oswbb

OS Watcher User's Guide

Carl Davis

Center of Expertise

August 29, 2017

Contents

[Introduction](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Introduction)

[Overview](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Overview)

[License](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#License)

[Best Practices](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Best Practices)

[Supported Platforms](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Supported Platform)

[Installation](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Installation)

[Installing oswbb](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Installing OSW)

[Uninstalling oswbb](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Uninstalling OSW)

[Configuration](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Configuration)

[RAC Considerations](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#RAC Considerations)

[Adding Custom Data Collections](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Adding Custom)

[Optional Unix Environment Variables](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Optional)

[Monitoring Private Networks](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Monitoring)

[Setting Non ENGLISH OS System Dates](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Dates)

[Starting / Stopping oswbb](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Starting)

[Starting oswbb](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Starting OSW)

[Stopping oswbb](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Stopping OSW)

[Diagnostic Data Output](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Diagnostic Data Output)

[oswiostat](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswiostat)

[oswmpstat](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswmpstat)

[oswnetstat](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswnetstat)

[oswprvtnet](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswprvtnet)

[oswifconfig](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswifconfig)

[oswps](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswps)

[oswtop](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswtop)

[oswvmstat](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswvmstat)

[Analyzing the Output](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Analyzing the Output)

[Known Issues](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Known Issues)

[Download](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Download)

[Reporting Feedback](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Reporting Feedback)

[Sending Files To Support](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Sending Files To Support)

Introduction

OSWatcher (oswbb) is a collection of UNIX shell scripts intended to collect and archive operating system and network metrics to aid support in diagnosing performance issues. As a best practice, all customers should install and run OSWatcher on every node that has a running Oracle instance. In the case of a performance issue, Oracle support can use this data to help diagnose performance problems which may outside the database. OSWatcher can be downloaded from MOS Note 301137.1 . Installation instructions for OSWatcher are provided in this user guide.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Overview

OSWatcher consists of a series of shell scripts. OSWatcher.sh is the main controlling executive, which spawns individual shell processes to collect specific kinds of data, using Unix operating system diagnostic utilities. Control is passed to individually spawned operating system data collector processes, which in turn collect specific data, timestamp the data output, and append the data to pre-generated and named files. Each data collector will have its own file, created and named by the File Manager process.

Data collection intervals are configurable by the user, but will be uniform for all data collector processes for a single instance of the OSWatcher tool. For example, if OSWatcher is configured to collect data once per minute, each spawned data collector process will generate output for its respective metric, write data to its corresponding data file, then sleep for one minute (or other configured interval) and repeat. Because we are collecting data every minute, the files generated by each spawned processes will contain 60 entries, one for each minute during the previous hour. Each file will contain, at most, one hour of data. At the end of each hour, File Manager will wake up and copy the existing current hour file to an archive location, then create a new current hour file.

The File Manager ensures only the last N hours of information are retained, where N is a configurable integer defaulting to 48. File Manager will wake up once per hour to delete files older than N hours. At any time, the entire output file set will consist of one current hour file, plus N archive files for each data collector process.

stopOSWbb.sh will terminate all processes associated with OSWatcher, and is the normal, graceful mechanism for stopping the tool's operation.

OSWatcher invokes these distinct operating system utilities, each as a distinct background process, as data collectors. These utilities will be supported, or their equivalents, as available for each supported target platform.

ps

top

ifconfig

mpstat

iostat

netstat

traceroute

vmstat

sar (HP-UX Only)

cpuinfo (Linux Only)

meminfo (Linux Only)

slabinfo (Linux Only)

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

License

OSWatcher use is under Oracle's standard licensing terms and does not require additional licenses for use.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Best Practices

As a best practice support recommends that all Oracle users deploy OSWatcher on their servers which are running Oracle. OSWatcher should be considered as a supplemental or complementary data collection to any other data collections that may be in place. The primary reason for this is if support has to file a bug with development, development will most likely insist that OSWatcher data be provided. If not, the bug may not be able to proceed until OSWatcher is installed and the problem happens again. Additionally, support analysts are familiar and trained on understanding the output of basic OS diagnostic utilities like vmstat, iostat, top, etc. Support analysts may not be familiar with other kinds of custom or OS specific data collections you have in place. And finally, support has the ability to analyze the OSWatcher data with tools in house avoiding the time consuming task of having to manually inspect dozens of files. This will greatly reduce your resolution time.

Support recommends that you run OSWatcher with the default snapshot interval of 30 seconds and the default retention period of 48 hours. Taking less frequent snapshots or sampling at a rate greater than 60 seconds is not useful for diagnosing performance issues.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Supported Platforms

OSWatcher is certified to run on the following platforms:

AIX

Solaris

HP-UX

Linux

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Installation

The following sections describe how to install and remove OSWatcher on your system..

Installing oswbb

OSWatcher can be installed as any user as long as that user has permission to execute the underlying Unix utilities such as vmstat, top, etc. In most cases you can install as the Oracle user. OSWatcher needs to be installed on each node if you are running in a RAC environment, one installation per node. Install by using the following procedure:

Downloaded the oswbb.tar file from MOS. Place the tar file is the desired location and untar the file. Next, make sure to change the file permissions on these files to execute by using chmod.

|  |
| --- |
| tar xvf oswbb.tar  chmod 744 \* |

A directory named oswbb will be created which contains the full installation of OSWatcher including the OSWatcher analyzer and all supporting files. OSWatcher is now installed.

Uninstalling oswbb

To uninstall OSWatcher issue the following command on the oswbb directory.

|  |
| --- |
| rm -rf oswbb |

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Configuration

OSWatcher collects and stores data to files in an archive directory. By default, this directory is created under the oswbb directory where oswbb is installed. There are 2 options if you want to change this location to point to any other directory or device.

1. set the UNIX environment variable OSWBB\_ARCHIVE\_DEST to the location desired before starting the tool. In this example the archive directory will be created in this location (/usr/app/archive) and not created under the home oswbb directory

|  |
| --- |
| export OSWBB\_ARCHIVE\_DEST=/usr/app/archive |

2. start oswbb by running the startOSWbb.sh script located in the directory where oswbb is installed and specify the 4th parameter on the command line.

|  |
| --- |
| ./startOSWbb 30 48 None /usr/app/archive |

This script accepts an optional 4th parameter which is the location where you want oswbb to write the the data it collects. If you use the optional 4th parameter you must also set the optional 3rd parameter which specifies the name of a compress or zip(gzip,compress, etc) utility. If you do not want to compress the files you can specify NONE as the 3rd parameter. See the startOSWbb.sh for more details.

OSWbb writes the archive location to a heartbeat file named osw.hb in the /tmp directory. This is done so other oracle utilities like RAC-DDT and RDA can find OSWbb data when these utilities are run. This file gets removed when OSWatcher is stopped.

