

Cluster Verification Utility (CVU)

Frequently Asked Questions

This document addresses some of the frequently asked questions and common concerns regarding the *Cluster Verification Framework* and the *cluvfy* tool.

Update Schedule

[What will be the update schedule for CVU on OTN?](#)

General concept

[What is CVU? What are its objectives and features?](#)

[What is a stage?](#)

[What is a component?](#)

[What is nodelist?](#)

[What is a configuration file?](#)

[Do I have to be root to use CVU?](#)

[What about discovery? Does CVU discover installed components?](#)

[What about locale? Does CVU support other languages?](#)

[How do I report a bug?](#)

[What version of Oracle Clusterware\(CRS\) or RAC is supported by CVU?](#)

Installation

[What are the requirements for CVU?](#)

[How do I install CVU?](#)

[What is 'cvuqdisk' rpm? Why should I install this rpm?](#)

[How do I install 'cvuqdisk' package?](#)

Usage

[How do I know about cluvfy commands? The usage text of cluvfy does not show individual commands.](#)

[What are the default values for the command line arguments?](#)

[Do I have to type the nodelist every time for the CVU commands? Is there any shortcut?](#)

[How do I get detailed output of a check?](#)

[How do I check network or node connectivity related issues?](#)

[How do I check whether OCFS is properly configured?](#)

[How do I check the CRS stack and other sub-components of it?](#)

[How do I check user accounts and administrative permissions related issues?](#)

[How do I check if SSH is configured properly on my cluster?](#)

CVU FAQ

[How do I check minimum system requirements on the nodes?](#)

[Can I check if the storage is shared among the nodes?](#)

[Is there a way to compare nodes?](#)

[Why the peer comparison with *-refnode* says "matched" when the group or user does not exist?](#)

[Is there a way to verify that the CRS is working properly before proceeding with RAC install?](#)

[At what point cluvfy is usable? Can I use cluvfy before installing CRS?](#)

[How do I turn on tracing?](#)

[Where can I find the CVU trace files?](#)

[Why cluvfy reports "unknown" on a particular node?](#)

[Why does CVU complain "ERROR: Could not find a suitable set of interfaces for VIPs"?](#)

Limitations

[What are the known issues with this release?](#)

Platform Specific

The following sections cover the frequently asked questions for platforms where CVU is available for download.

LINUX

[What Linux distributions are supported?](#)

[How do I make CVU work on SuSE ES 9?](#)

WINDOWS

[What Windows versions are supported?](#)

SOLARIS

[What Solaris versions are supported?](#)

HP-UX

[What HP-UX versions are supported?](#)

AIX

[What AIX versions are supported?](#)

CVU FAQ

What will be the update schedule for CVU on OTN?

CVU may be updated and uploaded on OTN every two months with fixes for known issues or important bugs. Please check README associated with platforms on OTN for updates on CVU.

[\[go to the top \]](#)

What is CVU? What are its objectives and features?

CVU is a utility that is distributed with Oracle Clusterware. It was developed to assist in the installation and configuration of Oracle Clusterware as well as RAC. Cluvfy will verify all the important components that need to be verified at different stages in configuring a RAC environment. The wide domain of deployment of CVU ranges from initial hardware setup through fully operational cluster for RAC deployment and covers all the intermediate stages of installation and configuration of various components. The command line tool is *cluvfy*. *Cluvfy* is a non-intrusive utility and will not adversely affect the system or operational stack.

[\[go to the top \]](#)

What is a stage?

CVU supports the notion of Stage verification. It identifies all the important stages in RAC deployment and provides each stage with its own entry and exit criteria. The entry criteria for a stage define a specific set of verification tasks to be performed before initiating that stage. This pre-check saves the user from entering into a stage unless its pre-requisite conditions are met. The exit criteria for a stage define another specific set of verification tasks to be performed after completion of the stage. The post-check ensures that the activities for that stage have been completed successfully. It identifies any stage specific problem before it propagates to subsequent stages; thus making it difficult to find its root cause. An example of a stage is "*pre-check of database installation*", which checks whether the system meets the criteria for RAC install.

[\[go to the top \]](#)

What is a component?

