manage this update.

A list of add-ons appears when you log in to the OEM.
Deploying the Adapter

1. Select the ODC adapter and click the **Deploy On** tab.

2. Choose **Management Server** from the **Deploy On** dropdown list to deploy it on OMS.
3. To add the adapter to the desired agents, select Management Agent from the Deploy On dropdown list.

4. Choose TargetType and the Agent to continue.

5. Deploy on the agent in a similar way.
Creating and Scheduling Jobs

Follow these steps:

1. From the Home page, select **Enterprise > Job > Library**.
1. On the Job Library page, select an appropriate job type from the **Create Library Job** drop down list.

2. On the job’s page, fill in the name and description for the job and select a target. Then click **Save to Library**.

3. Confirm that the job is created successfully.

4. Specify the job name and add the target.
5. Add the preferred credentials for the ansible user.

6. Once you have submitted the job, you can check its status.

Similarly, you can run other jobs on OEM. Here is a list of supported jobs:

- odc_clone_database_image
- odc_clone_database_image_si
- odc_clone_source_database_to_staging
- odc_clone_source_database_to_staging_si
- odc_clone_staging_to_target_database
**Cloning Tips**

- Set up a running listener on both source and target systems. Test to verify.
- Make sure you remove the `udev/multipath` entries after running the delete script manually and reloading/restarting the `udev/multipath` rules.
- Specify the scan order correctly while installing asmlib software, based on whether or not you are using multipath.
- Make sure to whitelist the devices on his target system when enabling multipath on target system by editing the `multipath.conf` file.
- The clone script for SI is not supported on HUS systems, because L2 pairs are not supported in HUS for SI.
- RAC database cloning is supported only for the ASM 1 node of the target RAC system. Follow Oracle’s guidelines to clone the rest of the nodes.
- When performing single-instance database cloning, remember that you can go only from a non-RAC source to a non-RAC target.
- We support only ASM disk group-based databases for cloning. No other types are not supported.
- Before starting the cold cloning script, make sure the source database is up and running. The adapter shuts down the database and restarts it.
- During Cloning, if a storage operation fails, and if subsequent auto cleanup fails, you need to manually perform the respective storage cleanup operations found in the `cleanup operations` section.
- If a cloning script fails while newly created logical units are not discoverable, make sure the logical units can be discoverable and then run `odc-clone-staging-to-target-database.yml`.
- For cloning, make sure the archive backup logs have enough space. Cleanup the `archivelogs` on the source database on regular basis, otherwise cloned database may fail to open.
- All the ASM disk groups on the target database are prepended with `target_asmdg_key`. This allows the same source ASM disk groups to be cloned multiple times on the target system.
- Before cloning, the source database should be up and running.
- The source and target database names should be different. Oracle allows a maximum of eight characters for database. Set the archive `logfile destpath` according to your needs.
- Back up an entire database before cloning it.
You can clone databases pertaining to same 11g Oracle database versions.

During cloning, if the script fails to mount the ASM Disk groups, a problem could occur with the target cluster. Reboot the target cluster and repeat the cloning procedure.

Ensure that the configuration of the database is set correctly, and run the adapter only once, uninterrupted, when creating a clone.

A password file is created on the clone system.

ASM disk groups should be in the same order in the configurations for `clonedb` and `sourcedb` files.

Temporary files located in the default temporary file location are replicated.

An Spfile for the cloned database after cloning is created in `DATADG/<dbname> spfile`.

After Cloning, user needs to change the cloned database Archive Log destination path.

Make sure the `lsb` core packages are installed for Redhat Linux

If the cloning script fails for any user configured ASM related issues such as failure in registering resource ora.asm, user have to re-run the script after making sure ASM is running.

The `odc-delete-clone-script` helps to clean up the disks when the script fails after the storage cloning is complete and before the creation of the disk groups. During this operation, a database/disk group delete failure will be ignored. Check the ansible logs to determine the cause of the failure.

When executing multiple cloning operations, you can change only the following parameters in the target system file each time and run the cloning script.

