Creating clone specifications

SnapManager Oracle

NetApp
November 04, 2025

This PDF was generated from https://docs.netapp.com/us-en/snapmanager-oracle/windows/task_cloning_databases_and_using_custom_plugin_scripts.html on November 04, 2025. Always check docs.netapp.com for the latest.

Table of Contents

Creating clone specifications 1
Clone specification example 5
Cloning databases and using custom plug-in scripts 6

Creating clone specifications

SnapManager for Oracle uses a clone specification XML file, which includes the mappings, options, and parameters for use in the clone operation. SnapManager uses this information to determine where to place the files it clones and how to handle diagnostic information, control files, parameters, and so on.

You can create the clone specification file by using the SnapManager graphical user interface (GUI), command-line interface (CLI), or a text editor.

When you create the clone specification file by using a text editor, you must save it as a .xml file. You can use this XML file for other clone operations.

You can also create a clone specification template and then customize it. You can use the smo clone template command or in the GUI, use the Clone wizard.

SnapManager for Oracle adds a version string to any clone specification template that it generates. SnapManager for Oracle assumes the latest version for any clone specification file that lacks a version string.

If you want to perform remote cloning, do not change the default locations of the data files, redo log files, and control files in the clone specification file. If you change the default location, SnapManager fails to create the clone or creates the clone on a database that does not support Snapshot capability. Therefore, the automatic creation of profile fails.

Though mount point and ASM disk group information are editable from the GUI, you can only change the file name and not the file locations.

You can execute a task multiple times, either with the same or different parameter and value combinations.

1. Open a text file and enter text as shown in the following example:

```
<clone-specification xmlns="http://www.example.com>
     <storage-specification/>
     <database-specification/>
 </clone-specification>
```

2. In the storage specification component, enter the mount points for the data files.

   The storage specification lists the locations for the new storage created for the clone such as data file mount points and raw devices. These items must be mapped from the source to the destination.

   The following example displays the data file mount point syntax that you use in the clone specification:

```
<mountpoint>
     <source>\mnt\path\source_data file_mountpoint</source>
     <destination>\mnt\path\target_data file_mountpoint</destination>
</mountpoint>
```

3. In the database specification component, identify the control file information as a list of the control files that you want created for the clone.

   The database specification specifies the database options for the clone such as control files, redo logs, archive logs, and Oracle parameters.

   The following example displays the control file syntax that you use in clone specification:

```
<controlfiles>
     <file>\mnt\path\clonename\control\control01.ctl</file>
     <file>\mnt\path\clonename\control\control02.ctl</file>
</controlfiles>
```

4. Specify the redo log structure for the clone.

   The following example displays the redo log directory structure for cloning:

```
<redologs>
     <redogroup>
          <file>\mnt\path\clonename\redo\redo01.log</file>
          <number>1</number>
          <size unit="M">100</size>
     </redogroup>
     <redogroup>
         <file>\mnt\path\clonename\redo\redo02.log</file>
         <number>2</number>
         <size unit="M">100</size>
     </redogroup>
</redologs>
```

5. Specify the Oracle parameters that should be set to different values in the cloned database. If you are using Oracle 10, you must specify the following parameters:

   - Background dump
   - Core dump
   - User dump
   - (Optional) Archive logs

   > If the parameter values are not set correctly, the clone operation is stopped and you receive an error message.

   If you do not specify the location where archive logs are stored, SnapManager creates the clone in noarchivelog mode. SnapManager copies this parameter information into the init.ora file of the clone.

   + The following example displays the parameter syntax that you use in clone specification:

   +

```
<parameters>
     <parameter>
          <name>log_archive_dest_1</name>
          <value>LOCATION=\mnt\path\clonename\archive</value>
     </parameter>
</parameters>
```

+ You can use a default value by using a default element within the parameter element. In the following example, the os_authentication_prefix parameter will take the default value because the default element is specified:

+

```
<parameters>
     <parameter>
          <name>os_authent_prefix</name>
          <default></default>
     </parameter>
</parameters>
```

+ You can specify an empty string as the value for a parameter by using an empty element. In the following example, the os_authentication_prefix will be set to an empty string:

+

```
<parameters>
     <parameter>
          <name>os_authent_prefix</name>
          <value></value>
     </parameter>
</parameters>
```

+ NOTE: You can use the value from the source database's init.ora file for the parameter by not specifying any element.