CVU supports the notion of Component verification. The verifications in this category are not associated with any specific stage. The user can verify the correctness of a specific cluster component. A component can range from a basic one, like free disk space to a complex one like Oracle Clusterware (CRS) Stack. The integrity check for the Oracle Clusterware stack will transparently span over verification of multiple sub-components associated with Oracle Clusterware stack. Bundling of several relevant tasks as a component is of great use to the user for verifying a specific cluster component.

[\[go to the top \]](#)

What is nodelist?

CVU FAQ

A nodelist is a comma-separated list of hostnames without domain. Cluvfy will run the requested verification on all nodes in the nodelist provided. Cluvfy will ignore any domain while processing the nodelist. If duplicate entities after removing the domain exist, cluvfy will eliminate the duplicate names while processing. Wherever supported, you can use '-n all' to check on all the cluster nodes. Check "[Do I have to type the nodelist every time for the CVU commands? Is there any shortcut?](#)" for more information on nodelist and shortcuts.

[\[go to the top \]](#)

What is a configuration file?

CVU supports a configuration file called *cvu_config* under CV_HOME/cv/admin folder. This file supports property-value style preferences in a persistent way. This might vary depending upon the platform. Here is a brief description of some of those properties:

[\[go to the top \]](#)

CV_NODE_ALL:

This property stores a comma separated list of nodes to be used for all the nodes in the cluster. This value will be used for "-n all" argument on the command line. For detail refer to "[Do I have to type the nodelist every time for the CVU commands? Is there any shortcut?](#)".

CV_RAW_CHECK_ENABLED:

If this property is set to TRUE, then CVU will perform scsi disk discovery and sharedness checks. For Linux platforms, CVU requires the cvuqdisk rpm installed on all nodes if this property is set. For detail refer to "[What is 'cvuqdisk' rpm? Why should I install this rpm?](#)".

CV_ASSUME_DISTID:

This property is used in cases where CVU cannot detect or support a particular platform or a distribution. It is not recommended to change this property as this might render CVU non-functional.

CV_XCHK_FOR_SSH_ENABLED:

If this property is set to TRUE, CVU will also check whether X-Windows is configured with SSH for user equivalence. For detail, refer to "[How do I check if SSH is configured properly on my cluster?](#)".

ORACLE_SRVM_REMOTESHELL:

This property stores alternative remote shell command location.

ORACLE_SRVM_REMOTECOPY:

This property stores alternative remote copy command location.

[\[go to the top \]](#)

Do I have to be root to use CVU?

CVU FAQ

No. CVU is intended for database and system administrators. CVU assumes the current user as *oracle* user.

[\[go to the top \]](#)

What about discovery? Does CVU discover installed components?

At present, CVU's discovery is limited to the following components. CVU discovers available network interfaces if you do not specify any interface in its command line. For storage related verification, CVU discovers all the supported storage types if you do not specify a particular storage. CVU discovers CRS HOME if one is available. CVU also discovers the statically configured nodelist for the cluster if an Oracle supported vendor clusterware or Oracle Clusterware is available.

[\[go to the top \]](#)

What about locale? Does CVU support other languages?

Yes. CVU complied with Oracle's NLS guidelines and supports locale.

[\[go to the top \]](#)

How do I report a bug?

Please refer to the "[What are the known issues with this release?](#)" section of this document and the README file before filing a bug. If the issue is not covered in those documents, open a TAR through Oracle Support.

[\[go to the top \]](#)

What version of Oracle Clusterware(CRS) or RAC is supported by CVU?

The current CVU release supports only RAC 10g and Oracle Clusterware(CRS). In other words, "the current version" of CVU can check 10g Release 1 as well as 10g Release 2 releases of Oracle Clusterware or RAC. However, it cannot check or verify pre-10g (Oracle 9i) products.

[\[go to the top \]](#)

What are the requirements for CVU?

