PURESTORAGE®

Oracle Copy Automation Tool

Implementation Guide UNIX

Version 1.5

08-January-2018

0 Introduction

0.1 Abstract

This document describes the steps to install and configure the Oracle CAT software for Oracle systems on UNIX (including Linux).

0.2 License Agreement

The use of the Oracle CAT tool is subject to the `Pure Storage EULA for Plugin / Adaptor / Provider / SDK / Management Pack` license agreement.

The `Plugin EULA` can be found on the `Product End User Page`:
http://www.purestorage.com/legal/productenduserinfo.html

A text version is also included in the software archive.

0.3 Version History

Version	Date	Author	Description
1.0	01-Sep-2017	Expertum	Initial version
1.1	13-Oct-2017	Expertum	Updates to: - prerequisites - configuration file storage_<SID> - conductor and client script: ssh configuration changes
1.2	24-Oct-2017	Expertum	- Added Appendix: SSH setup - Added Oracle prerequisite
1.3	25-Oct-2017	Expertum	- Added sample config files - Added License Agreement
1.4	08-Jan-2018	VLSS	- Changed for Oracle CAT
1.5	16-Jan-2018	VLSS	- Added Purity Run info

PURESTORAGE®

Table of Contents

0 Introduction 2

0.1 Abstract 2

0.2 License Agreement 2

0.3 Version History 2

1 Quick Start 5

1.1 Installation and configuration 5

1.2 Key terms 6

2 Prerequisites 7

2.1 Software versions 7

2.2 Languages 7

2.3 Oracle 8

3 Infrastructure 9

3.1 Hosts 9

3.2 Software directory 9

3.3 Shared area 9

3.4 Security and access 10

4 Setting up Oracle CAT 11

4.1 Software installation 11

4.2 Configuration 12

4.2.1 Overview of configuration files 12

4.2.2 Sequence file 12

4.2.2.1 Function 12

4.2.2.2 Format 13

4.2.3 Configuration file *oraclecopy.flow* 14

4.2.3.1 Function 14

4.2.3.2 Format 14

4.2.3.3 Recovery actions 15

4.2.4 Configuration file oraclecopy.paths 15

4.2.4.1 Function and format 15

4.2.4.2 Embargo entries 16

4.2.5 Configuration file *oraclecopy.properties* 17

4.2.5.1 Function and format 17
4.2.5.2 Default and system-specific properties 17
4.2.5.3 List of properties 17
4.2.6 Configuration file *storage_<SID>* 18
4.2.6.1 Function 18
4.2.6.2 Format 18
4.2.6.3 Example 20
4.2.7 Configuration file *truncate.conf* 20
4.2.7.1 Function 20
4.2.7.2 Format 20
4.2.7.3 Example 21

5 Running Oracle CAT 22

5.1 Conductor and client script 22
5.2 Calling *systemcopy* 23
5.2.1 Which user? 23
5.2.2 Syntax 23

6 Appendix 25

6.1 Password-less SSH setup 25
6.1.1 Generate private and public key 25
6.1.2 Add the public key to the Pure Array 25
6.1.3 Test 26
6.2 Purity RUN setup 26
6.2.1 Packages 26
6.2.2 NFS Server 26
6.2.3 User Setup 27

1 Quick Start

1.1 Installation and configuration

This section contains brief instructions for setting up and running your first system copy.

1. Check that the Oracle database and the operating systems meet the conditions of section 2.1.

2. If the operating system is Linux, make sure that the Korn shell (ksh) is installed (→ 2.2)

3. Determine the directories (software directory and shared directory) that the system copy process will use (→ 3.2 and 3.3)

4. Configure password-free SSH to enable fully automated system copies (→ 3.4)

5. Install the software into the software directory (→ 4.1)

6. Create a sequence file for each copy path. You can use the sequence file delivered with the software as a template. You will probably not need to change the context list, but you will have to adapt the system IDs, the host names and user names and possible the path to the ORACLECOPY script. The sequence file is described in section 4.2.2.

7. Create an `oraclecopy.paths` file (one file for all your copy paths) or use the one that comes with the software as template. The paths file is described in section 4.2.4.

8. Review and where necessary adapt the `oraclecopy.properties` file. The properties that most probably need changing are `dirshare` and `listener_name`. See section 4.2.5 for information about the file and the different properties.

9. Create the `clients.conf` file (one file for all copy paths) or use the file delivered with the software as a template. See **Error! Reference source not found.** for more information.

10. Create the `storage_<SID>` files for all source and target systems involved in the copy processes. These files contain configuration data for Pure Storage. See section 4.2.6.