Once oswbb is installed, scripts have been provided to start and stop the oswbb utility. When oswbb is started for the first time it creates the archive subdirectory, either in the default location under the oswbb directory or in an alternate location as specified above. The archive directory contains a minimum of 7 subdirectories, one for each data collector. Private networks can be monitored by using the traceroute command. This is done automatically in release 8 of OSWatcher. This can also be done manually by the user by creating an executable file in the oswbb directory named [private.net](http://private.net/). An example of what this file should look like is named [Exampleprivate.net](http://exampleprivate.net/) with samples for each operating system: Solaris, LINUX, AIX, and HP-UX in the oswbb directory. This file can be edited and renamed [private.net](http://private.net/) or a new file named [private.net](http://private.net/) can be created. This file contains entries for running the traceroute command to verify RAC private networks.

[Exampleprivate.net](http://exampleprivate.net/) entry on Solaris:

|  |
| --- |
| traceroute -r -F private\_nodename |

Where node1 and node2 are 2 nodes in addition to the hostnode of a 3 node RAC cluster. If the file [private.net](http://private.net/) does not exist or is not executable then no data will be collected and stored under the oswprvtnet directory.

oswbb will need access to the OS utilities: top, vmstat, iostat, mpstat, netstat, and traceroute. These OS utilities need to be installed on the system prior to running oswbb.  Execute permission on these utilities need to be granted to the user of oswbb.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

RAC Considerations

OSWbb needs to be installed on each node in the cluster. If installing on a shared file system then install each node's OSWbb into a unique directory.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Adding Custom Data Collections

You can have OSWbb run your own shell scripts and automatically store and manage the data in the same way OSWbb collects and manages the data it collects like vmstat, iostat, etc. This callable interface is provided "as is" and is not supported. You must write and test your own scripts and then link them to oswbb with this interface. The example provided is a very simple example of calling a standard UNIX utility.

Step 1: Create an executable shell script and place it in the oswbb directory. In that file put the following header lines at the top of the file:

|  |
| --- |
| #!/bin/sh  echo "zzz \*\*\*"`date '+%a %b %e %T %Z %Y'` >> $1 |

Step 2: Redirect the output of your script or UNIX command to $1. $1 is the OSWbb archive directory where OSWbb writes all the files it collects. In the following example the output of running the du command will be redirected to $1 (the oswbb archive directory).

|  |
| --- |
| du >> $1 |

See the example du.sh which is provided in the oswbb directory.

Step 3: Add a new entry in the file extras.txt for that file. See extras.txt for format of the entry.

In the above example, OSWbb will run the du command in this script at the same interval all other commands are run. The output of the script will be in the archive directory. The sample script du.sh is provided in the oswbb directory. You can review it along with the examples in extras.sh to see how to call your scripts.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Optional Unix Environment Variables

There are 2 optional environment variables to control the configuration of OSWatcher. The location of the archive directory can be controlled by specifying OSWBB\_ARCHIVE\_DEST as documented in the above Configuration section above.

A second optional environment variable to control the amount of samples the ps command collects is available. This can be done by specifying export OSW\_PS\_SAMPLE\_MULTIPLIER=n where n = number of samples to skip. Example:

|  |
| --- |
| export OSW\_PS\_SAMPLE\_MULTIPLIER=3 |

OSWatcher is started with a default value of 20 seconds. This would cause ps data to be collected 1 time every 60 seconds (20 \* 3) instead of every 20 seconds.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Monitoring Private Networks

OSWatcher will automatically generate traceroute information and create a file named [private.net](http://private.net/). It is recommended that OSWatcher configure traceroute information automatically which will happen when you start oswbb. Alternatively, you use configure the traceroute command manually by creating the file [private.net](http://private.net/). Create a file named [private.net](http://private.net/) in the oswbb directory. As an example, look at the [Exampleprivate.net](http://exampleprivate.net/) file and manually enter in the hostname or ipaddress you wish to monitor. Each UNIX os uses slightly different arguments to the traceroute command. Refer to [Exampleprivate.net](http://exampleprivate.net/) for examples for each UNIX os.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Setting Non ENGLISH OS System Dates

The OSWatcher analyzer expects the OS system date to be in standard ENGLISH format. To force the UNIX date mask to comply with the analyzer formatting, the parameter oswgCompliance by default is set to 1 in the OSWatcher.sh file.

|  |
| --- |
| oswgCompliance=1 |

Setting this parameter will force a date mask that is readable by the analyzer. Set this parameter to 0 if you do not want the date changed. The analyzer will not be able to analyze files unless the date is ENGLISH.

An additional workaround is suggested in the Known Issues section of this document for this issue.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Starting / Stopping oswbb

Starting oswbb

To start the oswbb utility execute the startOSWbb.sh shell script from the directory where oswbb was installed. This script has 2 arguments which control the frequency that data is collected and the number of hour's worth of data to archive.

ARG1 = snapshot interval in seconds.

ARG2 = the number of hours of archive data to store.

ARG3 = (optional) the name of a compress utility to compress each file automatically after it is created.

ARG4 = (optional) an alternate (non default) location to store the archive directory.

If you do not enter any arguments the script runs with default values of 30 and 48 meaning collect data every 30 seconds and store the last 48 hours of data in archive files.

Example 1: This would start the tool and collect data at default 30 second intervals and log the last 48 hours of data to archive files.

|  |
| --- |
| ./startOSWbb.sh |

Example 2: This would start the tool and collect data at 60 second intervals and log the last 10 hours of data to archive files and automatically compress the files.

|  |
| --- |
| ./startOSWbb.sh 60 10 gzip |

Example 3: This would start the tool and collect data at 60 second intervals and log the last 10 hours of data to archive files, compress the files and set the archive directory to a non-default location.

|  |
| --- |
| ./startOSWbb.sh 60 10 gzip /u02/tools/oswbb/archive |

Example 4: This would start the tool and collect data at 60 second intervals and log the last 48 hours of data to archive files, NOT compress the files and set the archive directory to a non-default location.

|  |
| --- |
| ./startOSWbb.sh 60 48 NONE /u02/tools/oswbb/archive |

Example 5: This would start the tool, put the process in the background, enable to the tool to continue running after the session has been terminated, collect data at 60 second intervals, and log the last 10 hours of data to archive files.

|  |
| --- |
| nohup ./startOSWbb.sh 60 10 & |

Stopping oswbb

To stop the oswbb utility execute the stopOSWbb.sh command from the directory where oswbb was installed. This terminates all the processes associated with the tool.

Example:

|  |
| --- |
| ./stopOSWbb.sh |

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Diagnostic Data Output

As stated above, when oswbb is started for the first time it creates the archive subdirectory under the oswbb installation directory. The archive directory contains a minimum of 7 subdirectories, one for each data collector. These directories are named oswiostat, oswmpstat, oswnetstat, oswifconfig, oswprvtnet, oswps, oswtop, and oswvmstat. If you are running Linux, 3 additional directories will exist: oswmeminfo, oswslabinfo and oswcpuinfo. If you are running HP-UX 1 additional directory will exist: oswsar. If you create a [private.net](http://private.net/) file or it is created automatically on startup, then an additional directory named oswprvtnet will be created which stores the results of running traceroute on the rac private interconnects specified in [private.net](http://private.net/).