CVU requires:

1. _ An area with at least 30MB of free space for containing software bits on the invocation node.
2. _ A Java 1.4.1(or higher) location on the invocation node.
3. _ A work directory with at least 25MB on **all** the nodes. CVU will attempt to copy the necessary bits as required to this location. Make sure, the location exists on all nodes and it has write permission for CVU user. This directory is set through the CV_DESTLOC environment variable. If this variable is not set, CVU will use the common temporary location such as "/tmp" for Linux and "C:\Temp" for Windows as the work dir.

CVU FAQ

4. **An optional package** 'cvuqdisk' is required on all the nodes for Linux distributions. This assists CVU in finding scsi disks and helps CVU to perform storage checks on disks. Please refer to [What is 'cvuqdisk' rpm?](#) for detail. Note that, this package should be installed only on Red Hat Linux 3(or higher) or SuSE 9 distribution or on other Linux flavors of comparable versions.

[\[go to the top \]](#)

How do I install CVU?

Here is how one can install CVU from a zip file.

1. **create a cvhome**(say */home/mycvhome*) directory. It should have at least 30M of free disk space.
2. **cd** /home/mycvhome
3. **copy** the zip file to /home/mycvhome
4. **unzip** the file:
> unzip <zip-file>
5. **set** these environmental variables:
CV_HOME: This should point to the cvhome.
> setenv CV_HOME /home/mycvhome

CV_JDKHOME: This should point to a valid jdk1.4 home
> setenv CV_JDKHOME /usr/local/packages/jdk14

CV_DESTLOC (optional): This should point to a writable area on **all** nodes. The tool will attempt to copy the necessary bits as required to this location. Make sure the location exists on all nodes and it has write permission for CVU user. It is strongly recommended that you should set this variable. If this variable has not been set, CVU will use "/tmp" as the default.

> setenv CV_DESTLOC /tmp/cvu_temp

To verify, run /home/mycvhome/bin/cluvfy. This should show the usage.

For Linux platforms (except Red Hat 2.1), an optional rpm package 'cvuqdisk' is required on all the nodes. Please refer to [How do I install 'cvuqdisk' package?](#) for detail.

[\[go to the top \]](#)

What is 'cvuqdisk' rpm? Why should I install this rpm?

CVU requires root privilege to gather information about the scsi disks during discovery. A small binary uses the setuid mechanism to query disk information as root. Note that this process is purely a read-only process with no adverse impact on the system. To make this secured, this binary is packaged in the cvuqdisk rpm and need root privilege to install on a machine.

When this package is installed on all the nodes, CVU performs discovery and shared storage accessibility checks for scsi disks. Otherwise, it complains about the missing

CVU FAQ

package 'cvuqdisk'. You can disable the scsi device check feature by setting the CV_RAW_CHECK_ENABLED to FALSE in \$CV_HOME/cv/admin/cvu_config file. CVU will not complain about the missing rpm if this variable is set to false.

Discovery of scsi disks for RedHat 2.1AS is not supported. Hence this package should not be installed on RedHat 2.1 AS.

[\[go to the top \]](#)

How do I install 'cvuqdisk' package?

Here are the steps to install cvuqdisk package.

1. _ Become root user
2. _ Copy the rpm (cvuqdisk-1.0.1-1.rpm or the latest version) to a local directory. You can find the rpm in Oracle's OTN site.
3. _ Set the environment variable to a group, who should own this binary. Typically it is the "dba" group.
export CVUQDISK_GRP=dba
4. _ Erase any existing package
rpm -e cvuqdisk
5. _ Install the rpm
rpm -iv cvuqdisk-1.0.1-1.rpm
6. _ Verify the package
rpm -qa | grep cvuqdisk

[\[go to the top \]](#)

How do I know about *cluvfy* commands? The usage text of *cluvfy* does not show individual commands.

Cluvfy has context sensitive help built into it. Cluvfy shows the most appropriate usage text based on the cluvfy command line arguments.

If you type 'cluvfy' on the command prompt, cluvfy displays the high-level generic usage text, which talks about valid stage and component syntax.

If you type '*cluvfy comp -list*', cluvfy will show valid components with brief description on each of them. If you type '*cluvfy comp -help*', cluvfy will show detail syntax for each of the valid components. Similarly, '*cluvfy stage -list*' and '*cluvfy stage -help*' will list valid stages and their syntax respectively.