1.2 Key terms

There are some important terms used throughout this document and also by the software itself:

- A copy path is a system copy identified by a source and target system. For example, if you regularly copy your Oracle production system to an Oracle reporting system and also to an Oracle performance tuning system, there will be two copy paths:
  - Oracle PROD to Oracle RPT
  - Oracle PROD to Oracle PERF

- A context is a major phase in a system copy. An end-to-end copy consists of a series of contexts. Examples of contexts are: pre-copy preparation of target, backup of source system, restore/recovery of target database, post-processing in target system.

- An action is an individual step in a context, such as: exporting tables from the database, stopping Oracle, etc. Each context is made up of a series of actions.

2 Prerequisites

2.1 Software versions

The Oracle database installation and underlying server platform must meet the following prerequisites:

Oracle DBMS	11.2.0.3 and higher
Operating System	Linux All Linux distributions and all versions supported by Oracle according to the certification matrix are supported with Oracle CAT Solaris Solaris 10 and higher AIX (TBD)
Utils	Unix utils: - `id` - `awk` - `sudo`

2.2 Languages

The Oracle CAT code is entirely made up of UNIX shell scripts. There is no compiled (C) code, Java code or code developed in any other scripting language.

The scripting language used is Korn shell (ksh).

NOTE

On Linux hosts, the Korn shell is not always installed by default. If it is not present, you must run the appropriate installation tool (which depends on the Linux distribution used, such as Red Hat, SUSE or others) to install the *ksh* package.

2.3 Oracle

The Oracle CAT script automatically copies the Oracle database data files from the source to the target system. In the process of recovering the target Oracle database, it needs information collected on the source system. This includes also the collected Oracle archive files.

The names of these files are generated based on 2 Oracle parameters:

```
LOG_ARCHIVE_FORMAT      %t_%s_%r.dbf

DB_RECOVERY_FILE_DEST   /u0?/app/oracle/fast_recovery_area
```

This for both the source and target systems. If other naming conventions are used, Oracle CAT may need modifications to capture the correct archive files.

3 Infrastructure

3.1 Hosts

For a given copy path, i.e. a system copy from a specific source system to a specific target system, three host computers are involved:

1. The database server of the source system
2. The Pure Storage flash array
3. The database server of the target system

3.2 Software directory

Oracle CAT requires little space: the actual software (scripts and configuration files) take up less than 2 MB. For non-persistent files (logs, control files, temporary files), a single copy creates about 2 MB of these. It is recommended to remove these files periodically.

The easiest installation option is to install the software in a directory shared by all servers, for example the shared area (see 3.3 below). This facilitates software management by maintaining just a single central copy. However, if for security or other reasons local installations are required, then this is supported as well. With multiple installations, make sure that any change you make to the configuration files or the scripts are always distributed to the other servers with minimum delay.

3.3 Shared area

A copy from a given source to a given target requires a directory shared between the source and target servers (only the Oracle servers, not the flash array). This can be in a file system that is NFS-exported by one server and NFS-imported by the other, or a file system shared by many. The Oracle CAT processes on all servers involved in the copy process read and write files in this directory.

Oracle CAT uses the shared area mainly for two types of files:

1. Non-persistent files shared between the source and target sides during the copy
2. Backups of archive logs of the source database needed for the recovery on the target side

Like the non-persistent files in the software directory (logs, etc.), the non-persistent files created in the shared area are small, typically 1-2 MB per copy. However, the backups of the Oracle archive logs may take up a substantial amount of space, which is difficult to predict. Oracle CAT compresses these backups (GZIP), which reduces their size typically by 70-80%. Nevertheless, we recommend having initially 2-4 GB free space available in the shared area.

Figure 1 below shows the software directory and shared area and the servers accessing it. Note that the flash array accesses neither; all operations on the flash array are initiated from one of the Oracle servers.

