

ORAAH 2.7.0 Installation Guide

Contents

1	Introduction.	1
2	Hadoop cluster setup.	1
3	First step with the ORAAH downloads.	1
4	Automatic installation on BDA clusters.	1
5	Automatic installation on non-BDA clusters.	2
6	Supporting scripts.	3
7	Manual installation.	4
8	Installing with non-Oracle's R distribution.	5
9	Other dependencies.	5
10	ORAAH important post-install configuration.	6
10.1	ORCH_HADOOP_HOME	7
10.2	ORCH_HAL_VERSION	7
10.3	ORCH_JAR_MR_VERSION	8
10.4	ORCH_STREAMING_LIB	8
11	<i>Renviron.site</i> file in \$R_HOME/etc.	8
12	Appendix 1: Examples of <i>Renviron.site</i>.	9
12.1	Cloudera Distribution of Hadoop 5.8.0.	9
12.2	Hortonworks Distribution of Hadoop 2.4.0.0-169.	10
12.3	Oracle's Big Data Lite Virtual Machine 4.5.0.	11
13	Copyright Notice	13

1 Introduction.

The following set of instructions will guide an administrator on the installation of the Oracle R Advanced Analytics for Hadoop on a Client Node (where R will run) and on the Hadoop nodes. These steps are currently validated on generic Hadoop, Cloudera and on Hortonworks clusters.

2 Hadoop cluster setup.

Until the ORAAH 2.2.1 release (formely ORCH), the installation of ORAAH (and the dependent ORE) packages was only required on the client node from where the R user interacts with the Hadoop deployment. In particular, there was no installation requirement on the actual cluster nodes of Hadoop. Starting from release 2.4.0, the installation on all Hadoop nodes that have YARN Node Manager running is required, and the current release 2.7.0 now supports the ORE release 1.5. The previous dependency on the OREmodels package was replaced with a dependency on the OREcommon package. Altogether ORAAH depends on 4 ORE packages that need to be installed on a client node and on the Hadoop nodes:

- OREbase
- OREcommon
- OREserver
- OREstats

In addition ORAAH now depends on Intel MKL libraries that are used in ORAAH advanced statistical functions to improve performance and precision of statistical computations. Intel MKL libraries need to be installed on the client node and every Hadoop node that has YARN Node Manager running. A full set of required libraries is included with the product distribution and will be automatically installed when using the automatic install scripts.

3 First step with the ORAAH downloads.

The first step is to Download the ORAAH release 2.7.0 and the Supporting packages from the Oracle Technology Network site: <http://www.oracle.com/technetwork/database/database-technologies/bdc/r-advanalytics-for-hadoop/downloads/index.html>

You must unzip both files on a folder in the node/machine that will be used as client, and eventually do the same for the Cluster nodes (please refer to instructions below).

4 Automatic installation on BDA clusters.

ORAAH includes a set of installation and uninstallation scripts that automate the administration of the installation, upgrade and uninstallation of the product. Note that automatic installation scripts for the Hadoop nodes will work on Oracle's Big Data Appliance (BDA) only; for installation steps on non-BDA Hadoop clusters please refer to the section "[Automatic installation on non-BDA clusters](#)".

Under the "ORAAH-2.7.0-install" folder, the following scripts can be found:

- **install-client.sh.** Installs the client side packages and libraries required to run ORAAH platform. The script must be run only on Hadoop cluster edge nodes and/or on a client hosts that are outside of the cluster that will be used for running R and using ORAAH. This script is not required to be run on Hadoop cluster compute nodes. The only exception is when Hadoop compute node is also used as an edge node for running ORAAH client software. Available options:
 - "-y" - automatically reply "yes" to all script questions, unattended installation mode.
- **uninstall-client.sh.** Removes all client side ORAAH packages and libraries. Available options:

- "-y" - automatically reply "yes" to all script questions, unattended installation mode.
 - "-f" - force to continue uninstallation even if errors are encountered.
- **install-server.sh.** Installs server side packages and libraries required to run ORAAH workloads on every compute node of Hadoop cluster, i.e. nodes that are managed by Hadoop's nodemanager. This script should be used only on Oracle's BDA Hadoop cluster since it requires the "dcli" tool to be available and configured. The script must be run only on one of the Hadoop cluster nodes, as it will automatically install all the required components on the rest of the BDA cluster. Available options:
 - "-y" - automatically reply "yes" to all script questions, unattended installation mode.
 - *<filename>* - ignored in BDA installation mode.
 - **uninstall-server.sh.** Removes server side ORAAH packages and libraries from every node of the Hadoop cluster. This script can be used only on Oracle's BDA Hadoop cluster and requires the "dcli" tool to be available and configured. The script must be run only on one of the Hadoop cluster nodes, as it will automatically uninstall the components on the rest of the BDA. Available options:
 - "-y" - automatically reply "yes" to all script questions, unattended installation mode.
 - "-f" - force to continue uninstallation even if errors are encountered.
 - *<filename>* - ignored in BDA installation mode.

Note

Note that if you are using one of the Hadoop cluster nodes as a client node and therefore run both "install-client.sh" and "install-server.sh" on this node, then when you are uninstalling ORAAH, the second uninstall script might fail because some of the ORAAH packages and libraries will be removed by a first uninstall script. For instance if you first run "uninstall-server.sh", then "uninstall-client.sh" might fail because it can't find ORE packages and MKL libraries which were already removed. To overcome this issue and finish uninstallation use "-f" option in your second uninstall script, i.e. "uninstall-client.sh -f".