If you type an invalid command, cluvfy will show the appropriate usage for that particular command. For example, if you type '*cluvfy stage -pre dbinst*', cluvfy will show the syntax for pre-check of dbinst stage.

[\[go to the top \]](#)

What are the default values for the command line arguments?

Here are the default values and behavior for different stage and component commands:

CVU FAQ

For component *nodecon*:

If no *-i* arguments is provided, then clufy runs in the discovery mode.

For component *nodereach*:

If no *-srcnode* is provided, then the local (node of invocation) will be used as the source node.

For components *ssa*:

If no *-n* argument is provided, then the local node will be used.

If no *-s* argument is provided, then clufy runs in the storage discovery mode.

For components *clu*:

If no *-n* argument is provided, then all the nodes in the cluster will be used for verification.

For components *cfs, ocr, crs, space, clumgr, nodeapp*:

If no *-n* argument is provided, then the local node will be used.

For components *sys*:

If no *-n* argument is provided, then the local node will be used.

If no *-r* argument is provided, then 10gR2 will be used.

If no *-osdba* argument is provided, then 'dba' will be used.

If no *-orainv* argument is provided, then 'oinstall' will be used.

For components *admprv*:

If no *-n* argument is provided, then the local node will be used.

If no *-osdba* argument is provided, then 'dba' will be used.

If no *-orainv* argument is provided, then 'oinstall' will be used.

For component *peer*:

If no *-r* argument is provided, then 10gR2 will be used.

If no *-osdba* argument is provided, then 'dba' will be used.

If no *-orainv* argument is provided, then 'oinstall' will be used.

For stage *-post hwos*:

If no *-s* argument is provided, then clufy runs in the discovery mode.

For stage *-pre crsinst*:

If no *-r* argument is provided, then 10gR2 will be used.

If no *-c* argument is provided, then clufy will skip OCR related checks.

If no *-q* argument is provided, then clufy will skip voting disk related checks.

If no *-osdba* argument is provided, then 'dba' will be used.

If no *-orainv* argument is provided, then 'oinstall' will be used.

For stage *-pre dbinst*:

If no *-r* argument is provided, then 10gR2 will be used.

If no *-osdba* argument is provided, then 'dba' will be used.

[\[go to the top \]](#)

Do I have to type the nodelist every time for the CVU commands? Is there any shortcut?

You do not have to type the nodelist every time for the CVU commands. Typing the nodelist for a large cluster is painful and error prone. Here are few shortcuts.

CVU FAQ

To provide **all** the nodes of the cluster, type '*-n all*'. Cluvfy will attempt to get the nodelist in the following order:

1. If a vendor clusterware is available, it will pick all the configured nodes from the vendor clusterware using *lsnodes* utility.
2. If CRS is installed, it will pick all the configured nodes from Oracle clusterware using *olsnodes* utility.
3. It will look for the CV_NODE_ALL property in the *cvu_config* file under `$CV_HOME/cv/admin`.
4. If none of the above, it will look for the CV_NODE_ALL environmental variable.
5. Otherwise, it will complain.

To provide a partial list (some of the nodes of the cluster) of nodes, you can set an environmental variable and use it in the CVU command. For example:

```
export MYNODES=node1,node3,node5
cluvfy comp nodecon -n $MYNODES
```

[\[go to the top \]](#)

How do I get detail output of a check?

Cluvfy supports a verbose mode. By default, cluvfy reports in non-verbose mode and just reports the summary of a test. To get detailed output of a check, use the flag '*-verbose*' in the command line. This will produce detail output of individual checks and where applicable will show per-node result in a tabular fashion.

[\[go to the top \]](#)

How do I check network or node connectivity related issues?

Use component verifications commands like 'nodereach' or 'nodecon' for this purpose. For detailed syntax of these commands, type `comp -help` command on the command prompt.

If the 'comp nodecon' command is invoked without *-i*, cluvfy will attempt to discover all the available interfaces and the corresponding IP address & subnet. Then cluvfy will try to verify the node connectivity per subnet. You can run this command in verbose mode to find out the mappings between the interfaces, IP addresses and subnets. Cluvfy will suggest interfaces for VIP and private interconnect if suitable interfaces are available.