One file per hour will be generated in each of the OSWatcher utility subdirectories A new file is created at the top of each hour during the time that oswbb is running. The file will be in the following format:

|  |
| --- |
| <node\_name>\_<OS\_utility>\_YY.MM.DD.HH24.dat |

Details about each type of data file can be viewed by clicking on the below links:

[oswiostat](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswiostat)

[oswmpstat](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswmpstat)

[oswnetstat](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswnetstat)

[oswprvtnet](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswprvtnet)

[oswifconfig](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswifconfig)

[oswps](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswps)

[oswtop](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswtop)

[oswvmstat](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#oswvmstat)

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

oswiostat

<node\_name>\_iostat\_YY.MM.DD:HH24.dat

These files will contain output from the 'iostat' command that is obtained and archived by OSWatcher at specified intervals.  These files will only exist if 'iostat' is installed on the OS and if the oswbb user has privileges to run the utility. Please keep in mind that what gets reported in iostat may be different depending upon you platform. You should refer to your OS iostat man pages for the most accurate up to date descriptions of these fields

The iostat command is used for monitoring system input/output device loading by observing the time the physical disks are active in relation to their average transfer rates. This information can be used to change system configuration to better balance the input/output load between physical disks and adapters.

The iostat utility is fairly standard across UNIX platforms, but really only useful for those platforms that support extended disk statistics: AIX, Solaris and Linux. Also each platform will have a slightly different version of the iostat utility. You should consult your operating system man pages for specifics. The sample provided below is for Solaris.

oswbb runs the iostat utility at the specified interval and stores the data in the oswiostat subdirectory under the archive directory. The data is stored in hourly archive files. Each entry in the file contains a timestamp prefixed by \*\*\* embedded in the iostat output. Notice there is one entry for each timestamp.

|  |
| --- |
| Sample iostat file produced by oswbb |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | extended device statistics |  |  |  |  |  |  |  |  |  |  | | r/s | w/s | kr/s | kw/s | wait | actv | wsvc\_t | asvc\_t | %w | %b | device | | 0.0 | 0.3 | 0.0 | 2.1 | 0.0 | 0.0 | 3.4 | 0.8 | 0 | 0 | c0t0d0 | | 0.0 | 2.1 | 0.1 | 12.9 | 0.0 | 0.0 | 0.6 | 0.4 | 0 | 0 | c0t2d0 | | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 | 0 | fd0 | | 2.9 | 1.2 | 240.8 | 1.5 | 0.0 | 0.1 | 0.0 | 13.3 | 0 | 5 | c1t0d0 | | 1.1 | 0.8 | 18.0 | 8.8 | 0.0 | 0.0 | 0.1 | 5.9 | 0 | 1 | c1t1d0 | | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 | 0 | c0t1d0 | |

Field Descriptions

The iostat output contains summary information for all devices.

|  |  |
| --- | --- |
| Field | Description |
| r/s | Shows the number of reads/second |
| w/s | Shows the number of writes/second |
| kr/s | Shows the number of kilobytes read/second |
| kw/s | Shows the number of kilobytes written/second |
| wait | Average number of transactions waiting for service (queue length) |
| actv | Average number of transactions actively being serviced |
| wsvc\_t | Average service time in wait queue, in milliseconds |
| asvc\_t | Average service time of active transactions, in milliseconds |
| %w | Percent of time there are transactions waiting for service |
| %b | Percent of time the disk is busy |
| device | Device name |

What to look for

Average service times greater than 20msec for long duration.

High average wait times.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

oswmpstat

<node\_name>\_mpstat\_YY.MM.DD:HH24.dat

These files will contain output from the 'mpstat' command that is obtained and archived by OSWatcher at specified intervals.  These files will only exist if 'mpstat' is installed on the OS and if the oswbb user has privileges to run the utility. Please keep in mind that what gets reported in mpstat may be different depending upon you platform. You should refer to your OS mpstat man pages for the most accurate up to date descriptions of these fields

The mpstat command collects and displays performance statistics for all logical CPUs in the system.

The mpstat utility is fairly standard across UNIX platforms. Each platform will have a slightly different version of the mpstat utility. You should consult your operating system man pages for specifics. The sample provided below is for Solaris.

oswbb runs the mpstat utility at the specified interval and stores the data in the oswmpstat subdirectory under the archive directory. The data is stored in hourly archive files. Each entry in the file contains a timestamp prefixed by \*\*\* embedded in the mpstat output. Notice there are 2 entries for each timestamp. You should always ignore the first entry as this entry is always invalid.

|  |
| --- |
| Sample mpstat file produced by oswbb |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | \*\*\*Fri Jan 28 12:50:36 EST 2005 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | CPU | minf | mjf | xcal | intr | ithr | csw | icsw | migr | smtx | srw | syscl | usr | sys | wt | idl | | 0 | 0 | 0 | 0 | 483 | 383 | 118 | 1 | 0 | 0 | 0 | 64 | 0 | 0 | 0 | 100 | | 0 | 1268 | 0 | 0 | 486 | 382 | 414 | 42 | 0 | 0 | 0 | 2902 | 8 | 24 | 0 | 68 | | 0 | 4 | 0 | 0 | 479 | 379 | 144 | 3 | 0 | 0 | 0 | 96 | 0 | 0 | 0 | 100 | |

Field Descriptions

|  |  |
| --- | --- |
| Field | Description |
| cpu | Processor ID |
| minf | Minor faults |
| mif | Major Faults |
| xcal | Processor cross-calls (when one CPU wakes up another by interrupting it). |
| intr | Interrupts |
| ithr | Interrupts as threads (except clock) |
| csw | Context switches |
| icsw | Involuntary context switches |
| migr | Thread migrations to another processor |
| smtx | Number of times a CPU failed to obtain a mutex |
| srw | Number of times a CPU failed to obtain a read/write lock on the first try |
| syscl | Number of system calls |
| usr | Percentage of CPU cycles spent on user processes |
| sys | Percentage of CPU cycles spent on system processes |
| wt | Percentage of CPU cycles spent waiting on event |
| idl | Percentage of unused CPU cycles or idle time when the CPU is basically doing nothing |

What to look for

Involuntary context switches (this is probably the more relevant statistic when examining performance issues.)

Number of times a CPU failed to obtain a mutex. Values consistently greater than 200 per CPU causes system time to increase.

xcal is very important, show processor migration

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

oswnetstat

<node\_name>\_netstat\_YY.MM.DD:HH24.dat

These files will contain output from the 'netstat' command that is obtained and archived by OSWatcher at specified intervals.  These files will only exist if 'netstat' is installed on the OS and if the oswbb user has privileges to run the utility. Please keep in mind that what gets reported in netstat may be different depending upon you platform. You should refer to your OS netstat man pages for the most accurate up to date descriptions of these fields

The netstat command displays current TCP/IP network connections and protocol statistics.

The netstat utility is standard across UNIX platforms. Each platform will have a slightly different version of the netstat utility. You should consult your operating system man pages for specifics. The sample provided below is for Solaris.

oswbb runs the netstat utility at the specified interval and stores the data in the oswnetstat subdirectory under the archive directory. The data is stored in hourly archive files. Each entry in the file contains a timestamp prefixed by \*\*\* embedded in the netstat output.

