Creating task specification file and scripts for SnapManager operations

SnapManager Oracle

NetApp
February 12, 2024

Table of Contents

Creating task specification file and scripts for SnapManager operations

SnapManager for Oracle uses a task specification Extensible Markup Language (XML) file that indicates the pretasks and post-tasks for the backup, restore, and clone operations. You can add the pretask and post-task script names in the XML file for the tasks to be performed before or after the backup, restore, and clone operations.

In SnapManager (3.1 or earlier), you can run the pretask and post-task scripts only for the clone operation. In SnapManager (3.2 or later) for Oracle, you can run the pretask and post-task scripts for the backup, restore, and clone operations.

In SnapManager (3.1 or earlier), the task specification section is part of the clone specification XML file. From SnapManager 3.2 for Oracle, the task specification section is a separate XML file.

SnapManager 3.3 or later does not support the use of the clone specification XML file created in the releases before SnapManager 3.2.

In SnapManager (3.2 or later) for Oracle, you must ensure that the following conditions are met for successful SnapManager operations:

- For backup and restore operations, use the task specification XML file.
- For the clone operation, provide two specification files: a clone specification XML file and a task specification XML file.

  If you want to enable pretask or post-task activity, you can optionally add the task specification XML file.

You can create the task specification file by using the SnapManager graphical user interface (GUI), command-line interface (CLI), or a text editor. You must use a .xml extension for the file to enable appropriate editing features. You might want to save this file so that you can use it for future backup, restore, and clone operations.

The task specification XML file includes two sections:

- The pretasks section includes scripts that could be run before the backup, restore, and clone operations.
- The post-tasks section includes scripts that could be run after the backup, restore, and clone operations.

The values included in the pretasks and post-tasks sections must adhere to the following guidelines:

- Task name: The name of the task must match the name of the script, which is displayed when you run the plugin.sh -describe command.

  If there is a mismatch, then you might receive the following error message: the file not found.

- Parameter name: The name of the parameter must be a string that can be used as an environment variable setting.

  The string must match the parameter name in the custom script, which is displayed when you run the plugin.sh -describe command.

You can create the specification file based on the structure of the following sample task specification file:

```
<task-specification>
  <pre-tasks>
<task>
      <name>name</name>
      <parameter>
          <name>name</name>
          <value>value</value>
       </parameter>
    </task>
  </pre-tasks>
  <post-tasks>
    <task>
      <name>name</name>
      <parameter>
          <name>name</name>
          <value>value</value>
       </parameter>
    </task>
  </post-tasks>
</task-specification>
```

The task specification XML file should not contain any policy.

From the SnapManager GUI, you can set the parameter value and save the XML file. You can use the Task Enabling page of the Backup Create wizard, Restore or Recovery wizard, and Clone Create wizard, to load the existing task specification XML file, and use the selected file for the pretask or post-task activity.

A task can be executed multiple times, either with the same or different parameter and value combinations. For example, you could use a Save task to save multiple files.

SnapManager uses the XML tags provided in the task specification file for the preprocessing or post-processing activity for the backup, restore, and clone operations irrespective of the file extension of the task specification file.

Creating pretask, post-task, and policy scripts

SnapManager enables you to create the scripts for the preprocessing activity, the post-processing activity, and policy tasks of the backup, restore, and clone operations. You must place the scripts in the correct installation directory to execute the preprocessing activity, post-processing activity, and policy tasks of the SnapManager operation.

Pretask and post-task script content

All scripts must include the following:

- Specific operations (check, describe, and execute)
- (Optional) Predefined environment variables
- Specific error handling code (return code (rc))

You must include correct error handling code to validate the script.

You can use the pretask scripts for many purposes, for example, cleaning up a disk space before the SnapManager operation starts. You can also use the post-task scripts, for instance, to estimate whether SnapManager has enough disk space to complete the operation.