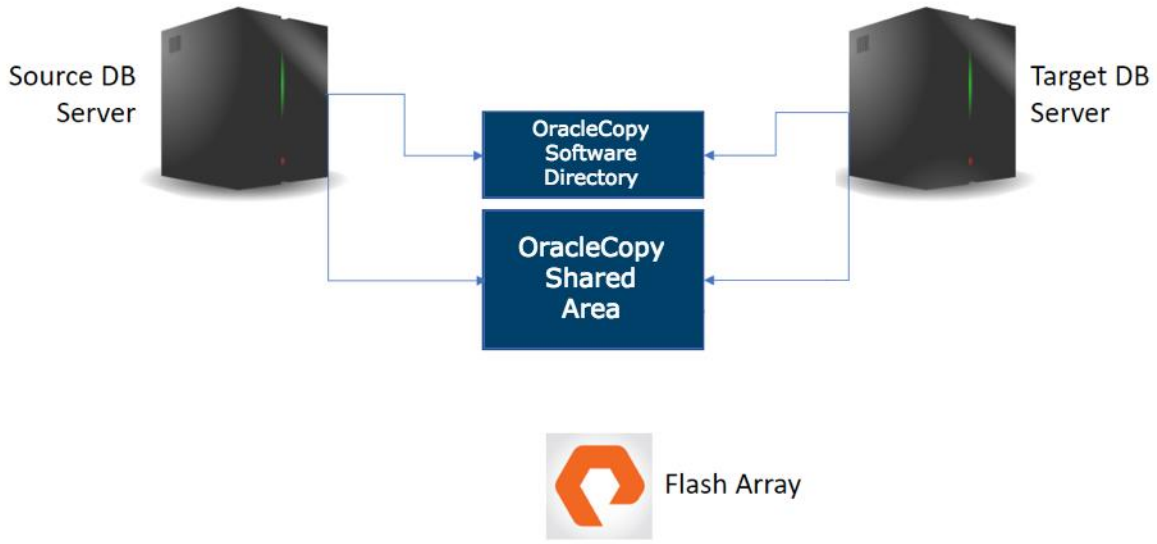

Figure 1: Software directory and shared area

3.4 Security and access

As we will explain later in this guide, the system copy process is controlled by a central "conductor" script, which starts the actions on the different servers. The conductor script connects to the servers via Secure Shell (SSH). To enable fully automatic system copies, it must be possible to use password-free SSH logins from the server/user running the conductor and the other servers.

The conductor can run under any O/S user, as long as this user has write access to both the software directory and shared area. In general, we recommend running the conductor either as `oracle`.

This user must be able to do a password-free SSH login to following servers and users:

Server	User(s)
Source database server	User authorized to connect to Oracle as SYSDBA (`oracle`)
Flash Array	Designated user authorized to execute the snapshot commands
Target database server	`oracle`

4 Setting up Oracle CAT

4.1 Software installation

The application is delivered in the form of a compressed tar file (compressed tar archive, extension tar.gz).

Install as follows:

1. Choose the directory where you want to install Oracle CAT; we denote this directory as **<INSTDIR>** in the following instructions.

   As explained in section 3.2 above, it is best to use a central location shared by all severs involved in the system copy process so that you only have to install and maintain the application once.

2. Log in as `oracle` or as a user who has write access to the place where you will create **<INSTDIR>**
3. Create the installation directory:

```
mkdir <INSTDIR>
```

4. Change to the installation directory:

```
cd <INSTDIR>
```

5. Copy the installation package to the server, e.g. in `/tmp`
6. Decompress and extract the tarball:

```
gunzip -c /tmp/oraclecat-v#.#.tar.gz | tar -xvf -
```

7. Only if you performed the installation as root: you have the option to change the owner and group owner of the software directory. For example, if you installed as root but you want the Oracle DBAs (that is: the members of group `dba`) to have the right to manage Oracle CAT (for example changing configuration files or even adding or modifying scripts), then you can make `dba` the group owner of the software directory and grant write access to that group.

```
(as root)

# On servers with an Oracle system (group 'dba' exists)
chgrp -R dba oraclecat

# Group must have write access (Oracle CAT checks this!)
chmod -R g+w oraclecat
```

4.2 Configuration

Oracle CAT uses several configuration files, which we describe in this section.

All configuration files reside in `<INSTDIR>/etc`.

4.2.1 Overview of configuration files

Table 1 lists the different configuration files used in the current version of Oracle CAT. The first column gives the name of the file in the "etc" directory. The second column contains a brief description. The third column shows whether adapting the configuration file delivered with the software is mandatory or optional. For example, configuration files that reference specific system names (SID) must obviously been adapted to your specific configuration, so modifying them is mandatory. On the other hand, some files (like the list of tables to be truncated in the restored target database) contain information that is valid in most cases; you are still free to change them according to your needs, but you can also use them as they are.

Some files are SID-specific and contain the SID (or the two SIDs of the source and target) in their name. For these files, we use following notation:

<SSID> SID of source system

<TSID> SID of target system

Filename	Description	Change?
`<SSID>_<TSID>.seq`	Sequence file	Mandatory
`oraclecopy.flow`	Flow file (contexts and actions)	Optional
`oraclecopy.paths`	Authorized copy paths	Mandatory
`oraclecopy.properties`	System copy properties	Mandatory
`storage_<SSID>`	Pure Storage configuration data for source system	Mandatory
`storage_<TSID>`	Pure Storage configuration data for target system	Mandatory
`truncate.conf`	Tables to be truncated	Optional

Table 1: Oracle CAT configuration files

4.2.2 Sequence file

4.2.2.1 Function

There is one sequence file for each copy path (= specific source system to specific target system). You are free to name sequence files as you like (you always refer to them explicitly in the command to run the copy), but the recommended location and name is:

```
<INSTDIR>/etc/<SSID>_<TSID>.seq
```

For example, if you have two copy paths, PROD → RPT and PROD → PERF, and your installation directory is `/opt/oraclecat`, then the corresponding sequence files will be called:

```
/opt/oraclecat/etc/prod_rpt.seq
/opt/oraclecat/etc/prod_perf.seq
```

Note: the Oracle CAT tool is case-specific. So if your Oracle SID is prod, use prod instead of PROD.