- `<target_asmdisk_key>`
- `<database_name>`
- `<instance name>`

**Manual Cleanup**

1. First, issue the following commands to clean up or delete a cloned database.
To delete the database:

```
srvctl stop database -d {{ oracle_instance_name }} -o immediate
```

In order to delete the database.

```
srvctl remove database -d {{ oracle_db_name }} -f
```

dbca -deleteDatabase -sourceDB {{ oracle_db_name }} -sysDBAUserName sys -sysDBAPassword {{ oracle_db_passwd }} -silent

2. To delete the disk groups, run these commands:

```
srvctl stop diskgroup -g < diskgroup>
srvctl remove diskgroup -g < diskgroup>
```

3. To delete the device entries (stale entries):

```
echo "1" > /sys/block/`device`/device/delete
```

**Note** You can also use `-f` to perform the above command for multiple paths.

4. Log in to your storage and delete the cloned LUNs and LUN paths.

5. Make sure you remove the udev/multipath entries after running the delete script manually and reloading the udev rules/multipath.

**Tip** If cloning fails with the below errors, delete the original disk group from target system and clean up the storage.

- ORA-15017: diskgroup "HMREDO" cannot be mounted
- ORA-15040: diskgroup is incomplete (DBD ERROR: OCISmtExecute)
This chapter includes two sections:

- Logging
- Messages
Logging

Log files reside in the following subfolder:

/opt/hitachi/storage/logs

When you delete the log folder, the application creates a new one.

The Hitachi Storage Adapter for ODC maintains its logs as follows:

/opt/hitachi/storage/logs/ansible.log

/opt/hitachi/storage/logs/hiodc.log

<table>
<thead>
<tr>
<th>Log File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ansible.log</td>
<td>Adapter log</td>
</tr>
<tr>
<td>hiodc.log</td>
<td>Storage subsystem API log</td>
</tr>
</tbody>
</table>

Specifying the Log Levels in the Configuration File

The ODC storage configuration is here:

path: /opt/hitachi/storage/odc/config directory

filename: log4cplus.property

Configure your root-level logger, and append the following descriptors.

log4cplus.rootLogger= STDOUT, LOG, TRACE_LOG

Log Generation and Rotation

Hitachi Storage Adapter for OEM - Database Cloning log files are created when the adapter server process starts. The log file generation and rotation of the adapter log files have the following features:

- Allow multiple processes to log to the same log file.
- User-specified, size-based log rotation.
- Rotate the current or active log file by incrementing the log file name, by appending consecutive numbers, starting from zero up to a user-specified limit, or a system default.
- Log file names are hiodc.log, hiodc.log.1, hiodc.log.2, and so on.
- The active log file name is hiodc.log.
- Specify the MaxFileSize config value as, for example, MaxFileSize=10. If you include a value outside this range, the default value of 10 is used.
- Specify the maximum number of backup files as, for example, MaxBackupIndex=5. This value can range from 1 to 255. If you include a value outside this range, the default value of 5 is used.
### Messages

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Explanation/Recommended Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>E45000025</td>
<td>Internal error. See <a href="#">Getting Help</a>.</td>
</tr>
<tr>
<td>E450002D</td>
<td>Check to make sure that the file has been correctly installed or had its location correctly set up, then see <a href="#">Getting Help</a> if needed.</td>
</tr>
<tr>
<td>E450000E</td>
<td>Get unit information failed</td>
</tr>
<tr>
<td>E450000E</td>
<td>Get lu list failed</td>
</tr>
<tr>
<td>E450010</td>
<td>Get lu information failed</td>
</tr>
<tr>
<td>E450012</td>
<td>Create lun failed</td>
</tr>
<tr>
<td>E450013</td>
<td>Add to host group failed</td>
</tr>
<tr>
<td>E450014</td>
<td>Create host group failed</td>
</tr>
<tr>
<td>E450015</td>
<td>Rename hostgroup failed</td>
</tr>
<tr>
<td>E450017</td>
<td>Present to host group failed</td>
</tr>
<tr>
<td>E450018</td>
<td>Unpresent to host group failed</td>
</tr>
<tr>
<td>E450019</td>
<td>Remove lu failed</td>
</tr>
<tr>
<td>E45001A</td>
<td>Get all info error</td>
</tr>
<tr>
<td>E45001B</td>
<td>Get pool list failed</td>
</tr>
<tr>
<td>E45001C</td>
<td>Get host group list failed</td>
</tr>
<tr>
<td>E45001D</td>
<td>Create Snapshots failed</td>
</tr>
<tr>
<td>E45001E</td>
<td>Remove snapshots failed</td>
</tr>
<tr>
<td>E45001F</td>
<td>Split snapshots failed</td>
</tr>
<tr>
<td>E450020</td>
<td>Restore snapshots failed</td>
</tr>
<tr>
<td>E450021</td>
<td>Get current snapshots failed</td>
</tr>
<tr>
<td>E450022</td>
<td>Create clones failed</td>
</tr>
<tr>
<td>E450023</td>
<td>Remove clones failed</td>
</tr>
<tr>
<td>E450024</td>
<td>Split clones failed</td>
</tr>
<tr>
<td>E450025</td>
<td>Restore clones failed</td>
</tr>
<tr>
<td>E450026</td>
<td>Get current clones failed</td>
</tr>
<tr>
<td>E450027</td>
<td>Get vvol list failed</td>
</tr>
</tbody>
</table>
Troubleshooting