Important



If you are installing ORAAH together with Oracle R Enterprise (ORE) then you need to make sure that ORAAH is installed **after** ORE. If you install ORE after ORAAH it will override some of the shared R packages with outdated versions that are not compatible with ORAAH causing ORAAH Hive transparency layer and some of the shared analytics functionality to misbehave and will cause runtime error. Starting from ORAAH 2.7.0 release we are validating versions of loaded packages to make sure that it was not overridden and will generate a fatal error during ORAAH loading sequence if any version mismatch or incompatibility is detected.

5 Automatic installation on non-BDA clusters.

ORAAH includes a set of installation and uninstallation scripts that automate management of installation, upgrade and uninstallation of the product on non-BDA clusters as well. For automated installation on Oracle's Big Data Appliance (BDA) clusters please refer to the section "Automatic installation on BDA clusters".

Under the "ORAAH-2.7.0-install" folder, the following scripts can be found:

- **install-client.sh.** Installs the client side packages and libraries required to run ORAAH platform. The script must be run only on Hadoop cluster edge nodes and/or on a client hosts that are outside of the cluster that will be used for running R and using ORAAH. This script is not required to be run on Hadoop cluster compute nodes. The only exception is when Hadoop compute node is also used as an edge node for running ORAAH client software. Available options:
 - "-y" - automatically reply "yes" to all script questions, unattended installation mode.
 - **uninstall-client.sh.** Removes all client side ORAAH packages and libraries. Available options:
-

- "-y" - automatically reply "yes" to all script questions, unattended installation mode.
 - "-f" - force to continue uninstallation even if errors are encountered.
- **install-server.sh.** Installs server side packages and libraries required to run ORAAH workloads on every compute node of Hadoop cluster, i.e. nodes that are managed by Hadoop's nodemanager. When this script is executed on a non Oracle's BDA Hadoop cluster, it will use "rsync" and "ssh" tools to distribute packages across the cluster and execute remote commands. The script must be run only once on one of the Hadoop cluster nodes. Note that the scripts expects that password-less ssh access to the cluster nodes is enabled. For the user's convenience a "extra/keyless-ssh.sh" script is included with ORAAH installation scripts to simplify passwordless ssh setup. Available options:
 - "-y" - automatically reply "yes" to all script questions, unattended installation mode.
 - *<filename>* - a plain text file that contains host names of all Hadoop cluster nodes where server-side components must be installed. The file format is one host name per line terminated by a new line symbol, no commas or special characters should be used. Note that all nodes that are running YARN Node Manager must be listed in this file or ORAAH jobs may fail randomly (a Job might eventually hit a node without the proper configuration).
 - **uninstall-server.sh.** Removes server side ORAAH packages and libraries from every node of Hadoop cluster. When this script is executed on a non Oracle's BDA Hadoop cluster, it will use "rsync" and "ssh" tools to distribute packages across the cluster and execute remote commands. The script must be run only once on one of the Hadoop cluster nodes. Note that the scripts expects that passwordless ssh access to the cluster nodes is enabled. For the user's convenience a "extra/keyless-ssh.sh" script is included with ORAAH installation scripts to simplify password-less ssh setup. Available options:
 - "-y" - automatically reply "yes" to all script questions, unattended installation mode.
 - "-f" - force to continue uninstallation even if errors are encountered.
 - *<filename>* - a plain text file that contains host names of all Hadoop cluster nodes where server-side components must be installed. The file format is one host name per line terminated by a new line symbol, no commas or special characters should be used. Note that all nodes that are running YARN Node Manager must be listed in this file for a complete uninstallation.

Note

Note that if you are using one of the Hadoop cluster nodes as a client node and therefore run both "install-client.sh" and "install-server.sh" on this node, then when you are uninstalling ORAAH, the second uninstall script might fail because some of the ORAAH packages and libraries will be removed by a first uninstall script. For instance if you first run "uninstall-server.sh", then "uninstall-client.sh" might fail because it can't find ORE packages and MKL libraries which were already removed. To overcome this issue and finish uninstallation use "-f" option in your second uninstall script, i.e. "uninstall-client.sh -f".

Important

If you are installing ORAAH together with Oracle R Enterprise (ORE) then you need to make sure that ORAAH is installed **after** ORE. If you install ORE after ORAAH it will override some of the shared R packages with outdated versions that are not compatible with ORAAH causing ORAAH Hive transparency layer and some of the shared analytics functionality to misbehave and will cause runtime error. Starting from ORAAH 2.7.0 release we are validating versions of loaded packages to make sure that it was not overridden and will generate a fatal error during ORAAH loading sequence if any version mismatch or incompatibility is detected.

6 Supporting scripts.

ORAAH comes with a set of supporting installation and management scripts that can help an ORAAH administrator automate some common supporting tasks that may be requested by ORAAH users.