The netstat utility has many command line flags, and the most commonly used to troubleshoot RAC is "ia(n)" for the interface level output and "s" for the protocol level statistics. The following are examples for the two different command parameters.

The command line options "-ain" have these effects:

|  |  |
| --- | --- |
| Option | Description |
| -a | The command output will use the logical names of the interface. It will also report the name of the IP address found through normal IP address resolution methods. |
| -i | This triggers the Interface specific statistics, the columns of which are outlined in table [bla-KR] |
| -n | This causes the output to use IP addresses instead of the resolved names |

Example netstat file produced by oswbb:

|  |
| --- |
| Sample netstat file produced by oswbb |
| |  | | --- | | \*\*\*Fri Jan 28 12:50:36 EST 2005 | | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Name | Mtu | Net/Dest | Address | Ipkts | Ierrs | Opkts | Oerrs | Collis | Queue | | lo0 | 8232 | 127.0.0.0 | 127.0.0.1 | 296065 | 0 | 296065 | 0 | 0 | 0 | | eri0 | 1500 | 138.1.140.0 | 138.1.140.96 |  | 0 | 176244 | 2 | 191951 | 0 |      |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | RAWIP |  |  |  |  |  |  |  | |  | rawipInDatagrams | = | 0 |  | rawipInErrors | = | 0 | |  | rawipInCksumErrs | = | 0 |  | rawipOutDatagrams | = | 0 | |  | rawipOutErrors | = | 0 |  |  |  |  | | UDP |  |  |  |  |  |  |  | |  | udpInDatagrams | = | 295719 |  | udpInErrors | = | 0 | |  | udpOutDatagrams | = | 295671 |  | udpOutErrors | = | 0 | | TCP |  |  |  |  |  |  |  | |  | tcpRtoAlgorithm | = | 4 |  | tcpRtoMin | = | 400 | |  | tcpRtoMax | = | 60000 |  | tcpMaxConn | = | -1 | |  | tcpActiveOpens | = | 27 |  | tcpPassiveOpens | = | 21 | |  | tcpAttemptFails | = | 6 |  | tcpEstabResets | = | 0 | |  | tcpCurrEstab | = | 15 |  | tcpOutSegs | = | 691 | |  | tcpOutDataSegs | = | 479 |  | tcpOutDataBytes | = | 43028 | |  | tcpRetransSegs | = | 0 |  | tcpRetransBytes | = | 0 | |  | tcpOutAck | = | 212 |  | tcpOutAckDelayed | = | 83 | |  | tcpOutUrg | = | 0 |  | tcpOutWinUpdate | = | 0 | |  | tcpOutWinProbe | = | 0 |  | tcpOutControl | = | 85 | |  | tcpOutRsts | = | 10 |  | tcpOutFastRetrans |  |  | |  | tcpInSegs | = | 915 |  |  | = | 0 | |  | tcpInAckSegs | = | 489 |  | tcpInAckBytes | = | 43023 | |  | tcpInDupAck | = | 42 |  | tcpInAckUnsent | = | 0 | |  | tcpInInorderSegs | = | 477 |  | tcpInInorderBytes | = | 40640 | |  | tcpInUnorderSegs | = | 0 |  | tcpInUnorderBytes | = | 0 | |  | tcpInDupSegs | = | 0 |  | tcpInDupBytes | = | 0 | |  | tcpInPartDupSegs | = | 0 |  | tcpInPartDupBytes | = | 0 | |  | tcpInPastWinSegs | = | 0 |  | tcpInPastWinBytes | = | 0 | |  | tcpInWinProbe | = | 0 |  | tcpInWinUpdate | = | 0 | |  | tcpInClosed | = | 0 |  | tcpRttNoUpdate | = | 0 | |  | tcpRttUpdate | = | 462 |  | tcpTimRetrans | = | 0 | |  | tcpTimRetransDrop | = | 0 |  | tcpTimKeepalive | = | 80 | |  | tcpTimKeepaliveProbe | = | 0 |  | tcpTimKeepaliveDrop | = | 0 | |  | tcpListenDrop | = | 0 |  | tcpListenDropQ0 | = | 0 | |  | tcpHalfOpenDrop | = | 0 |  | tcpOutSackRetrans | = | 0 | | IPv4 |  |  |  |  |  |  |  | |  | ipForwarding | = | 2 |  | ipDefaultTTL | = | 255 | |  | ipInReceives | = | 17858585 |  | ipInHdrErrors | = | 0 | |  | ipInAddrErrors | = | 0 |  | ipInCksumErrs | = | 0 | |  | ipForwDatagrams | = | 0 |  | ipForwProhibits | = | 0 | |  | ipInUnknownProtos | = | 0 |  | ipInDiscards | = | 0 | |  | ipInDelivers | = | 296623 |  | ipOutRequests | = | 17624403 | |  | ipOutDiscards | = | 0 |  | ipOutNoRoutes | = | 827 | |  | ipReasmTimeout | = | 60 |  | ipReasmReqds | = | 0 | |  | ipReasmOKs | = | 0 |  | ipReasmFails | = | 0 | |  | ipReasmDuplicates | = | 0 |  | ipReasmPartDups | = | 0 | |  | ipFragOKs | = | 0 |  | ipFragFails | = | 0 | |  | ipFragCreates | = | 0 |  | ipRoutingDiscards | = | 0 | |  | tcpInErrs | = | 0 |  | udpNoPorts | = | 225722 | |  | udpInCksumErrs | = | 0 |  | udpInOverflows | = | 0 | |  | rawipInOverflows | = | 0 |  | ipsecInSucceeded | = | 0 | |  | ipsecInFailed | = | 0 |  | ipInIPv6 | = | 0 | |  | ipOutIPv6 | = | 0 |  | ipOutSwitchIPv6 | = | 5 | | IPv6 |  |  |  |  |  |  |  | |  | ipv6Forwarding | = | 2 |  | ipv6DefaultHopLimit | = | 255 | |  | ipv6InReceives | = | 0 |  | ipv6InHdrErrors | = | 0 | |  | ipv6InTooBigErrors | = | 0 |  | ipv6InNoRoutes | = | 0 | |  | ipv6InAddrErrors | = | 0 |  | ipv6InUnknownProtos | = | 0 | |  | ipv6InTruncatedPkts | = | 0 |  | ipv6InDiscards | = | 0 | |  | ipv6InDelivers | = | 0 |  | ipv6OutForwDatagrams | = | 0 | |  | ipv6OutRequests | = | 0 |  | ipv6OutDiscards | = | 0 | |  | ipv6OutNoRoutes | = | 0 |  | ipv6OutFragOKs | = | 0 | |  | ipv6OutFragFails | = | 0 |  | ipv6OutFragCreates | = | 0 | |  | ipv6ReasmReqds | = | 0 |  | ipv6ReasmOKs | = | 0 | |  | ipv6ReasmFails | = | 0 |  | ipv6InMcastPkts | = | 0 | |  | ipv6OutMcastPkts | = | 0 |  | ipv6ReasmDuplicates | = | 0 | |  | ipv6ReasmPartDups | = | 0 |  | ipv6ForwProhibits | = | 0 | |  | udpInCksumErrs | = | 0 |  | udpInOverflows | = | 0 | |  | rawipInOverflows | = | 0 |  | ipv6InIPv4 | = | 0 | |  | ipv6OutIPv4 | = | 0 |  | ipv6OutSwitchIPv4 | = | 0 | | ICMPv4 |  |  |  |  |  |  |  | |  | icmpInMsgs | = | 17624914 |  | icmpInErrors | = | 0 | |  | icmpInCksumErrs | = | 0 |  | icmpInUnknowns | = | 0 | |  | icmpInDestUnreachs | = | 72 |  | icmpInTimeExcds | = | 0 | |  | icmpInParmProbs | = | 0 |  | icmpInSrcQuenchs | = | 0 | |  | icmpInRedirects | = | 0 |  | icmpInBadRedirects | = | 0 | |  | icmpInEchos | = | 17624842 |  | icmpInEchoReps | = | 0 | |  | icmpInTimestamps | = | 0 |  | icmpInTimestampReps | = | 0 | |  | icmpInAddrMasks | = | 0 |  | icmpInAddrMaskReps | = | 0 | |  | icmpInFragNeeded | = | 0 |  | icmpOutMsgs | = | 17624920 | |  | icmpOutDrops | = | 225716 |  | icmpOutErrors | = | 0 | |  | icmpOutDestUnreachs | = | 78 |  | icmpOutTimeExcds | = | 0 | |  | icmpOutParmProbs | = | 0 |  | icmpOutSrcQuenchs | = | 0 | |  | icmpOutRedirects | = | 0 |  | icmpOutEchos | = | 0 | |  | icmpOutEchoReps | = | 17624842 |  | icmpOutTimestamps | = | 0 | |  | icmpOutTimestampReps | = | 0 |  | icmpOutAddrMasks | = | 0 | |  | icmpOutAddrMaskReps | = | 0 |  | icmpOutFragNeeded | = | 0 | |  | icmpInOverflows | = | 0 |  |  |  |  | | ICMPv6 |  |  |  |  |  |  |  | |  | icmp6InMsgs | = | 0 |  | icmp6InErrors | = | 0 | |  | icmp6InDestUnreachs | = | 0 |  | icmp6InAdminProhibs | = | 0 | |  | icmp6InTimeExcds | = | 0 |  | icmp6InParmProblems | = | 0 | |  | icmp6InPktTooBigs | = | 0 |  | icmp6InEchos | = | 0 | |  | icmp6InEchoReplies | = | 0 |  | icmp6InRouterSols | = | 0 | |  | icmp6InRouterAds | = | 0 |  | icmp6InNeighborSols | = | 0 | |  | icmp6InNeighborAds | = | 0 |  | icmp6InRedirects | = | 0 | |  | icmp6InBadRedirects | = | 0 |  | icmp6InGroupQueries | = | 0 | |  | icmp6InGroupResps | = | 0 |  | icmp6InGroupReds | = | 0 | |  | icmp6InOverflows | = | 0 |  |  |  |  | |  | icmp6OutMsgs | = | 0 |  | icmp6OutErrors | = | 0 | |  | icmp6OutDestUnreachs | = | 0 |  | icmp6OutAdminProhibs | = | 0 | |  | icmp6OutTimeExcds | = | 0 |  | icmp6OutParmProblems | = | 0 | |  | icmp6OutPktTooBigs | = | 0 |  | icmp6OutEchos | = | 0 | |  | icmp6OutEchoReplies | = | 0 |  | icmp6OutRouterSols | = | 0 | |  | icmp6OutRouterAds | = | 0 |  | icmp6OutNeighborSols | = | 0 | |  | icmp6OutNeighborAds | = | 0 |  | icmp6OutRedirects | = | 0 | |  | icmp6OutGroupQueries | = | 0 |  | icmp6OutGroupResps | = | 0 | |  | icmp6OutGroupReds | = | 0 |  |  |  |  |      |  |  |  |  | | --- | --- | --- | --- | | IGMP: |  |  |  | |  | 2490 |  | messages received | |  | 0 |  | messages received with too few bytes | |  | 0 |  | messages received with bad checksum | |  | 2490 |  | membership queries received | |  | 0 |  | membership queries received with invalid field(s) | |  | 0 |  | membership reports received | |  | 0 |  | membership reports received with invalid field(s) | |  | 0 |  | membership reports received for groups to which we belong | |  |  |  |  | |  | 0 |  | membership reports sent | | |