You can check the connectivity among the nodes through specific interfaces by specifying the interface name(s) through *-i* argument.

[\[go to the top \]](#)

Can I check if the storage is shared among the nodes?

Yes, you can use 'cluvfy comp ssa' command to check the sharedness of the storage. Please refer to the [known issues](#) section for the type of storage supported by cluvfy.

[\[go to the top \]](#)

CVU FAQ

How do I check whether OCFS is properly configured?

You can use the component command 'cluvfy comp cfs' to check this. Provide the OCFS file system you want to check through the -f argument. Note that, the sharedness check for the file system is supported for OCFS version 1.0.14 or higher. At present, cluvfy can not handle OCFS2 discovery on RHEL4 and SUSE9 platforms.

[\[go to the top \]](#)

How do I check the CRS stack and other sub-components of it?

Cluvfy provides commands to check a particular sub-component of the CRS stack as well as the whole CRS stack. You can use the 'comp ocr' command to check the integrity of OCR. Similarly, you can use 'comp crs' and 'comp clumgr' commands to check integrity of crs and clustermanager sub-components.

To check the entire CRS stack, run the stage command 'cluvfy stage -post crsinst'.

[\[go to the top \]](#)

How do I check user accounts and administrative permissions related issues?

Use *admprv* component verification command. Refer to the usage text for detail instruction and type of supported operations. To check whether the privilege is sufficient for user equivalence, use '-o user_equiv' argument. You can force CVU to check user equivalence using SSH only by the '-sshonly' flag. Similarly, the '-o crs_inst' will verify whether the user has the correct permissions for installing CRS. The '-o db_inst' will check for permissions required for installing RAC and '-o db_config' will check for permissions required for creating a RAC database or modifying a RAC database configuration.

[\[go to the top \]](#)

How do I check if SSH is configured properly on my cluster?

You can use CVU's *admprv* component verification command 'comp admprv -n <nodelist> -o user_equiv -sshonly -verbose' to verify this. To check whether X-Windows is configured to work with SSH for user equivalence as per Oracle's requirement, set the following property "CV_XCHK_FOR_SSH_ENABLED=TRUE" in the \$CV_HOME/cv/admin/cvu_config file.

[\[go to the top \]](#)

How do I check minimum system requirements on the nodes?

The component verification command *sys* is meant for that. Note that, CVU can check the minimum system requirements for Oracle Clusterware versions 10gR1 as well as 10gR2. Use the '-p crs' argument to check requirements for Oracle Clusterware and -r argument for the desired version. Similarly, CVU can check the minimum system requirements for RAC versions 10gR1 and 10gR2. For RAC, you have to use the '-p

CVU FAQ

database' argument.

[\[go to the top \]](#)

Is there a way to compare nodes?

You can use the peer comparison feature of cluvfy for this purpose. The command 'comp peer' will list the values of different nodes for several pre-selected properties. You can use the *peer* command with *-refnode* argument to compare those properties of other nodes against the reference node.

[\[go to the top \]](#)

Why the peer comparison with *-refnode* says *matched* when the group or user does not exist?

Peer comparison with the *-refnode* feature acts like a baseline feature. It compares the system properties of other nodes against the reference node. If the value does not match(not equal to reference node value), then it flags that as a deviation from the reference node. If a group or user does not exist on reference node as well as on the other node, it will report this as 'matched' since there is no deviation from the reference node. Similarly, it will report as 'mismatched' for a node with higher total memory than the reference node for the above reason.

[\[go to the top \]](#)

Is there a way to verify that the CRS is working properly before proceeding with RAC install?

Yes. You can use the post-check command for cluster services setup(*-post crsinst*) to verify CRS status. A more appropriate test would be to use the pre-check command for database installation(*-pre dbinst*). This will check whether the current state of the system is suitable for RAC install.

[\[go to the top \]](#)

At what point cluvfy is usable? Can I use cluvfy before installing CRS?

You can run cluvfy at any time, even before CRS installation. In fact, cluvfy is designed to assist the user as soon as the hardware and OS is up. If you invoke a command which requires CRS or RAC on local node, cluvfy will report an error if those required products are not yet installed.