- **install-packages.sh.** Installs a set of R packages on every BDA Hadoop cluster node in bulk. This script can be used only on Oracle's BDA Hadoop cluster and requires the "dcli" tool to be available and configured. The script must be run only once on one of the Hadoop cluster nodes. Available options:
-

- "-y" - automatically reply "yes" to all script questions, unattended installation mode.
- `<filename>` - list of R packages to be installed. The file format is one file path per line terminated by a new line symbol, no commas or special characters should be used.
- **uninstall-packages.sh.** Uninstalls a set of R packages on every BDA Hadoop cluster node in bulk. This script can be used only on Oracle's BDA Hadoop cluster and requires the "dcli" tool to be available and configured. The script must be run only once on one of the Hadoop cluster nodes. Available options:
 - "-f" - force to continue uninstallation even if errors are encountered.
 - `<filename>` - list of R packages to be uninstalled. The file format is one file path per line terminated by a new line symbol, no commas or special characters should be used.
- **keyless-ssh.sh.** This script can be used to enable password-less ssh for a set of Hadoop nodes in bulk. Password-less ssh is currently required for automated installation and uninstallation of ORAAH server-side components on non-BDA Hadoop clusters.
 - `<filename>` - a plain text file that contains host names of all Hadoop cluster nodes where server-side components must be installed. The file format is one host name per line terminated by a new line symbol, no commas or special characters should be used.

7 Manual installation.

In case you are installing ORAAH on non-BDA Hadoop cluster and "dcli" tool is not available, automatic server side installation/uninstallation scripts may fail for various reasons. The following steps describe how to manually deploy all server-side components on any Hadoop cluster. Client side installation/uninstallation scripts can still be used. To manually install ORAAH on your Hadoop cluster follow the steps described below.

Copy the following files to every node of your Hadoop cluster:

```
OREserver_1.5_R_x86_64-unknown-linux-gnu.tar.gz
OREcommon_1.5_R_x86_64-unknown-linux-gnu.tar.gz
mkl/*
lib/*
```

Install R packages on every node of the cluster:

```
R --vanilla CMD INSTALL OREcommon_1.5_R_x86_64-unknown-linux-gnu.tar.gz
R --vanilla CMD INSTALL OREserver_1.5_R_x86_64-unknown-linux-gnu.tar.gz
```

Copy MKL libraries to R's library directory:

```
cp mkl/* /usr/lib64/R/lib
```

Copy ORAAH libraries to R's library directory:

```
cp lib/* /usr/lib64/R/lib
```

If the OREserver and OREcommon packages have NOT been installed on any of the cluster nodes where a mapper/reducer tasks could run, then 2 of the analytics, namely, "orch.glm" and "orch.lm" will error out. The following error can be seen in the mapper task log file:

```
Error in loadNamespace(name) : there is no package called 'OREserver':
Calls: source ... tryCatch -> tryCatchList -> tryCatchOne -> <Anonymous>
In addition: Warning message:
In library(package, lib.loc = lib.loc, character.only = TRUE, logical.return = TRUE, :
there is no package called 'OREserver'
Execution halted
```

8 Installing with non-Oracle's R distribution.

In case you choose to do not use Oracle's R distribution you will need to install one additional library *libOrdBlasLoader.so* to enable ORAAH statistical packages to function correctly. If the library is not installed then you will encounter failures when running most of ORAAH analytics.

If you use Oracle's R distribution (available via <http://public-yum.oracle.com/> for OEL5 and OEL6) the library is already included with the distribution and no addition actions are required.

To install the library, copy the library file *libOrdBlasLoader.so*, which is included in ORAAH distribution zip file, into one of the directories listed in `$LD_LIBRARY_PATH` or (the preferred way) into R's "lib" directory (which by default is `/usr/lib64/R/lib`).

```
cp libOrdBlasLoader.so /usr/lib64/R/lib
```

The same should be done on every node of Hadoop cluster where R is installed:

```
dcli -C -f libOrdBlasLoader.so -d /tmp/ libOrdBlasLoader.so
dcli -C "cp /tmp/libOrdBlasLoader.so /usr/lib64/R/lib"
```

Note

Note that you need root privileges to install this library.

9 Other dependencies.

ORAAH depends on the "rJava" and the "RJDBC" packages to work. Both packages come in the "ORAAH-2.7.0-Supporting" zip file and need to be installed into the client's host only. It should be installed by the installation scripts, but if the scripts are not executed, or you have encountered a failure while running the automated install scripts, it's required that you install them manually. You can also download the packages from CRAN:

- rJava: <https://cran.r-project.org/web/packages/rJava/index.html>
- RJDBC: <https://cran.r-project.org/web/packages/RJDBC/index.html>

And install it on the client's host only:

```
R CMD javareconf
R CMD INSTALL rJava_0.9-8.tar.gz
R CMD INSTALL RJDBC_0.2-5.tar.gz
```

ORAAH also has indirect dependency on "png" and "DBI" package. Same as with required dependencies, the packages are automatically installed from "ORAAH-2.7.0-Supporting" zip file or you can install them manually. You can also download the packages from CRAN:

- png: <http://cran.r-project.org/web/packages/png/index.html>
- DBI: <https://cran.r-project.org/web/packages/DBI/index.html>

And install it on the client's host only:

```
R CMD INSTALL png_0.1-6.tar.gz
```

10 ORAAH important post-install configuration.

ORAAH can be used with a wide variety of Hadoop versions and distribution but to operate correctly ORAAH needs to "know" how to interact with the particular Hadoop version you are using.



Important

Attention. It is critical that all the following configuration steps are done for the proper operation of ORAAH on the cluster for HDFS, HIVE and Spark. Even if `library(ORCH)` loads successfully, this does not mean the end of the configuration since there are important HIVE and Spark requirements below, so please follow along.