Policy task script content

You can execute the policy script without using specific operations such as check, describe, and execute. The script includes the predefined environmental variables (optional) and specific error handling code.

The policy script is executed before the backup, restore, and clone operations.

Supported format

A command file with a .cmd extension can be used as the prescript and post-script.

If you select the shell script file, the SnapManager operation does not respond. To resolve this, you must provide the command file in the plug-in directory, and perform the SnapManager operation again.

Script installation directory

The directory in which you install the script affects how it is used. You can place the scripts in the directory and execute the script before or after the backup, restore, or clone operation takes place. You must place the script in the directory specified in the table and use it on an optional basis when you specify the backup, restore, or clone operation.

You must ensure that the plugins directory has the executable permission before using the scripts for the SnapManager operation.

Activity	Backup	Restore	Clone
Preprocessing	<default_installation_directory>\plugins\backup\create\pre	<default_installation_directory>\plugins\restore\create\pre	<default_installation_directory>\plugins\clone\create\pre
Post-processing	<default_installation_directory>\plugins\backup\create\post	<default_installation_directory>\plugins\restore\create\post	<default_installation_directory>\plugins\clone\create\post
Policy-based	<default_installation_directory>\plugins\backup\create\policy	<default_installation_directory>\plugins\restore\create\policy	<default_installation_directory>\plugins\clone\create\policy

Sample scripts locations

The following are some samples of the pretask and post-task scripts for the backup and clone operations available in the installation directory path:

- <default_installation_directory>\plugins\examples\backup\create\pre
- <default_installation_directory>\plugins\examples\backup\create\post
- <default_installation_directory>\plugins\examples\clone\create\pre
- <default_installation_directory>\plugins\examples\clone\create\post

What you can change in the script

If you are creating a new script, you can change only the describe and execute operations. Each script must contain the following variables: context, timeout, and parameter.

The variables you have described in the describe function of the script must be declared at the start of the script. You can add new parameter values in parameter=() and then use the parameters in the execute function.

Sample script

The following is a sample script with a user-specified return code for estimating the space in the SnapManager host:

```
@echo off
REM $Id:
//depot/prod/capstan/Rcapstan_ganges/src/plugins/windows/examples/clone/cr
eate/policy/validate_sid.cmd#1 $
REM $Revision: #1 $ $Date: 2011/12/06 $
REM
REM

set /a EXIT=0

set name="Validate SID"
set description="Validate SID used on the target system"
set parameter=()

rem reserved system IDs
set INVALID_SIDS=("ADD" "ALL" "AND" "ANY" "ASC" "COM" "DBA" "END" "EPS"
"FOR" "GID" "IBM" "INT" "KEY" "LOG" "MON" "NIX" "NOT" "OFF" "OMS" "RAW"
"ROW" "SAP" "SET" "SGA" "SHG" "SID" "SQL" "SYS" "TMP" "UID" "USR" "VAR")

if /i "%1" == "-check" goto :check
if /i "%1" == "-execute" goto :execute
if /i "%1" == "-describe" goto :describe

:usage:
    echo usage: %0 "{ -check | -describe | -execute }"
    set /a EXIT=99
```

```
    goto :exit

:check
    set /a EXIT=0
    goto :exit

:describe
    echo SM_PI_NAME:%name%
    echo SM_PI_DESCRIPTION:%description%
    set /a EXIT=0
    goto :exit

:execute
    set /a EXIT=0

    rem SM_TARGET_SID must be set
    if "%SM_TARGET_SID%" == "" (
        set /a EXIT=4
        echo SM_TARGET_SID not set
        goto :exit
    )

    rem exactly three alphanumeric characters, with starting with a letter
    echo %SM_TARGET_SID% | findstr "\<[a-zA-Z][a-zA-Z0-9][a-zA-Z0-9]\>"
>nul
    if %ERRORLEVEL% == 1 (
        set /a EXIT=4
        echo SID is defined as a 3 digit value starting with a letter.
[%SM_TARGET_SID%] is not valid.
        goto :exit
    )

    rem not a SAP reserved SID
    echo %INVALID_SIDS% | findstr /i \"%SM_TARGET_SID%\" >nul
    if %ERRORLEVEL% == 0 (
        set /a EXIT=4
        echo SID [%SM_TARGET_SID%] is reserved by SAP
        goto :exit
    )

    goto :exit



:exit
    echo Command complete.
    exit /b %EXIT%
```

Operations in task scripts

The pretask or post-task scripts that you create must follow a standard SnapManager for Oracle plug-in structure.