Field Descriptions:

The netstat output produced by oswbb contains 2 sections. The first section contains information about all the network interfaces. The second section contains information about per-protocol statistics.

Section 1: Netstat -ain

|  |  |
| --- | --- |
| Field | Description |
| name | Device name of interface |
| Mtu | Maximum transmission unit |
| Net | Network Segment Address |
| address | Network address of the device |
| ipkts | Input packets |
| Ierrs | Input errors |
| opkts | Output Packets |
| Oerrs | Output errors |
| collis | Collisions |
| queue | Number in the Queue |

Section 2: Protocol Statistics

The per-protocol statistics can be divided into several categories:

RAWIP (raw IP) packets

TCP packets

IPv4 packets

ICMPv4 packets

IPv6 packets

ICMPv6 packets

UDP packets

IGMP packet

Each protocol type has a specific set of measures associated with it. Network analysis requires evaluation of these measurements on an individual level and all together to examine the overall health of the network communications.

The TCP protocol is used the most in Oracle database and applications. Some implementations for RAC use UDP for the interconnect protocol instead of TCP. The statistics cannot be divided up on a per-interface basis, so these should be compared to the "-i" statistics above.

What to look for:

Section 1

The information in Section 1 will help diagnose network problems when there is connectivity but response is slow.

Values to look at:

Collisions (Collis)

Output packets (Opkts)

Input errors (Ierrs)

Input packets (Ipkts)

The above values will give information to workout network collision rates as follows:

Network collision rate = Output collision / Output packets

For a switched network, the collisions should be 0.1 percent or less (see the [Cisco web site](http://www.cisco.com/en/US/customer/products/hw/optical/ps2011/prod_troubleshooting_guide_chapter09186a00801aa0cf.html) as a reference) of the output packets. Excessive collisions could lead to the switch port the interface is plugged into to segment, or pull itself off-line, amongst other switch-related issues.

For the input error statistics:

Input Error Rate = Ierrs / Ipkts.

If the input error rate is high (over 0.25 percent), the host is excessively dropping packets. This could mean there is a mismatch of the duplex or speed  settings of the interface card and switch.  It could also imply a failed patch cable.

If ierrs or oerrs show an excessive amount of errors, more information can be found by examination of the netstat -s output.

For Sun systems, further information about a specific interface can be found by using the "-k" option for netstat. The output will give fuller statistics for the device, but this option is not mentioned in the netstat man page.

Section 2

The information in Section 2 contains the protocol statistics.

Many performance problems associated with the network involve the retransmission of the TCP packets.