There is an environment variable that will override any Hadoop's native environment values that specify Hadoop's component home path. For example if both `HADOOP_HOME` and `ORCH_HADOOP_HOME` environment variables are set ORAAH will prioritize the use of the `ORCH_HADOOP_HOME` variable.

ORAAH can be automatically configured for its Hadoop Abstraction Layer (HIVE and Spark require more manual steps included in a later portion of this document) with most Cloudera Distributions of Hadoop, but in case you are using another type of distribution (Hortonworks, MapR, Apache) or have built your own Hadoop environment from separate components, you most likely will encounter the following error at ORAAH startup:

```
Oracle R Connector for Hadoop 2.7.0
Info: using native C base64 encoding implementation
Info: Hadoop distribution is unknown
Error: unsupported version 2.2.0-cdh5.0.0-beta-2 of Hadoop
Info: use "ORCH_HAL_VERSION" envvar to define HAL version
DBG: 22:41:22 [FA] HAL was not initialized
Error : .onLoad failed in loadNamespace() for 'ORCHcore', details:
call: NULL
error: execution aborted
Error: package 'â' could not be loaded
```

or:

```
Oracle R Connector for Hadoop 2.7.0
Info: using native C base64 encoding implementation
Info: Hadoop distribution is Cloudera's CDH v5.0.0
Info: using auto-detected ORCH HAL v4.2
Info: HDFS workdir is set to "/user/oracle"
Error: unrecognized response from Hadoop
Error: mapReduce is not ready, hadoop.*() may fail
Info: HDFS is functional
Error: Failed to connect to Hadoop cluster.
Loading required package: ORCHstats
<...>
```

This indicates that ORAAH does not know how to interact this particular Hadoop distribution and has to be manually configured using special OS environment variables.

In order to preserve the configuration between R sessions there are many options. The preferred one is to put the configuration variables into `/usr/lib64/R/etc/Renviron.site` as `root`. This file does not exist by default in R (do not confuse it with the existing `/usr/lib64/R/etc/Renviron`), but it will allow any R session to pickup environmental variables from it.

Other ways include putting these variables into the end user's `.cshrc` (or `.bashrc`, or `.profile` depending on the OS and shell interpreter) shell startup scripts, create a user's `.Rprofile` file that will be loaded every time they start R, or use `Sys.setenv()` R base functions during an R session to configure the variables before loading ORAAH.

Listed below are the most important ORAAH configuration variables. For the complete list of ORAAH configuration options use `help("ORCH-config")` after loading `ORCHcore` package.

10.1 ORCH_HADOOP_HOME

You can set this ORAAH environment variable before starting R and loading the ORCH library. It allows you to override auto-detection of a Hadoop's home path or home path to other Hadoop's components used by ORAAH.



Important

Attention. Setting this environment variable will override any Hadoop's native environment values that specify Hadoop's component home path. For example if both `HADOOP_HOME` and `ORCH_HADOOP_HOME` environment variables are set ORAAH will prioritize the use of the `ORCH_HADOOP_HOME` variable.

Other supported Hadoop components home environment variables are:

- `ORCH_HDFS_HOME`
- `ORCH_HIVE_HOME`
- `ORCH_MAHOUT_HOME`
- `ORCH_SQOOP_HOME`
- `ORCH_OLH_HOME`

10.2 ORCH_HAL_VERSION

You can set this ORAAH environment variable before starting R and loading the ORCH library. It enables you to override the auto-detection of a Hadoop version and to specify the use of an exact version of the ORAAH Hadoop Abstraction Layer.

Supported versions are:

- **1** = Apache/IDC/Hortonworks 1.*
- **2** = Cloudera CDH3u*
- **3** = Cloudera CDH4.* with MR1
- **4** = Cloudera CDH4.[0-3] with MR1
- **4.1** = Cloudera CDH4.4 with MR1
- **4.2** = Cloudera CDH5.x with MR2 or Hortonworks 2.x

If ORAAH auto-detection cannot identify the Hadoop version, a message indicating that `ORCH_HAL_VERSION` is used will be shown to the user upon loading of the ORCH library. If ORAAH auto-detection can identify the Hadoop version and it is not consistent with the one specified by `ORCH_HAL_VERSION` then a warning message will be issued upon loading of the ORCH library and the version specified by `ORCH_HAL_VERSION` will be used instead.

If `ORCH_HAL_VERSION` is not set (the default), then ORAAH uses Hadoop version auto-detection. If it cannot identify the Hadoop distribution or version, then ORAAH issues an error message and remains in an error state (not initialized). This state prevents HDFS and mapReduce operations from functioning correctly. You must unload ORAAH, set the correct value of `ORCH_VAL_VERSION`, and reload ORAAH.

Note

If `ORCH_HAL_VERSION` is set to an invalid value, then an error message is issued when loading ORAAH and the value is ignored. ORAAH will continue to operate as if the variable was not set. You can unload ORAAH, set the correct value of `ORCH_VAL_VERSION`, and reload ORAAH in order to correct this.

Note

You can override the HAL version when you are testing ORAAH against a new Hadoop distribution. In this case, ORAAH loads and initializes, but you may encounter failures when invoking ORAAH API functions. ORAAH does not provide any functional guarantees as this case.