The pretask and post-task scripts must include the following operations:

- check
- describe
- execute

If any one of these operations is not specified in the pretask or post-task script, then the script becomes invalid.

When you run the smo plugin check command for the pretask or post-task scripts, the returned status of the scripts display error (because the returned status value is not zero).

Operation	Description
check	The SnapManager server runs the plugin.sh -check command to ensure that the system has execution permission on the plug-in scripts. You might also include file permission checking on the remote system.

describe	The SnapManager server runs the plugin.sh -describe command to obtain information about your script and match the elements provided by the specification file. Your plug-in script must contain the following description information:
	
	• SM_PI_NAME: Script name. You must provide a value for this parameter.
	• SM_PI_DESCRIPTION: Description of the script's purpose. You must provide a value for this parameter.
	• SM_PI_CONTEXT: Context in which the script should run-for example, root or oracle. You must provide a value for this parameter.
	• SM_PI_TIMEOUT: The maximum time (in milliseconds) that SnapManager should wait for the script to complete processing and terminate execution. You must provide a value for this parameter.
	• SM_PI_PARAMETER: One or more custom parameters necessary for your plug-in script to perform processing. Each parameter should be listed in a new output line and include the name of the parameter and a description. When the script completes processing, the parameter value will be provided to your script by an environment variable.
	
	The following is the sample output of the Followup_activities script.
	
	`plugin.sh - describe`
	
	`SM_PI_NAME:Followup_activities`
	`SM_PI_DESCRIPTION:this script contains follow-up activities to be executed after the clone create operation.`
	`SM_PI_CONTEXT:root`
	`SM_PI_TIMEOUT:60000`
	`SM_PI_PARAMETER:SCHEMAOWNER:Name of the database schema owner.`
	`Command complete.`

execute	The SnapManager server runs the plugin.sh -execute command to start your script to execute the script.

Variables available in the task scripts for the backup operation

SnapManager provides context information in the form of environment variables related to the backup operation that is being performed. For example, your script can retrieve the name of the original host, the name of the retention policy, and the label of the backup.

The following table lists the environment variables that you can use in your scripts:

Variables	Description	Format
SM_OPERATION_ID	Specifies the ID of the current operation	string
SM_PROFILE_NAME	Specifies the name of the profile used	string
SM_SID	Specifies the system identifier of the database	string
SM_HOST	Specifies the host name of the database	string
SM_OS_USER	Specifies the operating system (OS) owner of the database	string
SM_OS_GROUP	Specifies the OS group of the database	string
SM_BACKUP_TYPE	Specifies the type of the backup (online, offline, or auto)	string
SM_BACKUP_LABEL	Specifies the label of the backup	string
SM_BACKUP_ID	Specifies the ID of the backup	string
SM_BACKUP_RETENTION	Specifies the retention period	string
SM_BACKUP_PROFILE	Specifies the profile used for this backup	string
SM_ALLOW_DATABASE_SHUTDOWN	Specifies if you want to start up or shut down the databaseIf required you can use the -force option from the command-line interface.	boolean
SM_BACKUP_SCOPE	Specifies the scope of the backup (full or partial)	string