To find the segment retransmission rate:

%segment-retrans=(tcpRetransSegs / tcpOutDataSegs) \* 100

To find the byte retransmission rate:

%byte-retrans = ( tcpRetransBytes / tcpOutDataBytes ) \* 100

Most network analyzers report TCP retransmissions as segments (frames) and not in bytes.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

oswprvtnet

<node\_name>\_prvtnet\_YY.MM.DD:HH24.dat

These files will contain output from running the '[private.net](http://private.net/) 'script that must be created first by the customer. A template for what this file should look like is supplied in the oswbb directory and is named [Exampleprivate.net](http://exampleprivate.net/). A new file named [private.net](http://private.net/) needs to be created based on the sample file first and then granted execute priviledge. You should test this file works by executing it standalone (./[private.net](http://private.net/)). oswbb will then execute this file along with the other data collectors.

Information about the status of RAC private networks should be collected. This requires the user to manually add entries for these private networks into the [private.net](http://private.net/) file located in the base oswbb directory. Instructions on how to do this are contained in the README file.

oswbb uses the traceroute command to obtain the status of these private networks. Each operating system uses slightly different arguments to the traceroute command. Examples of the syntax to use for each operating system are contained in the sample [Exampleprivate.net](http://exampleprivate.net/) file located in the base oswbb directory. This will result in the output appearing differently across UNIX platforms. oswbb runs the [private.net](http://private.net/) file at the specified interval and stores the data in the oswprvtnet subdirectory under the archive directory. The data is stored in hourly archive files. Each entry in the file contains a timestamp prefixed by \*\*\* embedded in the top output.

|  |
| --- |
| Sample file produced by oswbb |
| |  | | --- | | \*\*\*Fri Jan 28 12:50:36 EST 2005 | | traceroute to [celdecclu2.us.oracle.com](http://celdecclu2.us.oracle.com/) (138.2.71.112): 1-30 hops  (initial packetsize = 1500)    1  [celdecclu2.us.oracle.com](http://celdecclu2.us.oracle.com/) (138.2.71.112) 1.95ms  2.92 ms 1.95 ms | |

What to Look For

Example 1:  Interface is up and responding:

|  |
| --- |
| traceroute to X.X.X.X, (X.X.X.X) 30 hops max, 1492 byte packets  1 X.X.X.X 1.015 ms 0.766 ms 0.755 ms |

Example 2:  Target interface is not on a directly connected network, so validate that the address is correct or the switch it is plugged in is on the same VLAN (or other issue):

|  |
| --- |
| traceroute to X.X.X.X, (X.X.X.X) 30 hops max, 40 byte packets  traceroute: host X.X.X.X is not on a directly-attached network |

Example 3:  Network is unreachable:

|  |
| --- |
| traceroute to X.X.X.X, (X.X.X.X) 30 hops max, 40 byte packets  Network is unreachable |

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

oswifconfig

<node\_name>\_ifconfig\_YY.MM.DD:HH24.dat

These files will contain output from the 'ifconfig -a' command that is obtained and archived by OSWatcher at specified intervals.  These files will only exist if 'ifconfig' is available on the OS and if the oswbb user has privileges to run the utility. Please keep in mind that what gets reported in ifconfig may be different depending upon you platform. You should refer to your OS ifconfig man pages for the most accurate up to date descriptions of these fields

The ifconfig command displays the current status of network interfaces.

The ifconfig utility is standard across UNIX platforms. Each platform will have a slightly different version of the ifconfig utility. You should consult your operating system man pages for specifics. The sample provided below is for Linux.

oswbb runs the ifconfig utility at the specified interval and stores the data in the oswifconfig subdirectory under the archive directory. The data is stored in hourly archive files. Each entry in the file contains a timestamp prefixed by \*\*\* embedded in the ifconfig output.

The ifconfig -a command utility is most commonly used to troubleshoot RAC network interface issues. The output of this command is used with the output of netstat and [private.net](http://private.net/) to determine any network interface issues that may exist on your server.

|  |
| --- |
| Sample file produced by oswbb |
| |  | | --- | | \*\*\*Tue Apr 29 12:50:36 EST 2014 | | eth0 Link encap:Ethernet HWaddr 00:16:3E:66:14:00  inet addr:10.141.154.225 Bcast:10.141.154.255 Mask:255.255.254.0  inet6 addr: fe80::216:3eff:fe66:1400/64 Scope:Link  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  RX packets:8098395 errors:0 dropped:0 overruns:0 frame:0  TX packets:35772 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:1000  RX bytes:609160321 (580.9 MiB) TX bytes:17141198 (16.3 MiB) | |

What to Look For

Example 1:  Interface is up and responding:

|  |
| --- |
| UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 |

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

oswps

<node\_name>\_ps\_YY.MM.DD:HH24.dat

These files will contain output from the 'ps' command that is obtained and archived by OSWatcher at specified intervals.  These files will only exist if 'ps' is installed on the OS and if the oswbb user has privileges to run the utility. Please keep in mind that what gets reported in ps may be different depending upon you platform. You should refer to your OS ps man pages for the most accurate up to date descriptions of these fields

The ps (process state) command list all the processes currently running on the system and provides information about CPU consumption, process state, priority of the process, etc. The ps command has a number of options to control which processes are displayed, and how the output is formatted. oswbb runs the ps command with the -elf option.

The ps command is fairly standard across UNIX platforms Each platform will have a slightly different version of the ps utility. You should consult your operating system man pages for specifics. The sample provided below is for Solaris.

oswbb runs the ps command at the specified interval and stores the data in the oswps subdirectory under the archive directory. The data is stored in hourly archive files. Each entry in the file contains a timestamp prefixed by \*\*\* embedded in the ps output.

|  |
| --- |
| Sample ps file produced by oswbb |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | \*\*\*Wed Feb 2 09:26:54 EST 2005 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | F | S | UID | PID | PPID | C | PRI | NI | ADDR | SZ | WCHAN | STIME | TTY | TIME | CMD | | 19 | T | root | 0 | 0 | 0 | 0 | SY | ? | 0 |  | Jan 31 | ? | 0:13 | sched | | 8 | S | root | 1 | 0 | 0 | 41 | 20 | ? | 107 | ? | Jan 31 | ? | 0:00 | /etc | | 19 | S | root | 2 | 0 | 0 | 0 | SY | ? | 0 | ? | Jan 31 | ? | 0:00 | page | | 19 | S | root | 3 | 0 | 0 | 0 | SY | ? | 0 | ? | Jan 31 | ? | 0:50 | fsflu | | 8 | S | root | 355 | 1 | 0 | 41 | 20 | ? | 232 | ? | Jan 31 | ? | 0:00 | /usr/ | | 8 | S | root | 297 | 296 | 0 | 41 | 20 | ? | 379 | ? | Jan 31 | ? | 0:00 | htt\_s | | 8 | S | cedavis | 391 | 381 | 0 | 89 | 20 | ? | 301 | ? | Jan 31 | ? | 0:00 | /usr/ | |