10.3 ORCH_JAR_MR_VERSION

You can set this ORAAH environment variable before starting R and loading the ORCH library. It allows you to override auto-detection of a Hadoop mapReduce API version and to specify the use of the appropriate version of the ORAAH Hadoop JAR library.

Supported versions are:

- **1** = mapReducer v1.
- **2** = mapReducer v2, aka YARN.

If ORAAH can not auto-detect the Hadoop version and HAL then mapReduce version will default to version 2.

10.4 ORCH_STREAMING_LIB

You can set this ORAAH environment variable before starting R and loading the ORCH library. It allows you to override auto-detection of a Hadoop's Streaming Java library path and to specify custom path to the streaming JAR file. The path should be specified including the library file name.



Important

Attention. Setting this environment variable will override any Hadoop's native environment values that specify streaming library path. For example if both `HADOOP_STREAMING_JAR` and `ORCH_STREAMING_LIB` environment variables are set ORAAH will opt in to use its own `ORCH_HADOOP_HOME` variable.

11 *Renviron.site* file in `$R_HOME/etc`.

R loads environment variables from a several different files before the R Session is started, and for a server with the expected multi-user sessions connecting to R, it is suggested that the file *Renviron.site* is created and be used.

R searches for site and user files to process for setting environment variables. The name of the site file is the one pointed to by the environment variable `R_ENVIRON`; if this is unset, `$R_HOME/etc/Renviron.site` is used (if it exists, which it does not in a "factory-fresh" installation). The name of the user file can be specified by the `R_ENVIRON_USER` environment variable; if this is unset, the files searched for are `.Renviron` in the current or in the user's home directory (in that order).

There is also a file `$R_HOME/etc/Renviron` which is read very early in the start-up processing. It contains environment variables set by R in the configure process. Values in that file can be overridden in site or user environment files: do not change `$R_HOME/etc/Renviron` itself. Note that this is distinct from `$R_HOME/etc/Renviron.site`.

The recommendation is to then create a new *Renviron.site* file with settings that are related to ORAAH, its HIVE and Spark configuration requirements, and other environmental variables that might be needed if using ORAAH on a mixed Oracle Database environment (or when an Oracle Database Client it also configured in the same Node).

This file should be put into `$R_HOME/etc`, or by default `/usr/lib64/R/etc/Renviron.site`.

The following examples show sample *Renviron.site* files for a Cloudera Distribution of Hadoop cluster, a Hortonworks Distribution of Hadoop Cluster, and Oracle's Big Data Lite Virtual machine. In the case of the Cloudera cluster, the installation is assuming that Parcels are used. In the case of the Hortonworks cluster, it assumes that Apache Ambari was used for setup. In the case of the Oracle Big Data Lite VM, it was configured not using parcels.



Important

Attention. Carefully inspect each folder and file location. Depending on the release of CDH, HDP or other Hadoop cluster release, you might have to search for the appropriate files. Also, the correct name of the file is *Renviron.site* with a capital R. Finally, this file needs to exist in each node that has R installed in the cluster.

12 Appendix 1: Examples of *Renviron.site*.

Below you can find example of "Renviron.site" file taken from a "live" setup of ORAAH on different Hadoop distribution. Please use it for reference or as a base template when setting up ORAAH on your cluster.