SM_TARGET_FILER_NAME	Specifies the target storage system name **Note:** If more than one storage system is used, then the storage system names must be separated by commas.	string
SM_TARGET_VOLUME_NAME	Specifies the target volume name **Note:** The target volume name must be prefixed with storage device name, for example, SM_TARGET_FILER_NAME/SM_TARGET_VOLUME_NAME.	string
SM_HOST_FILE_SYSTEM	Specifies the host file system	string
SM_SNAPSHOT_NAMES	Specifies the Snapshot list **Note:** Name of the Snapshot copies must be prefixed with the storage system name and volume name. Names of the Snapshot copies are separated by commas.	string array
SM_ARCHIVE_LOGS_DIRECTORY	Specifies the archive logs directory **Note:** If the archive logs are located in more than one directory, then the names of those directories are separated by commas.	string array
SM_REDO_LOGS_DIRECTORY	Specifies the redo logs directory **Note:** If the redo logs are located in more than one directory, then the names of those directories are separated by commas.	string array
SM_CONTROL_FILES_DIRECTORY	Specifies the control files directory **Note:** If the control files are located in more than one directory, then the names of those directories are separated by commas.	string array
SM_DATA_FILES_DIRECTORY	Specifies the data files directory **Note:** If the data files are located in more than one directory, then the names of those directories are separated by commas.	string array
user_defined	Specifies additional parameters defined by the user. User-defined parameters are not available for plug-ins that are used as policies.	user-defined

Variables available in the task scripts for the restore operation

SnapManager provides context information in the form of environment variables related to the restore operation that is being performed. For example, your script can retrieve the name of the original host and the label of the backup that is restored.

The following table lists the environment variables that you can use in your scripts:

Variables	Description	Format
SM_OPERATION_ID	Specifies the ID of the current operation	string
SM_PROFILE_NAME	Specifies the name of the profile used	string
SM_HOST	Specifies the host name of the database	string
SM_OS_USER	Specifies the operating system (OS) owner of the database	string
SM_OS_GROUP	Specifies the OS group of the database	string
SM_BACKUP_TYPE	Specifies the type of the backup (online, offline, or auto)	string
SM_BACKUP_LABEL	Specifies the backup label	string
SM_BACKUP_ID	Specifies the backup ID	string
SM_BACKUP_PROFILE	Specifies the profile used for the backup	string
SM_RECOVERY_TYPE	Specifies the recovery configuration information	string
SM_VOLUME_RESTORE_MODE	Specifies the volume restore configuration	string
SM_TARGET_FILER_NAME	Specifies the target storage system name **Note:** If more than one storage system is used, then the storage system names must be separated by commas.	string

Variables	Description	Format
SM_TARGET_VOLUME_NAME	Specifies the target volume name **Note:** The target volume name must be prefixed with storage device name, for example, SM_TARGET_FILER_NAME/SM_TARGET_VOLUME_NAME.	string
SM_HOST_FILE_SYSTEM	Specifies the host file system	string
SM_SNAPSHOT_NAMES	Specifies the Snapshot list **Note:** Name of the Snapshot copies must be prefixed with the storage system name and volume name. Names of the Snapshot copies are separated by commas.	string array
SM_ARCHIVE_LOGS_DIRECTORY	Specifies the archive logs directory **Note:** If the archive logs are located in more than one directory, then the names of those directories are separated by commas.	string array
SM_REDO_LOGS_DIRECTORY	Specifies the redo logs directory **Note:** If the redo logs are located in more than one directory, then the names of those directories are separated by commas.	string array
SM_CONTROL_FILES_DIRECTORY	Specifies the control files directory **Note:** If the control files are located in more than one directory, then the names of those directories are separated by commas.	string array
SM_DATA_FILES_DIRECTORY	Specifies the data files directory **Note:** If the data files are located in more than one directory, then the names of those directories are separated by commas.	string array

Variables available in the task scripts for clone operation

SnapManager provides context information in the form of environment variables related to the clone operation being performed. For example, your script can retrieve the name of the original host, the name of the clone database, and the label of the backup.