- Troubleshooting Tips
- Directory Structure
Troubleshooting Tips

Oracle Error

After storage cloning, database cloning fails while starting up the database with the Oracle error:

ORA-01041: internal error. hostdef extension doesn’t exist

According to Oracle, this is a known problem. To overcome the problem, just rerun the script.

Adapter Installation

“build/temp.linux-x86_64-2.7/check_libyaml.c:2:18: fatal error: yaml.h: No such file or directory

#include <yaml.h>

^compilation terminated.”

The above error can occur during adapter rpm installation when it’s not able to locate the file.

This error is a known issue, will not have any impact and can be ignored.

Cloning Tips

See Cloning Tips.

Archive Backup Directory

Make sure that there is sufficient space in the archive log backup directory, or in the fast recovery area if it is configured on source. Otherwise, cloning can fail.

Cloning RAC Tips

If using RAC, make sure that the cluster is up and running.

Also, only node1 is cloned. Add the second instance to node2 and use it as a RAC database.

Naming Tips

- The directory structure in the ASM is retained similarly to the production database. Move files according to your specifications and later rename the corresponding data files and log files.
- The clonedb and instance name should be the same for single-instance setup.
- There should not be any database instance with same name as that of source database or production database on the target system.
The production database name should be less than or equal to eight characters, and the instance name should be less than or equal to nine characters including the instance number.

**Directory Structure**

**Note** ODC does not support multiple clones of the source database on the same target system. Multiple clones on different machines are supported.

The control files and the data files for a cloned database remain the same as that of the source database directory. During cloning, the database is brought up using one control file. You can create more control files if required.

The online logs are freshly created on the cloned database.

According to your needs, you can alter the database file location by using an ASM disk group.

If the script fails before storage cloning, delete the /tmp/odc folder from all the systems (ansible system, clone system and primary system).

1. Recheck the configuration.
2. Run the script.

If the ODC script fails after performing the storage cloning, clean up the LUNs and try again. Follow these steps:

1. Delete the newly-created replication pair.
2. Delete the LUN paths created for secondary volumes.
3. Delete the secondary volumes.
4. Fix the issue or recheck the configuration.
5. Re-run the script.

Make sure the configurations of the database are set correctly. Complete the cloning process in one session. If there is a failure, delete any created directories and database-related processes.

**Deleting Database created using True Copy**

There is not script to delete Database created using True Copy, user need to manually delete the Storage, ASM and Database entities

- Delete the target database
- Delete ASM disk groups used by the target database
- Delete Operating system device paths used by ASM Disk groups
- Unmap the logical units from Host Group
- Delete the logical units used by ASM Disk Groups
Hitachi Data Systems

Corporate Headquarters
2845 Lafayette Street
Santa Clara, California 95050-2627
U.S.A.
Phone: 1 408 970 1000
www.hds.com
info@hds.com

Asia Pacific and Americas
2845 Lafayette Street
Santa Clara, California 95050-2627
U.S.A.
Phone: 1 408 970 1000
info@hds.com

Europe Headquarters
Sefton Park
Stoke Poges
Buckinghamshire SL2 4HD
United Kingdom
Phone: + 44 (0)1753 618000
info.eu@hds.com