12.1 Cloudera Distribution of Hadoop 5.8.0.

```
##### HOME DIR's#####
# If you have an Oracle CLient Configured:
# ORACLE_HOME=/usr/lib/oracle/12.1/client64
# ORACLE_HOSTNAME=localhost

CDH_VERSION=5.8.0

# Optional Settings to skip verification of HDFS and Map Reduce functionality.
# Recommended to be set to 0 (default is 1 if not set) after the configuration
# has been successfull to speed up initialization.
ORCH_HDFS_CHECK=0
ORCH_MAPRED_CHECK=0

##### HOME DIR's#####
R_HOME=/usr/lib64/R
JAVA_HOME=/usr/java/default
HADOOP_HOME=/opt/cloudera/parcels/CDH/lib/hadoop
HIVE_HOME=/opt/cloudera/parcels/CDH/lib/hive
HADOOP_MAPRED_HOME=/opt/cloudera/parcels/CDH/lib/hadoop-mapreduce
ORCH_HADOOP_HOME=/opt/cloudera/parcels/CDH
SQOOP_HOME=/opt/cloudera/parcels/CDH/lib/sqoop
PIG_HOME=/opt/cloudera/parcels/CDH/lib/pig
IMPALA_HOME=/opt/cloudera/parcels/CDH/lib/impala
YARN_HOME=/opt/cloudera/parcels/CDH/lib/hadoop-yarn
SPARK_HOME=/opt/cloudera/parcels/CDH/lib/spark

# OPTIONAL: pointers to folders for different Oracle Big Data Appliance or
# Oracle Big Data Connectors and Oracle NoSQL components. Depending on the
# folders the products are installed.
OLH_HOME=/opt/oracle/oraloader-3.4.0-h2
KVHOME=/u01/nosql/kv-ee
OSCH_HOME=/u01/connectors/osch
COPY2BDA_HOME=/u01/orahivedp

##### CONF DIR's#####
HIVE_CONF_DIR=/opt/cloudera/parcels/CDH/lib/hive/conf
SPARK_CONF_DIR=/opt/cloudera/parcels/CDH/lib/spark/conf

# Spark Java options
SPARK_JAVA_OPTS="-Djava.library.path=/usr/lib64/R/lib"
LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib:/usr/lib64/R/lib:/usr/lib64/R/library/ ↵
rJava:/usr/lib64/R/library/RImpala

ORCH_STREAMING_LIB=/opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-streaming.jar
HADOOP_CLASSPATH=$COPY2BDA_HOME/jlib/*:$OLH_HOME/jlib/*:$HIVE_CONF_DIR:$OSCH_HOME/jlib/*: ↵
$ORACLE_HOME/jdbc/lib/*:$KVHOME/lib/kvstore.jar:/usr/lib/hive-hcatalog/share/hcatalog/ ↵
hive-hcatalog-core.jar

PATH=/usr/lib64/qt-3.3/bin:/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/bin:/bin:/usr/ ↵
bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/oracle/bda/bin:/usr/lib/oracle/12.1/client64/ ↵
bin:/usr/java/default/bin:/home/oracle/bin:$ORCH_HADOOP_HOME:$ORCH_STREAMING_LIB: ↵
$SQOOP_HOME:$OLH_HOME:$SPARK_HOME:$SPARK_JAVA_OPTS:$HADOOP_HOME
```

```

CLASSPATH=/opt/cloudera/parcels/CDH/lib/spark/conf:/etc/hadoop/conf:/opt/cloudera/parcels/ ←
CDH/lib/spark/lib/spark-assembly.jar:/opt/cloudera/parcels/CDH/lib/hadoop/client/htrace- ←
core.jar:/opt/cloudera/parcels/CDH/lib/hadoop/client/jackson-annotations.jar:/opt/ ←
cloudera/parcels/CDH/lib/hadoop/client/jackson-core.jar:/opt/cloudera/parcels/CDH/lib/ ←
hadoop/client/jackson-databind.jar:/opt/cloudera/parcels/CDH/lib/hadoop/lib/slf4j- ←
log4j12.jar:/opt/cloudera/parcels/CDH/lib/hadoop/lib/slf4j-api-1.7.5.jar:/opt/cloudera/ ←
parcels/CDH/lib/hadoop/lib/log4j-1.2.17.jar:/opt/cloudera/parcels/CDH/lib/hadoop-hdfs/ ←
hadoop-hdfs-nfs.jar:/opt/cloudera/parcels/CDH/lib/hadoop/lib/commons-cli-1.2.jar:/opt/ ←
cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-client-core.jar:/opt/cloudera ←
/parcels/CDH/lib/hadoop-yarn/hadoop-yarn-common.jar:/opt/cloudera/parcels/CDH/lib/hadoop ←
-yarn/hadoop-yarn-api.jar:/opt/cloudera/parcels/CDH/lib/hadoop-yarn/hadoop-yarn-client. ←
jar:/opt/cloudera/parcels/CDH/lib/hadoop-yarn/hadoop-yarn-server-web-proxy.jar:/opt/ ←
cloudera/parcels/CDH/lib/hadoop/hadoop-common.jar:/opt/cloudera/parcels/CDH/lib/hadoop/ ←
lib/guava-11.0.2.jar:/opt/cloudera/parcels/CDH/lib/hadoop/lib/commons-collections-3.2.2. ←
jar:/opt/cloudera/parcels/CDH/lib/hadoop/lib/commons-configuration-1.6.jar:/opt/cloudera ←
/parcels/CDH/lib/hadoop/lib/commons-lang-2.6.jar:/opt/cloudera/parcels/CDH/lib/hadoop/ ←
lib/snappy-java-1.0.4.1.jar:/opt/cloudera/parcels/CDH/lib/hadoop/lib/protobuf-java ←
-2.5.0.jar:/opt/cloudera/parcels/CDH/lib/hadoop/hadoop-auth.jar:/opt/cloudera/parcels/ ←
CDH/lib/hadoop-hdfs/hadoop-hdfs.jar:/opt/cloudera/parcels/CDH/lib/hadoop/lib/commons- ←
logging-1.1.3.jar:/opt/cloudera/parcels/CDH/lib/hadoop/client/commons-logging.jar:/opt/ ←
cloudera/parcels/CDH/lib/hadoop/lib/jersey-core-1.9.jar:/opt/cloudera/parcels/CDH/lib/ ←
hadoop/lib/jersey-server-1.9.jar:/opt/cloudera/parcels/CDH/lib/hadoop/lib/htrace-core4 ←
-4.0.1-incubating.jar:/opt/cloudera/parcels/CDH/lib/hadoop/lib/avro.jar:/opt/cloudera/ ←
parcels/CDH/lib/hadoop/parquet-hadoop.jar:/opt/cloudera/parcels/CDH/lib/hadoop/parquet- ←
jackson.jar:/opt/cloudera/parcels/CDH/lib/hadoop/parquet-common.jar:/opt/cloudera/ ←
parcels/CDH/lib/hadoop/parquet-encoding.jar:/opt/cloudera/parcels/CDH/lib/hadoop/parquet ←
-format.jar:/opt/cloudera/parcels/CDH/lib/hadoop/parquet-column.jar

# END OF Renviron.site