The following table lists the environment variables that you can use in your scripts:

Variables	Description	Format
SM_ORIGINAL_SID	SID of the original database	string
SM_ORIGINAL_HOST	Host name associated with the original database	string
SM_ORIGINAL_OS_USER	OS owner of the original database	string
SM_ORIGINAL_OS_GROUP	OS group of the original database	string
SM_TARGET_SID	SID of the clone database	string
SM_TARGET_HOST	Host name associated with the clone database	string
SM_TARGET_OS_USER	OS owner of the clone database	string
SM_TARGET_OS_GROUP	OS group of the clone database	string
SM_TARGET_DB_PORT	Port of the target database	integer
SM_TARGET_GLOBAL_DB_NAME	Global database name of the target database	string
SM_BACKUP_LABEL	Label of the backup used for the clone	string

Error handling in custom scripts

SnapManager processes the custom script based on the specific return codes. For example, if your custom script returns a value of 0, 1, 2, or 3, SnapManager continues with the clone process. The return code also influences how SnapManager processes and returns the standard output of your script execution.

Return code	Description	Continue processing the operation
0	The script completed successfully.	Yes
1	The script completed successfully, with informational messages.	Yes
2	The script completed, but includes warnings	Yes

3	The script fails, but the operation continues.	Yes
4 or >4	The script fails and the operation stops.	No

Viewing sample plug-in scripts

SnapManager includes scripts that you can use as examples for how to make your own scripts or as a basis for your custom scripts.

You can find the sample plug-in scripts at the following location:

- <default_install_directory>\plugins\examples\backup\create
- <default_install_directory>\plugins\examples\clone\create
- <default_install_directory>\plugins\windows\examples\backup\create\post

The directory that contains the sample plug-in scripts includes the following subdirectories:

- policy: Contains scripts that, when configured, always run on the clone operation.
- pre: Contains scripts that, when configured, run before the clone operation.
- post: Contains scripts that, when configured, run after the clone operation.

The following table describes the sample scripts:

Script name	Description	Type of script
validate_sid.sh	Contains additional checks to the SID used on the target system. The script checks that the SID has the following characteristics: • Contains three alphanumeric characters • Begins with a letter	Policy
cleanup.sh	Cleans the target system so that it is ready to store the newly created clone. Preserves or deletes files and directories depending on the need.	Pretask
Mirror_the_backup.cmd	Mirrors the volumes after the backup operation occurs in a Windows environment when you are using either Data ONTAP operating in 7-Mode.	Post-task

Vault_the_backup.cmd	Vaults the qtrees after the backup operation occurs in a Windows environment when you are using either Data ONTAP operating in 7-Mode.	Post-task
Mirror_the_backup_cDOT.cmd	Mirrors the volumes after the backup operation occurs in a Windows environment when you are using clustered Data ONTAP.	Post-task
Vault_the_backup_cDOT.cmd	Vaults the qtrees after the backup operation occurs in a Windows environment when you are using clustered Data ONTAP.	Post-task

Scripts delivered with SnapManager use the BASH shell by default. You must ensure that support for the BASH shell is installed on your operating system before attempting to run any of the sample scripts.

1. To verify that you are using the BASH shell, enter the following command at the command prompt: bash

   If you do not see an error, the BASH shell is operating properly.

   Alternately, you can enter the which-bash command at the command prompt.