Field Descriptions

|  |  |
| --- | --- |
| Field | Description |
| f | Flags s State of the process |
| uid | The effective user ID number of the process |
| pid | The process ID of the process |
| ppid | The process ID of the parent process. |
| d | Processor utilization for scheduling (obsolete). |
| pri | The priority of the process. |
| ni | Nice value, used in priority computation. |
| addr | The memory address of the process. |
| sz | The total size of the process in virtual memory, including all mapped files and devices, in pages. |
| wchan | The address of an event for which the process is sleeping (if blank, the process is running). |
| stime | The starting time of the process, given in hours, minutes, and seconds. |
| tty | The controlling terminal for the process (the message ?, is printed when there is no controlling terminal). |
| time | The cumulative execution time for the process. |
| cmd | The command name process is executing. |

What to look for

The information in the ps command will primarily be used as supporting information for RAC diagnostics. If for example, the status of a process prior to a system crash may be important for root cause analysis. The amount of memory a process is consuming is another example of how this data can be used.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

oswtop

<node\_name>\_top\_YY.MM.DD:HH24.dat

These files will contain output from the 'top' command that is obtained and archived by OSWatcher at specified intervals.  These files will only exist if 'top' is installed on the OS and if the oswbb user has privileges to run the utility. Please keep in mind that what gets reported in top may be different depending upon you platform. You should refer to your OS top man pages for the most accurate up to date descriptions of these fields

Top is a program that will give continual reports about the state of the system, including a list of the top CPU using processes. Top has three primary design goals:

provide an accurate snapshot of the system and process state,

not be one of the top processes itself,

be as portable as possible.

Each operating system uses a different version of the UNIX utility top. This will result in the top output appearing differently across UNIX platforms. You should consult your operating system man pages for specifics. The sample provided below is for Solaris.

oswbb runs the top utility at the specified interval and stores the data in the oswtop subdirectory under the archive directory. The data is stored in hourly archive files. Each entry in the file contains a timestamp prefixed by \*\*\* embedded in the top output.

|  |
| --- |
| Sample top file produced by oswbb |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | \*\*\*Fri Jan 28 12:50:36 EST 2005  load averages: 0.11, 0.07, 0.06 12:50:36  136 processes: 133 sleeping, 2 running, 1 on cpu    Memory: 2048M real, 1061M free, 542M swap in use, 1605M swap free |  |  |  |  |  |  |  |  |  |  | | PID | USERNAME | THR | PRI | NICE | SIZE | RES | STATE | TIME | CPU | COMMAND | | 704 | cedavis | 16 | 49 | 0 | 346M | 276M | sleep | 222:33 | 3.51% | java | | 362 | root | 1 | 59 | 0 | 34M | 75M | sleep | 11:49 | 0.21% | Xsun | | 20675 | cedavis | 1 | 0 | 0 | 1584K | 1064K | cpu | 0:00 | 19% | top | | 20640 | cedavis | 1 | 0 | 0 | 1904K | 1240K | sleep | 0:00 | 0.14% | OSWatcher.sh | | 20657 | cedavis | 1 | 20 | 0 | 1904K | 1240K | sleep | 0:00 | 0.14% | oswsub.sh | | 16881 | cedavis | 1 | 59 | 0 | 199M | 159K | sleep | 23:04 | 0.10% | oracle | | 20671 | cedavis | 1 | 0 | 0 | 1904K | 1240K | run | 0:00 | 0.09% | oswsub.sh | | 20653 | cedavis | 1 | 0 | 0 | 1904K | 1240K | sleep | 0:00 | 0.09% | OSWatcherFM.sh | | 20665 | cedavis | 1 | 0 | 0 | 1904K | 1240K | sleep | 0:00 | 0.09% | oswsub.sh | | 20672 | cedavis | 1 | 0 | 0 | 1264K | 1031K | sleep | 0:00 | 0.09% | iostat | | 20659 | cedavis | 1 | 10 | 0 | 1904K | 1240K | sleep | 0:00 | 0.09% | oswsub.sh | | 20661 | cedavis | 1 | 30 | 0 | 1096K | 880K | sleep | 0:00 | 0.09% | vmstat | | 20668 | cedavis | 1 | 0 | 0 | 1904K | 1240K | run | 0:00 | 0.05% | oswsub.sh | | 20674 | cedavis | 1 | 0 | 0 | 968K | 624K | sleep | 0:00 | 0.05% | sleep | | 20663 | cedavis | 1 | 20 | 0 | 1080K | 864K | sleep | 0:00 | 0.05% | mpstat | |

Field Descriptions

load averages: 0.11, 0.07, 0.06 12:50:36

This line displays the load averages over the last 1, 5 and 15 minutes as well as the system time. This is quite handy as top basically includes a timestamp along with the data capture.

Load average is defined as the average number of processes in the run queue. A runnable Unix process is one that is available right now to consume CPU resources and is not blocked on I/O or on a system call. The higher the load average, the more work your machine is doing.

The three numbers are the average of the depth of the run queue over the last 1, 5, and 15 minutes. In this example we can see that .11 processes were on the run queue on average over the last minute, .07 processes on average on the run queue over the last 5 minutes, etc. It is important to determine what the average load of the system is through benchmarking and then look for deviations. A dramatic rise in the load average can indicate a serious performance problem.

136 processes: 133 sleeping, 2 running, 1 on cpu

This line displays the total number of processes running at the time of the last update. It also indicates how many Unix processes exist, how many are sleeping (blocked on I/O or a system call), how many are stopped (someone in a shell has suspended it), and how many are actually assigned to a CPU. This last number will not be greater than the number of processors on the machine, and the value should also correlate to the machine's load average provided the load average is less than the number of CPUs. Like load average, the total number of processes on a healthy machine usually varies just a small amount over time. Suddenly having a significantly larger or smaller number of processes could be a warning sign.

Memory: 2048M real, 1061M free, 542M swap in use, 1605M swap free

The "Memory:" line is very important. It reflects how much real and swap memory a computer has, and how much is free. "Real" memory is the amount of RAM installed in the system, a.k.a. the "physical" memory. "Swap" is virtual memory stored on the machine's disk.

Once a computer runs out of physical memory, and starts using swap space, its performance deteriorates dramatically. If you run out of swap, you'll likely crash your programs or the OS.

Individual process fields

|  |  |
| --- | --- |
| Field | Description |
| PID | Process ID of process |
| USERNAME | Username of process |
| THR | Process thread PRI Priority of process |
| NICE | Nice value of process |
| SIZE | Total size of a process, including code and data, plus the stack space in kilobytes |
| RES | Amount of physical memory used by the process |
| STATE | Current CPU state of process. The states can be S for sleeping, D for uninterrupted, R for running, T for stopped/traced, and Z for zombied |
| TIME | The CPU time that a process has used since it started |
| %CPU | The CPU time that a process has used since the last update |
| COMMAND | The task's command name |

What to Look For

Large run queue. Large number of processes waiting in the run queue may be an indication that your system does not have sufficient CPU capacity.

Process consuming lots of CPU. A process which is "hogging" CPU is always suspect. If this process is an oracle foreground process it's most likely running an expensive query that should be tuned. Oracle background process should not hog CPU for long periods of time.