+ If a parameter has multiple values, then you can provide the parameter values separated by commas. For example, if you want to move the data files from one location to another, then you can use the db_file_name_convert parameter and specify the data file paths separated by commas as seen in the following example:

+

```
<parameters>
     <parameter>
          <name>db_file_name_convert</name>
          <value>>\mnt\path\clonename\data
file1,\mnt\path\clonename\data file2</value>
     </parameter>
</parameters>
```

+ If you want to move the log files from one location to another, then you can use the log_file_name_convert parameter and specify the log file paths separated by commas, as seen in the followowing example:

+

```
<parameters>
     <parameter>
          <name>log_file_name_convert</name>

<value>>\mnt\path\clonename\archivle1,\mnt\path\clonename\archivle2</val
ue>
     </parameter>
</parameters>
```

6. Optional: Specify arbitrary SQL statements to execute against the clone when it is online.

   You can use the SQL statements to perform tasks such as re-creating the temp files in the cloned database.

   You must ensure that a semicolon is not included at the end of the SQL statement.

   The following is a sample SQL statement that you execute as part of the clone operation:

```
<sql-statements>
   <sql-statement>
     ALTER TABLESPACE TEMP ADD
     TEMPFILE 'E:\path\clonename\temp_user01.dbf'
     SIZE 41943040 REUSE AUTOEXTEND ON NEXT 655360
     MAXSIZE 32767M
   </sql-statement>
</sql-statements>
```

Clone specification example

The following example displays the clone specification structure, including both the storage and database specification components, for a Windows environment:

```
<clone-specification xmlns="http://www.example.com>

<storage-specification>
    <storage-mapping>
        <mountpoint>
            <source>D:\oracle\<SOURCE SID>_sapdata</source>
            <destination>D:\oracle\<TARGET SID>_sapdata</destination>
        </mountpoint>
    </storage-mapping>
</storage-specification>

<database-specification>
    <controlfiles>
        <file>D:\oracle\<TARGET SID>\origlogA\cntrl\cntrl<TARGET
SID>.dbf</file>
        <file>D:\oracle\<TARGET SID>\origlogB\cntrl\cntrl<TARGET
SID>.dbf</file>
        <file>D:\oracle\<TARGET SID>\sapdata1\cntrl\cntrl<TARGET
SID>.dbf</file>
     </controlfiles>

     <redologs>
        <redogroup>
            <file>D:\oracle\<TARGET SID>\origlogA\log_g11m1.dbf</file>
            <file>D:\oracle\<TARGET SID>\mirrlogA\log_g11m2.dbf</file>
            <number>1</number>
            <size unit="M">100</size>
        </redogroup>
        <redogroup>
            <file>D:\oracle\<TARGET SID>\origlogB\log_g12m1.dbf</file>
            <file>D:\oracle\<TARGET SID>\mirrlogB\log_g12m2.dbf</file>
            <number>2</number>
            <size unit="M">100</size>
        </redogroup>
        <redogroup>
            <file>D:\oracle\<TARGET SID>\origlogA\log_g13m1.dbf</file>
            <file>D:\oracle\<TARGET SID>\mirrlogA\log_g13m2.dbf</file>
            <number>3</number>
            <size unit="M">100</size>
        </redogroup>
        <redogroup>
```

```
            <file>D:\oracle\<TARGET SID>\origlogB\log_g14m1.dbf</file>
            <file>D:\oracle\<TARGET SID>\mirrlogB\log_g14m2.dbf</file>
            <number>4</number>
            <size unit="M">100</size>
        </redogroup>
    </redologs>

    <parameters>
        <parameter>
            <name>log_archive_dest</name>
            <value>LOCATION=>D:\oracle\<TARGET SID>\oraarch</value>
        </parameter>
        <parameter>
            <name>background_dump_dest</name>
            <value>D:\oracle\<TARGET SID>\saptrace\background</value>
        </parameter>
        <parameter>
            <name>core_dump_dest</name>
            <value>D:\oracle\<TARGET SID>\saptrace\background</value>
        </parameter>
        <parameter>
            <name>user_dump_dest</name>
            <value>D:\oracle\<TARGET SID>\saptrace\usertrace</value>
        </parameter>
    </parameters>
   </database-specification>
</clone-specification>
```

Related information

Cloning databases and using custom plug-in scripts

Cloning databases from backups

Cloning databases in the current state

Considerations for cloning a database to an alternate host

Cloning databases and using custom plug-in scripts

SnapManager provides a method for using your custom scripts before and after a clone operation occurs. For example, you might have created a custom script that validates a clone database SID and ensures the SID is allowed by your naming policy. Using the SnapManager clone plug-in, you can include your custom scripts and have them run automatically before or after a SnapManager clone operation.

1. View sample plug-in scripts.

2. Create a script from scratch or modify one of the sample plug-in scripts.

   Create your custom script according to SnapManager plug-in script guidelines.

3. Place your custom script in a specified directory location.
4. Update the clone specification XML file and include information about your custom script that should be used during the cloning process.
5. Using a SnapManager command, verify that the custom scripts are operational.
6. When you initiate the clone operation, include the script name and optional parameters.

Copyright information

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.