2. Locate the script in the following directory:

   <installdir>\plugins\examples\clone\create

3. Open the script in a script editor such as vi.

Sample script

The following sample custom script validates database SID names and prevents invalid names from being used in the cloned database. It includes three operations (check, describe, and execute), which are called after you run the script. The script also includes error message handling with codes 0, 4 and >4.

```
@echo off
REM $Id:
//depot/prod/capstan/Rcapstan_ganges/src/plugins/windows/examples/clone/cr
eate/policy/validate_sid.cmd#1 $
REM $Revision: #1 $ $Date: 2011/12/06 $
REM
REM

set /a EXIT=0

set name="Validate SID"
set description="Validate SID used on the target system"
```

```
set parameter=()

rem reserved system IDs
set INVALID_SIDS=("ADD" "ALL" "AND" "ANY" "ASC" "COM" "DBA" "END" "EPS"
"FOR" "GID" "IBM" "INT" "KEY" "LOG" "MON" "NIX" "NOT" "OFF" "OMS" "RAW"
"ROW" "SAP" "SET" "SGA" "SHG" "SID" "SQL" "SYS" "TMP" "UID" "USR" "VAR")

if /i "%1" == "-check" goto :check
if /i "%1" == "-execute" goto :execute
if /i "%1" == "-describe" goto :describe

:usage:
    echo usage: %0 "{ -check | -describe | -execute }"
    set /a EXIT=99
    goto :exit

:check
    set /a EXIT=0
    goto :exit

:describe
    echo SM_PI_NAME:%name%
    echo SM_PI_DESCRIPTION:%description%
    set /a EXIT=0
    goto :exit

:execute
    set /a EXIT=0

    rem SM_TARGET_SID must be set
    if "%SM_TARGET_SID%" == "" (
        set /a EXIT=4
        echo SM_TARGET_SID not set
        goto :exit
    )

    rem exactly three alphanumeric characters, with starting with a letter
    echo %SM_TARGET_SID% | findstr "\<[a-zA-Z][a-zA-Z0-9][a-zA-Z0-9]\>"
>nul
    if %ERRORLEVEL% == 1 (
        set /a EXIT=4
        echo SID is defined as a 3 digit value starting with a letter.
[%SM_TARGET_SID%] is not valid.
        goto :exit
    )
```

```
    rem not a SAP reserved SID
    echo %INVALID_SIDS% | findstr /i \"%SM_TARGET_SID%\" >nul
    if %ERRORLEVEL% == 0 (
        set /a EXIT=4
        echo SID [%SM_TARGET_SID%] is reserved by SAP
        goto :exit
    )

    goto :exit



:exit
    echo Command complete.
    exit /b %EXIT%
```

Creating task scripts

You can create the pretask, post-task, and policy task scripts for backup, restore, and clone operations, write your script, and include the predefined environment variables in your parameters. You can either create a new script or modify one of the SnapManager sample scripts.

Before you start creating the script, ensure that:

- You must structure the script in a particular manner for it to be run in the context of a SnapManager operation.
- You must create the script based on the expected operations, available input parameters, and return code conventions.
- You must include log messages and redirect the messages to user-defined log files.

1. Create the task script by customizing the sample script.

   Perform the following:

   a. Locate a sample script in the following installation directory:

      <default_install_directory>\plugins\examples\backup\create

      <default_install_directory>\plugins\examples\clone\create

   b. Open the script in your script editor.
   c. Save the script with a different name.
2. Modify the functions, variables, and parameters as needed.
3. Save the script in one of the following directories:

   Backup operations scripts

- <default_install_directory>\plugins\backup\create\pre: Executes the script before the backup operation occurs. Use it optionally when you specify the backup creation.
- <default_install_directory>\plugins\backup\create\post: Executes the script after the backup operation occurs. Use it optionally when you specify the backup creation.
- <default_install_directory>\plugins\backup\create\policy: Always executes the script before the backup operation occurs. SnapManager always uses this script for all the backups in the repository.
Restore operation scripts
- <default_install_directory>\plugins\restore\create\pre: Executes the script before the backup operation occurs. Use it optionally when you specify the backup creation.
- <default_install_directory>\plugins\restore\create\post: Executes the script after the backup operation occurs. Use it optionally when you specify the backup creation.
- <default_install_directory>\plugins\restore\create\policy: Always executes the script before the backup operation occurs. SnapManager always uses this script for all the backups in the repository.
Clone operation scripts
- <default_install_directory>\plugins\clone\create\pre: Executes the script before the backup operation occurs. Use it optionally when you specify the backup creation.
- <default_install_directory>\plugins\clone\create\post: Executes the script after the backup operation occurs. Use it optionally when you specify the backup creation.
- <default_install_directory>\plugins\clone\create\policy: Always executes the script before the backup operation occurs. SnapManager always uses this script for all the backups in the repository.