4.2.2.2 Format

In this example, the copy path is PROD → RPT. The host name of the DB server of PROD is `proddbhost` and the host name of the DB server of RPT is `rptdbhost`.

The context names are defined in the flow file (see `oraclecopy.flow` later).

```
# Sequence file for system copy process using Oracle CAT

# 'sourcesid' and 'targetsid' define the system ID of the source and target
sourcesid prod
targetsid rpt

# 'oraclecopypath' is the full path to the ORACLECOPY script
# This must be the same on all servers involved!
oraclecopypath /opt/oraclecat/bin/ORACLECOPY

# The 'context' entries define the sequence of the copy process for this
source/target combination
# Each entry has the form:
#       context host user ctx
# host : host where this context must run
# user : user ID for the run
# ctx  : name of the context, as defined in the oraclecopy.flow file

context rptdbhost oracle target_prepare
context proddbhost oracle sourcedb_part1
context rptdbhost oracle dbsnapshot
context proddbhost oracle sourcedb_part2
context rptdbhost oracle target_dbrecover
context rptdbhost oracle target_db_postprocess
```

4.2.3 Configuration file *oraclecopy.flow*

4.2.3.1 Function

The purpose of the flow file is simple but crucial: it defines *what* Oracle CAT does and *where* to do it. Specifically, the flow file defines the different contexts and for each context the actions to be performed.

A "context" is simply a phase in the entire system copy process. For example, there will very normally always be a "pre-target" context in which configuration information and optionally data that must be preserved is exported from the target system. You can define as many contexts as you want, as long as joined together they define an end-to-end system copy.

An "action" is an individual activity inside a context. Each context therefore consists of a series of activities carried out in a given sequence. Physically, an action is a shell script.

NOTE

The flow file that comes with the software defines a series of contexts that perform an end-to-end copy for Oracle.

4.2.3.2 Format

The flow file is divided into sections. Each section corresponds to a context. The name of the context, enclosed in square brackets, is the first line of the section.

Inside the section for a given context there are three types of entries:

- The system entry, which indicates whether this context operates on the source system or the target system.
- Several action entries listing the actions to execute for this context. An action is really the name of a shell script, optionally followed by command line arguments.
- Optionally one or more host entries; these indicate for a given Oracle system on which host the context must run

Here is a (theoretical) example:

```
[copytest]
system=source
action=always:"first_action"
action="second_action -open"
action="third_action" recovery="fix_error"
action="last_action"
host=prod:proddbhost
host=rpt:rptdbhost
```

This section defines a context called `copytest`. The context runs on the source side and consists of four actions:

1. The script `first_action` is preceded by the qualifier "always:". This means it is a mandatory action. Mandatory actions are always executed; you do not have the option to skip over them. Such actions typically collect information needed by the following actions in the context; skipping them would not make sense because the later actions then would be unable to do their work.
2. When ORACLECOPY runs the script `second_action` in the `copytest` context, it will pass the command line argument "`-open`" to that script.
3. The action script `third_action` has an associated recovery script (see *Recovery actions* below). If `third_action` fails with a fatal error (exit code >= 64), then ORACLECOPY will invoke the script `fix_error` to undo the effects of the failed action script.
4. The final script to be called is `last_action`. This shows the simplest form of action entry: the action can be skipped (there is no "always:" qualifier), it takes no command line parameters and there is no recovery action.

In this context of the example there are also two "host" entries. These inform ORACLECOPY that for source system PROD the `copytest` context must run on the server `proddbhost`. For source system RPT, it must run on server `rptdbhost`. Host entries are optional: if you omit them ORACLECOPY does not check whether it is running on the correct host.

4.2.3.3 Recovery actions

If an action script runs into a fatal error, it returns an error to the controlling script (ORACLECOPY), which displays information about the error and then stops. In some situations, however, it would not be acceptable to simply stop everything halfway through. In those cases, the driver script will start a recovery action. A practical example is an action script that performs certain actions after shutting down the Oracle database. If this fails and ORACLECOPY stops, then you may want ORACLECOPY to restart Oracle before terminating. In that case, you add the script that starts Oracle as a recovery action for this step in the context.