High load averages. Processes should not be backed up on the run queue for extended periods of time.

Low swap space. This is an indication you are running low on memory.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

oswvmstat

<node\_name>\_vmstat\_YY.MM.DD:HH24.dat

These files will contain output from the 'vmstat' command that is obtained and archived by OSWatcher at specified intervals.  These files will only exist if 'vmstat' is installed on the OS and if the oswbb user has privileges to run the utility. Please keep in mind that what gets reported in vmstat may be different depending upon you platform. You should refer to your OS vmstat man pages for the most accurate up to date descriptions of these fields.

The name vmstat comes from "report virtual memory statistics".  The vmstat utility does a bit more than this, though. In addition to reporting virtual memory, vmstat reports certain kernel statistics about processes, disk, trap, and CPU activity.

The vmstat utility is fairly standard across UNIX platforms. Each platform will have a slightly different version of the vmstat utility. You should consult your operating system man pages for specifics. The sample provided below is for Solaris.

oswbb runs the vmstat utility at the specified interval and stores the data in the oswvmstat subdirectory under the archive directory. The data is stored in hourly archive files. Each entry in the file contains a timestamp prefixed by \*\*\* embedded in the vmstat output.

|  |
| --- |
| Sample vmstat file produced by oswbb |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | \*\*\*Fri Jan 28 12:50:36 EST 2005 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | procs |  |  | memory |  | page |  |  |  |  |  |  | disk |  |  |  | faults |  |  | cpu |  |  | | r | b | w | swap | free | re | mf | pi | po | fr | de | sr | dd | f0 | s0 |  | in | sy | cs | us | sy | id | | 0 | 0 | 0 | 1761344 | 1246520 | 1 | 6 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 380 | 1364 | 900 | 4 | 1 | 95 | | 0 | 0 | 0 | 1643920 | 1086776 | 331 | 1485 | 8 | 16 | 16 | 0 | 0 | 31 | 0 | 0 | 0 | 447 | 4966 | 1315 | 15 | 31 | 54 | | 0 | 0 | 0 | 1643872 | 1086728 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 389 | 1472 | 932 | 0 | 0 | 100 | |

Field Descriptions

The vmstat output is actually broken up into six sections: procs, memory, page, disk, faults and CPU. Each section is outlined in the following table.

|  |  |
| --- | --- |
| Field | Description |
| PROCS |  |
| r | Number of processes that are in a wait state and basically not doing anything but waiting to run |
| b | Number of processes that were in sleep mode and were interrupted since the last update |
| w | Number of processes that have been swapped out by mm and vm subsystems and have yet to run |
| MEMORY |  |
| swap | The amount of swap space currently available free The size of the free list |
| PAGE |  |
| re | page reclaims |
| mf | minor faults |
| pi | kilobytes paged in |
| po | kilobytes paged out |
| fr | kilobytes freed |
| de | anticipated short-term memory shortfall (Kbytes) |
| sr | pages scanned by clock algorithm |
| DISK |  |
| Bi | Disk blocks sent to disk devices in blocks per second |
| FAULTS |  |
| In | Interrupts per second, including the CPU clocks |
| Sy | System calls |
| Cs | Context switches per second within the kernel |
| CPU |  |
| Us | Percentage of CPU cycles spent on user processes |
| Sy | Percentage of CPU cycles spent on system processes |
| Id | Percentage of unused CPU cycles or idle time when the CPU is basically doing nothing |

What to look for

The following information should be used as a guideline and not considered hard and fast rules. The information documented below comes from Adrian Cockcroft's book, Sun Performance Tuning. Other operating systems like HP and Linux may have different thresholds.

Large run queue. Adrian Cockcroft defines anything over 4 processes per CPU on the run queue as the threshold for CPU saturation. This is certainly a problem if this last for any long period of time.

CPU utilization. The amount of time spent running system code should not exceed 30% especially if idle time is close to 0%.

A combination of large run queue with no idle CPU is an indication the system has insufficient CPU capacity.

Memory bottlenecks are determined by the scan rate (sr) . The scan rate is the pages scanned by the clock algorithm per second. If the scan rate (sr) is continuously over 200 pages per second then there is a memory shortage.

Disk problems may be identified if the number of processes blocked exceeds the number of processes on run queue.

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Analyzing the Output

OSWatcher comes bundled with an analyzer (oswbba). This utility provides analysis and graphical capabilities. See the [oswbba User Guide](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1531223.1&id=461053.1)

|  |
| --- |
|  |

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Known Issues

Note that if you haven't installed net-tools on Oracle Linux 7 you may see warnings when starting OSWatcher that it can't find netstat and ifconfig. This appears to be OL 7 specific. You may get a warning that /proc/slabinfo does not exist. It does exist, but the permissions have changed to 0400 and the file is owned by root:root.

There may be issues with the analyzer parsing timestamps that are not standard English Language timestamps. Setting the parameter oswgCompliance=1 (default) should resolve this. There have been reported cases where this alone did not correct the problem. As a workaround try adding the LANG environment in startOSWbb.sh.

|  |
| --- |
| # set LANG environment  export LANG=en\_US.UTF8  # restart OSWatcher |

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Download

Download the latest version of oswbb by clicking on the download link provided in [Document 301137.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1531223.1&id=301137.1). The download link no longer exists from inside this document. If you are still unable to download the file, you may request that we email you a copy: [carl.davis@oracle.com](mailto:carl.davis@oracle.com?subject=Unix%20OSW%20Request:)

Note that you can also download oswbb in the RAC and DB Support Tools Bundle:

[Document 1594347.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1531223.1&id=1594347.1) RAC and DB Support Tools Bundle

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Reporting Feedback

If you encounter problems running OSWatcher which is not listed under the [Known Issue](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Known Issues) section or would like to provide comments/feedback about OSWatcher (including enhancement requests) please send email to [carl.davis@oracle.com](mailto:carl.davis@oracle.com?subject=Unix%20oswbb%20Feedback:)

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

Sending Files To Support

Two utilities have been provided to bundle OSWatcher data. We expect OSWatcher data to be collected and bundled in such a way as to maintain the existing directory structure created by running the tool. Failure to bundle OSWatcher data will prevent the automatic processing and analysis of the data and prolong the response you receive by opening an SR with support. To bundle the entire OSWatcher data collection use the script tar\_up\_full\_archive.sh. This script prompts you to enter the fully qualified path name of the archive directory and creates a tarball named osw\_archive.tar that you can upload to an SR. To bundle a specific time window, which can come in handy if you have a very large OSWatcher archive, or just want to send a subset of the data for a specific time window use script tar\_up\_partial\_archive.sh. This script prompts you to enter the fully qualified path name of the archive directory and then prompts for a starting and ending log. The script creates a tarball named osw\_archive.tar that you can upload to an SR.

|  |
| --- |
| ./tar\_up\_full\_archive.sh |

[Back to Contents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=451495462240406&id=1531223.1&_adf.ctrl-state=fk1s7g0s2_744#Contents)

[Legal Notices and Terms of Use](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1531223.1&id=225559.1)