Storing the task scripts

You must store the pretask, post-task, and policy task scripts in a specified directory on the target server where the backups or clones will be created. For the restore operation, the scripts must be placed in the specified directory on the target server where you want to restore the backup.

1. Create your script.
2. Save the script in one of the following locations:

For the backup operation

Directory	Description
<default_install_directory >\plugins\backup\create\policy	The policy scripts run before the backup operations.
<default_install_directory >\plugins\backup\create\pre	The preprocessing scripts run the before backup operations.
<default_install_directory >\plugins\backup\create\pre	The post-processing scripts run after the backup operations.

For the restore operation

Directory	Description

<default_install_directory >\plugins\restore\create\policy	The policy scripts run before the restore operations.
<default_install_directory >\plugins\restore\create\pre	The preprocessing scripts run before the restore operations.
<default_install_directory >\plugins\restore\create\post	The post-processing scripts run after the restore operations.

For the clone operation

Directory	Description
<default_install_directory >\plugins\clone\create\policy	The policy scripts run before the clone operations.
<default_install_directory >\plugins\clone\create\pre	The preprocessing scripts run before the clone operations.
<default_install_directory >\plugins\clone\create\post	The post-processing scripts run after the clone operations.

Verifying the installation of plug-in scripts

SnapManager enables you to install and use custom scripts to perform various operations. SnapManager provides plugins for the backup, restore, and clone operations, which you can use to automate your custom scripts before and after the backup, restore, and clone operations.

1. Enter the following command:

   smo plugin check -osaccount os db user name

   If you do not provide the -osaccount option, verification of the plug-in script installation happens for the administrator rather than for a specified user.

   The following output indicates that the policy1, pre-plugin1, and pre-plugin2 scripts have been installed successfully. However, the post-plugin1 script is not operational.

```
            smo plugin check
Checking plugin directory structure ...
<installdir>\plugins\clone\policy
   OK: 'policy1' is executable

<installdir>\plugins\clone\pre
   OK: 'pre-plugin1' is executable and returned status 0
   OK: 'pre-plugin2' is executable and returned status 0

<installdir>\plugins\clone\post
   ERROR: 'post-plugin1' is executable and returned status 3
Command complete.
```

Creating a task specification file

You can create the task specification files by using graphical user interface (GUI), command-line interface (CLI), or a text editor. These files are used for performing preprocessing or post-processing activity of the backup, restore, or clone operations.

1. Create a task specification file by using GUI, CLI, or a text editor.

   You can create the specification file based on the structure of the following sample task specification file:

```
<task-specification>
  <pre-tasks>
                <task>
      <name>name</name>
      <parameter>
          <name>name</name>
          <value>value</value>
       </parameter>
    </task>
  </pre-tasks>
  <post-tasks>
    <task>
      <name>name</name>
      <parameter>
          <name>name</name>
          <value>value</value>
       </parameter>
    </task>
  </post-tasks>
</task-specification>
```

2. Enter the script name.
3. Enter the parameter name and the value assigned to the parameter.
4. Save the XML file in the correct installation directory.