4.2.4 Configuration file oraclecopy.paths

4.2.4.1 Function and format

This file defines authorized copy paths and optional embargo entries. Authorized copy paths serve as a protection against accidentally copying to the wrong system. For example, if PROD and RPT are both Oracle systems for the HR department and PERF is an Oracle system for the Finance department, then it makes sense to copy PROD to RPT but not to PERF. What you *definitely* wouldn't want to do is copy RPT to PROD or PERF to PROD …

To ensure that only acceptable paths are used, you must explicitly add these paths to the `oraclecopy.paths` file. Authorized copy paths have a simple format, giving just the source and target SID:

```
<source_sid>:<target_sid>
```

An alternative format is:

```
*:<target_sid>
```

The wildcard indicates that `target_sid` may be refreshed from any source. This is not necessarily a good idea (for example it would allow copying PROD to PERF).

4.2.4.2 Embargo entries

Even if copies to a specific system are normally allowed, it is possible that during certain periods you do not want that system to be refreshed. Some examples of this:

- Important training sessions are scheduled over a 6-week period and one of your training systems has been reserved for this. During this time, you don't want this training system to be refreshed.
- A test system is reserved for a critical development project which will take several months. All this time no refreshes from the production system are allowed (as these would destroy all development, customizing and data created for the new project).

You could of course achieve this by deleting or commenting out the copy path, but this is neither an elegant nor a safe method; you would have to remember to remove the copy path on the first day of the critical period and to reinstate it the first day after the end of that period. Instead, you can leave the copy path as it is and create an *embargo entry*. This is a special entry in the same configuration file as the copy paths, which defines periods during which no copies to the specified system are allowed.

Embargo entries take the form:

```
<SID> <fromdate> <todate>
```

The "from" date and "to" date must be specified as YYYYMMDD.

Example: copies from the production system PROD to the test system RPT are normally allowed. However, in the fourth quarter of 2017, RPT will be used for a new project and no refreshes must happen during that time. In the `oraclecopy.paths` configuration file you then have to add an embargo entry:

```
# Normally allow copies PROD -> RPT
prod:rpt

# Embargo: no EEQ refreshes in 4th quarter of 2017:
deny rpt 20171001 20171231
```

The embargo entry consists of the keyword "deny" followed by the SID of the affected system, the start date of the embargo and the end date.

4.2.5 Configuration file *oraclecopy.properties*

4.2.5.1 Function and format

This file contains the properties (parameters) to be used for the system copy.

Properties (parameters) take the form of a simple key-value pair:

```
property_name property_value
```

The parameter name and value are separated by one or more spaces.

4.2.5.2 Default and system-specific properties

The name of a property always has the form:

```
prefix/property
```

Most properties have a default value which can be overridden for a specific system. If no system-specific value exists for a given Oracle system then the default value is used. The default property has the prefix `default` whereas the system-specific property, if defined, has the system ID as prefix.

In the example below, the default directory for sharing data between source and target system (property "dirshare") is `/opt/oraclecopy/share`. However, some systems will use a different share with the path `/u03/share`. All other systems use the default share. In the properties file this will look as follows:

```
default/dirshare /opt/oraclecopy/share
PROD/dirshare /u03/share
```

4.2.5.3 List of properties

The table below does not contain an exhaustive list of all properties, because - like the action scripts - these may differ between environments. The properties listed here are the most important ones for Oracle CAT in Oracle/UNIX configurations.

The two rightmost columns indicate whether a default (cross-system) value can be set (Dflt = X) and whether system-specific settings that override the default are possible (Sys = X).

Name	Meaning	Dflt	Sys
`dirshare`	Full path to shared directory	X	X
`exportdir`	Directory used for exporting data from target database (during pre-processing on target)	X	X
`listener_control`	Stop/start Oracle Listener during target refresh	-	X

	• false: do not stop/start Listener • true: stop/start Listener (name must be in `listener_name`)		
`listener_name`	Name of Oracle Listener to start/stop during target refresh; only needed if property `SID/listener_control` is set to true	-	X
`maxage`	Maximum age (in seconds) of the source copy. If it is older than this, the target process will refuse to use it for the refresh.	X	X
`redosize`	Redo log size for target database. If not specified, ORACLECOPY creates the redo logs of the target with the same size as the source. This option is useful for example when the production database has huge redo logs, which are not especially useful (and increase the recovery time) on the target.	-	X
`signalwait`	Time (in seconds) the target process will wait for completion of the source process	X	X

Table 2: Oracle CAT properties

4.2.6 Configuration file *storage_<SID>*

4.2.6.1 Function

These files contain settings specific for the Pure Storage snapshot process. Each copy process will normally involve two storage configurations:

- File storage_<SSID> for the source system
- File storage_<TSID> for the target system

4.2.6.2 Format

Each record in the storage file has the form

```
type:value
```

Records of a given type may appear more than once in the same file.