[\[go to the top \]](#)

How do I turn on tracing?

Set the environmental variable SRVM_TRACE to true. For example, in bash "export SRVM_TRACE=true" will turn on tracing.

[\[go to the top \]](#)

CVU FAQ

Where can I find the CVU trace files?

CVU log files can be found under `$CV_HOME/cv/log` directory. The log files are automatically rotated and the latest log file has the name `cvutrace.log.0`. It is a good idea to clean up unwanted log files or archive them to reclaim disk place.

Note that, no trace files will be generated if tracing has not been turned on.

[\[go to the top \]](#)

Why clufvy reports "unknown" on a particular node?

Clufvy reports unknown when it can not conclude for sure if the check passed or failed.

[\[go to the top \]](#)

Why does CVU complain "ERROR: Could not find a suitable set of interfaces for VIPs"?

CVU checks for the following criteria before considering a set of interfaces for VIP:

- the interfaces should have the same name across nodes
- they should belong to the same subnet
- they should have the same netmask
- they should be on public(and routable) network.

Oftentimes, the interfaces planned for the VIPs are configured on `10.*`, `172.16.*` - `172.31.*` or `192.168.*` networks, which are not routable. Hence CVU does not consider them as suitable for VIPs. If none of the available interfaces satisfy this criteria, CVU complains "ERROR: Could not find a suitable set of interfaces for VIPs.". It is worth noting that, such addresses will actually work if they're public, but CVU just thinks they're private and reports accordingly.

[\[go to the top \]](#)

What are the known issues with this release?

1. Shared storage accessibility(ssa) check reports

Current release of clufvy has the following limitations on Linux regarding shared storage accessibility check.

- a. Currently NAS storage (r/w, no attribute caching), OCFS(version 1.0.14 or higher) and scsi disks(if `cvuqdisk` package is installed) are supported. Note that, discovery of scsi disks for RedHat Linux 2.1 is not supported.
- b. For sharedness check on NAS, clufvy requires the user to have write permission on the specified path. If the clufvy user does not have write permission, clufvy reports the path as not-shared.

2. OCFS2 discovery issue with RedHat 4 and SuSE 9.

At present, CVU can not perform discovery of OCFS2 on RedHat 4 and SuSE9.

CVU FAQ

3. CVU does not recognize the disk bindings (e.g. /dev/raw/raw1) as valid storage paths or identifiers. Please use the underlying disk(e.g. /dev/sdm etc) for the storage path or storage identifier. On Windows, use "\\Device\Harddisk<n>" notation such as "\\Device\Harddisk1" for the storage path or identifier.

[\[go to the top \]](#)

LINUX

What Linux distributions are supported?

RedHat:

RedHat 2.1AS: Not available

RedHat EL-3(update 2 or higher): Supported

RedHat EL-4 : Supported

Note that, the CVU binary for RedHat 2.1AS is different from CVU binaries for other RedHat distributions; they are not binary compatible. In other words, CVU bits for RedHat 3 and RedHat 2.1 are not the same.

SuSE:

SuSE ES 9: Supported

[\[go to the top \]](#)

How do I make CVU work on SuSE ES 9?

For this, you have to edit the configuration file called *cvu_config* under CV_HOME/cv/admin folder. Modify the property CV_ASSUME_DISTID=Taroon to CV_ASSUME_DISTID=Pensacola.

[\[go to the top \]](#)

WINDOWS

What Windows versions are supported?

CVU supports both Windows 2000 and Windows2003 servers.

[\[go to the top \]](#)

SOLARIS

What Solaris versions are supported?

CVU supports the following Solaris releases:

Solaris 8: Supported

Solaris 9: Supported

Solaris 10: Supported

[\[go to the top \]](#)

HP-UX

What HP-UX versions are supported?

CVU supports the following HP-UX releases:

HP UX 11.11: Supported

CVU FAQ

HP UX 11.23: Supported

[\[go to the top \]](#)

AIX

What AIX versions are supported?

CVU supports the following AIX release:

AIX 5L: Supported

[\[go to the top \]](#)