Task specification example

```
  <task-specification>
    <pre-tasks>
      <task>
        <name>clone cleanup</name>
        <description>pre tasks for cleaning up the target
system</description>
        </task>
    </pre-tasks>
    <post-tasks>
      <task>
        <name>SystemCopy follow-up activities</name>
        <description>SystemCopy follow-up activities</description>
        <parameter>
          <name>SCHEMAOWNER</name>
          <value>SAMSR3</value>
        </parameter>
      </task>
      <task>
        <name>Oracle Users for OS based DB authentication</name>
        <description>Oracle Users for OS based DB
authentication</description>
        <parameter>
          <name>SCHEMAOWNER</name>
          <value>SAMSR3</value>
        </parameter>
        <parameter>
          <name>ORADBUSR_FILE</name>


<value\>E:\\mnt\\sam\\oradbusr.sql</value\>
        </parameter>
      </task>
      </post-tasks>
  </task-specification>
```

Performing backup, restore, and clone operations using prescript and post-scripts

You can use your own script while initiating a backup, restore, or clone operation. SnapManager displays a Task-enabling page in the Backup Create wizard, Restore or Recover wizard, or Clone Create wizard, where you can select the script and provide values for any parameters required by the script.

- Install the plug-in scripts in the correct SnapManager installation location.
- Verify that the plug-ins are installed correctly by using the smsap plugin check command.
- Ensure that you are using the BASH shell.

In the command-line interface (CLI), list the script name, select the parameters, and set the values.

1. To verify that you are using the BASH shell, enter the following command at the command prompt:bash

   Alternately, you can enter the which-bash command at the prompt, and use the command output as the start parameter of the script.

   The BASH shell is operating properly if you do not see an error.

2. For the backup operation, enter the -taskspec option and provide the absolute path of the task specification XML file for performing a preprocessing or a post-processing activity to occur before or after the backup operation: smo backup create -profile profile_name {[-full {-online | -offline | -auto} [-retain {-hourly | [-daily | -weekly | -monthly | -unlimited}] [-verify] | [-data [[-filesfiles [files]] | [-tablespaces-tablespaces [-tablespaces]] [-datalabellabel] {-online | -offline | -auto} [-retain {-hourly | [-daily | -weekly | -monthly | -unlimited]} [-verify] | [-archivelogs [-labellabel] [-commentcomment] [-backup-destpath1 [,[path2]]] [-exclude-destpath1 [,path2]]] [-prunelogs {-all | -untilSCNuntilSCN | -before {-dateyyyy-MM-dd HH:mm:ss | -months | -days | -weeks | -hours}} -prune-destprune_dest1,[prune_dest2]] [-taskspectaskspec] [-include-with-online-backups | -no-include-with-online-backups]} -dump [-force] [-quiet | -verbose]

   If the backup plug-in operation failed, only the plug-in name and return code are displayed. Your plug-in script must include log messages and redirect the messages to the user-defined log files.

3. For the backup restore operation, enter the -taskspec option and provide the absolute path of the task specification XML file for performing a preprocessing or a post-processing activity to occur before or after the restore operation: smo backup restore -profileprofile_name {-label<label> | -id<id>} {-files<files>|-tablespaces<tablespaces> | -complete | -controlfiles} [-recover {-alllogs | -nologs | -until <until>}][-restorespec<restorespec>] [-taskspec<taskspec>] [-verify][-force] backup restore -fast [require | override | fallback | off] [-preview] -dump [-quiet | -verbose]

   If the restore plug-in operation failed, only the plug-in name and return code are displayed. Your plug-in script must include log messages and redirect the messages to the user-defined log files.

4. For the clone create operation, enter the -taskspec option and provide the absolute path of the task specification XML file for performing a preprocessing or a post-processing activity to occur before or after the clone operation: smo clone create -profileprofile_name {-backup-labelbackup_name | -backup -id<backup-id>| -current} -newsidnew_sid-clonespecfull_path_to_clonespecfile [-reserve<yes, no, inherit>] [-host<host>] [-label<label>] [-comment<comment>] {-taskspec<taskspec>] -dump [-quiet | -verbose]

   If the clone plug-in operation failed, only the plug-in name and return code are displayed. Your plug-in script must include log messages and redirect the messages to the user-defined log files.

Example of creating a backup using the task specification XML file

```
smo backup create -profile SALES1 -full -online -taskspec
sales1_taskspec.xml -force -verify
```

Copyright information

Trademark information