Following record types exist:

`flashArray`

Host name or IP address of the Flash Array. This entry must exist both on the source side and on the target side.

flashArrayUser

User ID for logging in to the Flash Array; exists both on the source side and on the target side.

pgroup

Name of the protection group.

For source systems, the protection group may **only** contain the Pure Storage volumes with the database data files. We'll match the volumes snapped with the volumes mapped in the storage file of the target system. If more (or less) volumes are snapped, the copy won't start.

On target systems, a protection group of the whole system can be used to create a snapshot backup before starting the copy procedure.

fs

File system mount path; exists only on the target side.

Only valid for: AIX filesystems

For AIX source systems, this entry should also be created. These will be used to:

1) Gather the AIX LVM storage information.
2) Temporary freeze the file systems for maximum 60 seconds when making the snapshot.

WARNING

On an AIX system, the file system paths are used to derive the physical volumes from the containing volume group (via the logical volume). These are unmounted; varied off and exported on the target system!

Thus, physical volumes may **only** contain logical volumes related to the database data files. Meaning, database data files should be created in a separate volume group; on separate dedicated physical volumes/disks.

End of: AIX filesystems

volmap_<SSID>_<TSID>

Volume mappings from source to target; exists only on the target side.

NOTE

This is the brain of Oracle CAT, here you define which Pure Storage volumes are to be copied from the source to the target machine. As a

prerequisite, the number of Pure Storage volumes on the source side should match the number of volumes on the target side.

Only valid for: AIX filesystems

Oracle CAT will handle the volume management on the AIX target system, including attaching the disks and importing the volume groups on the target system based on the storage information collected on the source system.

If the source system A has more Pure Storage volumes then target system B, it is required to manually create (empty) Pure Storage volumes for the target system B.

The final step is then listing the volume mapping between the volumes of system A and system B in the storage file of the target system.

End of: AIX filesystems

4.2.6.3 Example

Example of a storage file for a source system:

```
flashArray:10.148.25.2
flashArrayUser:pureuser
pgroup:OracleSRCGroup
```

Example of a storage file for a target system:

```
flashArray: 10.148.25.2
flashArrayUser:pureuser
fs:/dev/mapper/ora_redo
fs:/dev/mapper/ora_data1
fs:/dev/mapper/ora_data2
volmap_prod_rpt:OracleDataVol1 OracleCPDataVolume1
volmap_prod_rpt:OracleDataVol2 OracleCPDataVolume2
volmap_prod_rpt:OracleOrigLogA OracleCPOrigLogA
```

4.2.7 Configuration file *truncate.conf*

4.2.7.1 Function

This file defines the sets of tables to be truncated (emptied) in the target system after the refresh.

4.2.7.2 Format

An entry in the truncation list can take the following forms:

```
<schema>:TABLE
```

You can specify truncation of a table in a schema by explicitly entering the schema name.

4.2.7.3 Example

This file lists a set of tables to be truncated in a schema "PERF":

```
# List of tables to truncate after refresh
# Format:
#    <name>:TABLE
#
PERF:COLLECTOR_INFO
PERF:COLLECTOR_HIST
```

5 Running Oracle CAT

5.1 Conductor and client script

To carry out a complete system copy, you run a "conductor" script. This script takes as input a sequence file (see 4.2.2) to determine which contexts (phases) must be executed, in what order and on which server. The conductor is a shell script called **`systemcopy`**.

To run a context, the conductor logs in to the appropriate server via Secure Shell (SSH) and invokes the client script responsible for executing the actions in that context. The complete flow is shown in Figure 2:

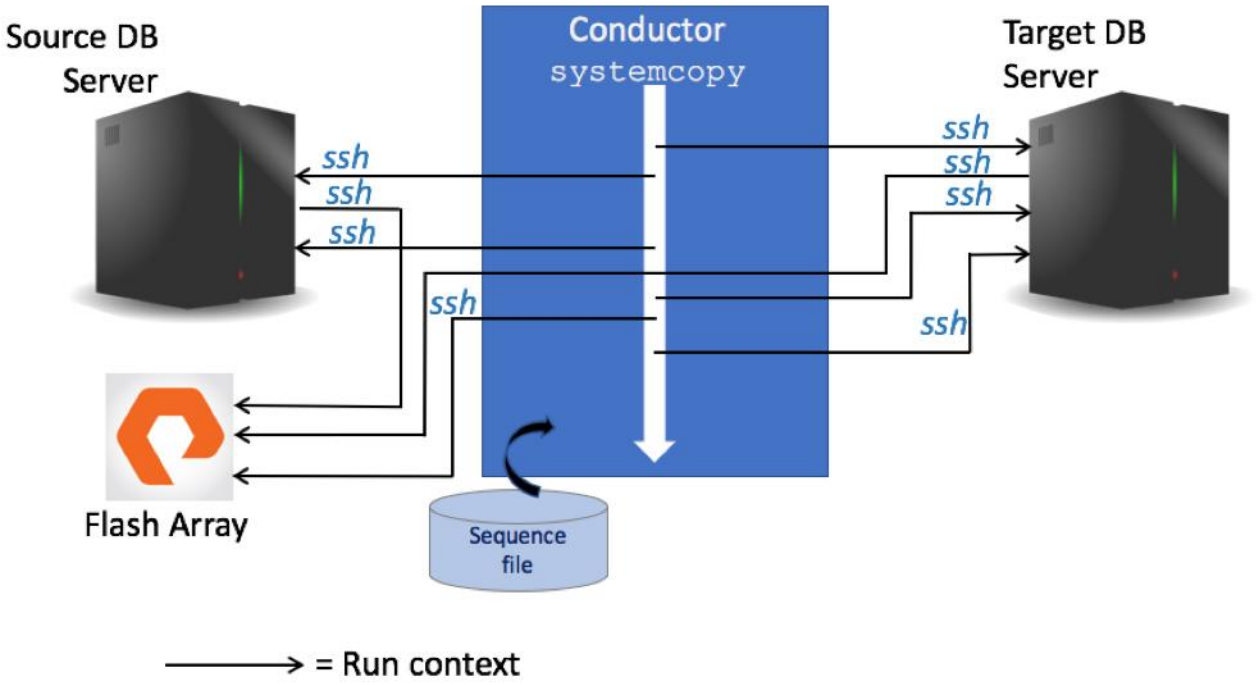

Figure 2: Conductor script ***`systemcopy`***

The client script, which is called `ORACLECOPY`, receives the name of the context as one of its parameters. ORACLECOPY then looks up that context in the flow file (see 4.2.3), where it finds the list of actions that belong to the context. It then executes one action after the other. This is shown schematically in Figure 3: to execute a context `do_xyz`, the conductor launches ORACLECOPY on the remote host. ORACLECOPY reads the action list of `do_xyz` from the flow file and runs the actions (scripts) `action1`, `action2`, `action3`, etc.

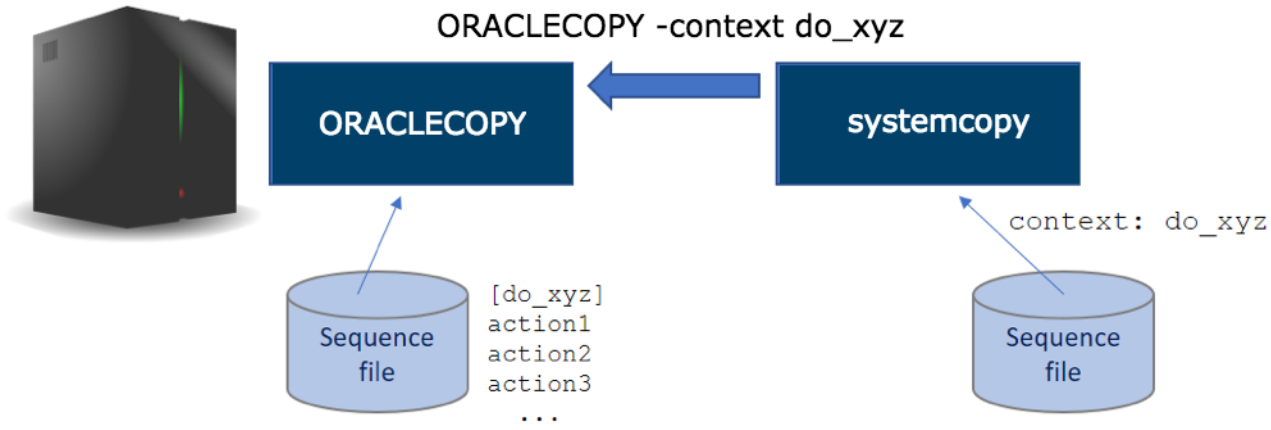

*Figure 3: Client script **ORACLECOPY***

5.2 Calling *systemcopy*

5.2.1 Which user?

You can run `systemcopy` as any user defined on the server, provided this user meets the following conditions:

- Read and write access to the shared directory
- Password-less SSH to all servers involved in the system copy (source DB server, target DB server and Flash Array)
- Member of the `sudo` group

NOTE

For several actions, it is required for the `oracle` user to start the `sudo` command. For example:

- AIX volume handling
- Run as another user: `sudo su – orasid –c "command…"`
- `mount/umount`