```

12.2 Hortonworks Distribution of Hadoop 2.4.0.0-169.

Note

Tested with ORAAH 2.6.0 release.

```

# Set the correct HAL version for Hortonworks 2.4
ORCH_HAL_VERSION=4.2

##### HOME DIR's#####
# If using and Oracle Database Client
# ORACLE_HOME=/usr/lib/oracle/12.1/client64
R_HOME=/usr/lib64/R
JAVA_HOME=/usr/lib/jvm/java/bin/java
HADOOP_HOME=/usr/hdp/current/hadoop-client
# Not needed if the previous is set ORCH_HADOOP_HOME=/usr/hdp/2.4.0.0-169/hadoop
HADOOP_MAPRED_HOME=/usr/hdp/2.4.0.0-169/hadoop-mapreduce
SQOOP_HOME=/usr/hdp/2.4.0.0-169/sqoop
HIVE_HOME=/usr/hdp/2.4.0.0-169/hive
PIG_HOME=/usr/hdp/2.4.0.0-169/pig
YARN_HOME=/usr/hdp/2.4.0.0-169/hadoop-yarn
SPARK_HOME=/usr/hdp/2.4.0.0-169/spark

# Optional Settings to skip verification of HDFS and Map Reduce functionality.
# Recommended to be set to 0 (default is 1 if not set) after the configuration
# has been successfull to speed up initialization.
ORCH_HDFS_CHECK=0
ORCH_MAPRED_CHECK=0

```

```

##### CONF DIR's#####
# HADOOP_CONF_DIR=/etc/hadoop/conf
HIVE_CONF_DIR=/usr/hdp/2.4.0.0-169/hive/conf
SPARK_CONF_DIR=/usr/hdp/current/spark-historyserver/conf

# Spark Java options
SPARK_JAVA_OPTS="-Djava.library.path=/usr/lib64/R/lib"

LD_LIBRARY_PATH=/usr/lib64/R/lib:/usr/lib64/R/library/rJava:/usr/lib64/R/library/RImpala

ORCH_STREAMING_LIB=/usr/hdp/2.4.0.0-169/hadoop-mapreduce/hadoop-streaming.jar

HADOOP_CLASSPATH=$HIVE_CONF_DIR:/usr/hdp/2.4.0.0-169/hive-hcatalog/share/hcatalog/hive- ↵
hcatalog-core.jar

PATH=/usr/lib64/qt-3.3/bin:/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/bin:/bin:/usr/ ↵
bin:/usr/local/sbin:/usr/sbin:/sbin:/usr/java/default/bin:$ORCH_HADOOP_HOME: ↵
$ORCH_STREAMING_LIB:$SQOOP_HOME:$SPARK_HOME:$HADOOP_HOME

# One has to be careful with the proper releases here. Even if the cluster has
# newer releases of jackson-annotations for example, those might be used by
# other tools tha are not HDP.

CLASSPATH=/usr/hdp/current/spark-historyserver/conf:/etc/hadoop/conf:/usr/hdp/2.4.0.0-169/ ↵
spark/lib/spark-hdp-assembly.jar:/usr/hdp/2.4.0.0-169/hadoop/client/htrace-core.jar:/usr ↵
/hdp/2.4.0.0-169/hadoop/lib/jackson-annotations-2.2.3.jar:/usr/hdp/2.4.0.0-169/hadoop/ ↵
client/jackson-core.jar:/usr/hdp/2.4.0.0-169/hadoop/lib/jackson-databind-2.2.3.jar:/usr/ ↵
hdp/2.4.0.0-169/hadoop/lib/slf4j-log4j12-1.7.10.jar:/usr/hdp/2.4.0.0-169/hadoop/lib/ ↵
slf4j-api-1.7.10.jar:/usr/hdp/2.4.0.0-169/hadoop/lib/log4j-1.2.17.jar:/usr/hdp ↵
/2.4.0.0-169/hadoop-hdfs/hadoop-hdfs-nfs.jar:/usr/hdp/2.4.0.0-169/hadoop/lib/commons-cli ↵
-1.2.jar:/usr/hdp/2.4.0.0-169/hadoop-mapreduce/hadoop-mapreduce-client-core.jar:/usr/hdp ↵
/2.4.0.0-169/hadoop-yarn/hadoop-yarn-common.jar:/usr/hdp/2.4.0.0-169/hadoop-yarn/hadoop- ↵
yarn-api.jar:/usr/hdp/2.4.0.0-169/hadoop-yarn/hadoop-yarn-client.jar:/usr/hdp ↵
/2.4.0.0-169/hadoop-yarn/hadoop-yarn-server-web-proxy.jar:/usr/hdp/2.4.0.0-169/hadoop/ ↵
hadoop-common.jar:/usr/hdp/2.4.0.0-169/hadoop/lib/guava-11.0.2.jar:/usr/hdp/2.4.0.0-169/ ↵
hadoop/lib/commons-collections-3.2.2.jar:/usr/hdp/2.4.0.0-169/hadoop/lib/commons- ↵
configuration-1.6.jar:/usr/hdp/2.4.0.0-169/hadoop/lib/commons-lang-2.6.jar:/usr/hdp ↵
/2.4.0.0-169/hadoop/lib/snappy-java-1.0.4.1.jar:/usr/hdp/2.4.0.0-169/hadoop/lib/protobuf ↵
-java-2.5.0.jar:/usr/hdp/2.4.0.0-169/hadoop/hadoop-auth.jar:/usr/hdp/2.4.0.0-169/hadoop- ↵
hdfs/hadoop-hdfs.jar:/usr/hdp/2.4.0.0-169/hadoop/lib/commons-logging-1.1.3.jar:/usr/hdp ↵
/2.4.0.0-169/hadoop/client/commons-logging.jar:/usr/hdp/2.4.0.0-169/hadoop/client/jersey ↵
-core.jar:/usr/hdp/2.4.0.0-169/hadoop/lib/jersey-server-1.9.jar

# END OF Renviron.site