5.2.2 Syntax

This is the syntax of the command:

```
systemcopy [ -dialog | -auto | -fullauto ]
           [ -restart | -norestart ]
           seqfile
```

Command arguments:

`-dialog`	Run in dialog (interactive) mode. In this mode, `systemcopy` will prompt you at the beginning of every context and `ORACLECOPY` at the beginning of every action.
`-auto`	Run in automatic mode. In this mode, `systemcopy` will prompt you at the beginning of every context; the context itself, however, will run without interaction.
`-fullauto`	Run in fully automatic mode. No user interaction will happen at all and the system copy will run from beginning to end without human intervention.
`-restart`	If the previous system copy on this path failed, this option instructs `systemcopy` to continue at the point of failure. If the previous copy ended successfully, this option has no effect.
`-norestart`	If the previous system copy on this path failed, this option instructs `systemcopy` to start a fresh copy from the beginning. If the previous copy ended successfully, this option has no effect. Note: if neither option is specified and the previous copy failed, then the behavior is as follows: • Dialog mode: you will be prompted interactively for a decision (continue failed copy or start anew) • Auto / full auto mode: the script will report an error and stop. You will be obliged to specify either `-restart` or `-norestart`.
`seqfile`	Name of the sequence file for this copy path. This is the only mandatory argument. See section 4.2.2 for a description of the sequence file.

6 Appendix

6.1 Password-less SSH setup

To avoid password prompts during the execution of Oracle CAT, we can setup a SSH key. The public and private key will then be used to automatically logon on the different systems.

In this example, we'll setup a user (`oracle`) to automatically login onto the Pure Storage array with the `pureuser` user account.

6.1.1 Generate private and public key

First, if it doesn't exist yet, we'll have to generate a key pair for the user we want to automatically login onto the remote system:

```
oracle> ssh-keygen
```

Accept the default path:

```
Generating public/private key pair.

Enter file in which to save the key (/home/oracle/.ssh/id_rsa):
```

Just press `ENTER` when asked for passwords:

```
Enter passphrase (empty for no passphrase):

Enter same passphrase again:
```

This generates following output:

```
Your identification has been saved in /home/oracle/.ssh/id_rsa.

Your public key has been saved in /home/oracle/.ssh/id_rsa.pub.

…
```

Copy the contens of the public key file to your buffer:

```
oracle> cat /home/oracle/.ssh/id_rsa.pub
```

Select the output and copy it for example with CTRL+C.

6.1.2 Add the public key to the Pure Array

In this step, we'll paste our public SSH key into the GUI of the storage array.

1. Log via the web browser into the storage array with the user `pureuser`.
2. Got to: `System` > `User` > `Me`

3. Click on `Setting` (the gear icon)
4. Select `Update Public Key`
5. Paste the above copied public key. Make sure additional line breaks are removed and click `Save`.

6.1.3 Test

You should be able now to start a ssh session as `ecqadm` to the array with user `pureuser` without a password prompt.

For example:

```
oracle> ssh pureuser@<your array> purevol list
```

6.2 Purity RUN setup

6.2.1 Packages

Purity RUN offers a Linux for Docker version that is running a minimal version of Ubuntu 16.04. The minimal install is missing some packages (AWK and Kourne shell) needed for the Oracle CAT tool to run.

```
$ apt-get install gawk ksh

$ ln -s /usr/bin/gawk /bin/gawk
```

6.2.2 NFS Server

If desired, the Purity RUN app can serve as the NFS server for the Oracle CAT tool. First, the NFS server packages need to be installed:

```
$ apt-get install nfs-kernel-server
```

Then we need to create some directories and mounts for the NFS server to export:

```
$ mkdir -p /opt/oraclecopy/share

$ mkdir -p /export/oraclecopy

$ mount --bind /opt/oraclecopy /export/oraclecopy
```

Edit the /etc/exports file to add the new directory as an NFS export:

```
/export/oraclecopy      *(rw,sync,no_root_squash,no_subtree_check)
```

Then restart the NFS server:

```
$ systemctl start nfs-server.service
```

6.2.3 User Setup

The `root` user can be used for the Oracle CAT tool, but it is not advisable to do so for security concerns. Instead, we recommend creating a new user, usually `oracle`, to connect to the various hosts and run the Oracle CAT tool.

```
$ groupadd -g 54321 oinstall

$ groupadd -g 54322 dba

$ useradd -g 54321 -G dba -u 54321 oracle

$ passwd oracle

$ chown oracle:oinstall -R /opt/oraclecopy
```

For the Oracle CAT tool to run seamlessly, the user needs to have passwordless sudo privileges. Run `visudo` and add the following line:

```
oracle      ALL=(ALL) NOPASSWD: ALL
```