```

12.3 Oracle's Big Data Lite Virtual Machine 4.5.0.

```

# Required settings for the Oracle Database and Oracle R Enterprise
TZ=EDT
ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/dbhome_1
ORACLE_SID=cdb
ORACLE_HOSTNAME=localhost

##### HOME DIR's#####
# Configuration for the Oracle Loader for Hadoop
OLH_HOME=/u01/connectors/olh
SQOOP_HOME=/usr/lib/sqoop
JAVA_HOME=/usr/java/latest
HADOOP_CONF_DIR=/etc/hadoop/conf
HIVE_CONF_DIR=/etc/hive/conf

```

```
HIVE_HOME=/usr/lib/hive
R_HOME=/usr/lib64/R
KVHOME=/u01/nosql/kv-ee
OSCH_HOME=/u01/connectors/osch
COPY2BDA_HOME=/u01/orahivedp

# Optional Settings to skip verification of HDFS and Map Reduce functionality.
# Recommended to be set to 0 after the configuration has been successful to
# speed up initialization
ORCH_HDFS_CHECK=0
ORCH_MAPRED_CHECK=0

LD_LIBRARY_PATH=/usr/java/latest/jre/lib/amd64/server:/u01/app/oracle/product/12.1.0.2/ ↵
dbhome_1/lib:/usr/lib64/R/lib:/usr/lib/hadoop/lib/native

PATH=/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/ ↵
usr/lib64/R/bin:/u01/Middleware/jdeveloper/jdev/bin:/usr/java/latest/bin:/u01/app/oracle ↵
/product/12.1.0.2/dbhome_1/bin:/home/oracle/scripts:/opt/bin:/u01/sqlcl/bin:/home/oracle ↵
/bin:/usr/lib64/R/bin:/u01/Middleware/jdeveloper/jdev/bin:/usr/java/latest/bin:/u01/app/ ↵
oracle/product/12.1.0.2/dbhome_1/bin:/home/oracle/scripts:/opt/bin

SPARK_HOME=/usr/lib/spark
HADOOP_HOME=/usr/lib/hadoop
CDH_VERSION=5.7.0

# Spark Java options
SPARK_JAVA_OPTS="-Djava.library.path=/usr/lib64/R/lib"

ORCH_STREAMING_LIB=/usr/lib/hadoop-mapreduce/hadoop-streaming.jar

CLASSPATH=/usr/lib/spark/conf:/etc/hadoop/conf:/usr/lib/spark/lib/spark-assembly.jar:/usr/ ↵
lib/hadoop/client/htrace-core4.jar:/usr/lib/hadoop/client/jackson-annotations.jar:/usr/ ↵
lib/hadoop/client/jackson-core.jar:/usr/lib/hadoop/client/jackson-databind.jar:/usr/lib/ ↵
hadoop/hadoop-common.jar:/usr/lib/hadoop/lib/slf4j-log4j12.jar:/usr/lib/hadoop/lib/slf4j ↵
-api-1.7.5.jar:/usr/lib/hadoop/lib/log4j-1.2.17.jar:/usr/lib/hadoop/lib/guava-11.0.2.jar ↵
:/usr/lib/hadoop/lib/commons-collections-3.2.2.jar:/usr/lib/hadoop/lib/commons- ↵
configuration-1.6.jar:/usr/lib/hadoop/lib/commons-lang-2.6.jar:/usr/lib/hadoop/hadoop- ↵
auth.jar:/usr/lib/hadoop/lib/snappy-java-1.0.4.1.jar:/usr/lib/hadoop/lib/protobuf-java ↵
-2.5.0.jar:/usr/lib/hadoop-hdfs/hadoop-hdfs.jar:/usr/lib/hadoop/lib/commons-cli-1.2.jar ↵
:/usr/lib/hadoop-mapreduce/hadoop-mapreduce-client-core.jar:/usr/lib/hadoop-yarn/hadoop- ↵
yarn-common.jar:/usr/lib/hadoop-yarn/hadoop-yarn-api.jar:/usr/lib/hadoop-yarn/hadoop- ↵
yarn-client.jar:/usr/lib/hadoop-yarn/hadoop-yarn-server-web-proxy.jar:/usr/lib/hadoop/ ↵
client/jersey-core.jar:/usr/lib/hadoop/lib/jersey-server-1.9.jar

HADOOP_CLASSPATH=/u01/orahivedp/jlib/*:/u01/connectors/olh/jlib/*:/etc/hive/conf:/u01/ ↵
connectors/osch/jlib/*:/u01/app/oracle/product/12.1.0.2/dbhome_1/jdbc/lib/*:/u01/nosql/ ↵
kv-ee/lib/kvstore.jar:/usr/lib/hive-hcatalog/share/hcatalog/hive-hcatalog-core.jar

# END OF Renviron.site
```

13 Copyright Notice

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.
0116
