www.phoronix-test-suite.com

Phoronix Test Suite v3.2.0m2 (Grimstad)

User Manual

User Options

The following options are currently supported by the Phoronix Test Suite client. A list of available options can also be found by running *phoronix-test-suite help.*

Test Installation

install *[Test | Suite | OpenBenchmarking.org ID | Test Result] ...*
This option will install the selected test(s) inside the testing environment directory). The install process from downloading of the test files to the installation is fully automated. The install option needs to be supplied with the test name or suite as an argument. Optionally, a OpenBenchmarking.org ID or the name of a saved results file can be supplied as well and the test(s) to install will automatically be extracted from that information. If the test is already installed and was run by the latest version of the installation process, no action will be taken. Multiple arguments can be supplied to install additional tests at the same time.

install-dependencies *[Test | Suite | OpenBenchmarking.org ID | Test Result] ...*
This option will install the external dependencies needed by the selected test(s) using the distribution's package management system. For example, some tests depend upon GCC for compiling code. If GCC is not detected on the system, the Phoronix Test Suite will attempt to install GCC using the distribution's package management system. If you're running this command as a local user, you may be prompted for the root password while the process is running. For unsupported distributions, the dependency names will be displayed along with common names for the package. The install-dependencies option needs to be supplied with the test name or suite as an argument. When using the install option, the external dependencies are automatically checked.

make-download-cache
This option will create a download cache for use by the Phoronix Test Suite.

remove-installed-test *[Test]*
This option will permanently remove a installed test by the Phoronix Test Suite.

Testing

benchmark *[Test | Suite | OpenBenchmarking.org ID | Test Result] ...*
This option will install the selected test(s) (if needed) and will proceed to run the test(s). This option is equivalent to running phoronix-test-suite with the install option followed by the run option. Multiple arguments can be supplied to run additional tests at the same time and save the results into one file.

finish-run *[Test Result]*
This option can be used if a test run had not properly finished running all tests within a saved results file. Using this option when specifying a saved results file where all tests had not completed will attempt to finish testing on the remaining tests where there are missing results.

recover-run *[Unknown Object]*
This option is similar to finish-run and can be used to finish running a set of tests that were started but

never finished. Using this option when specifying a saved results file where all tests had not completed will attempt to finish testing on the remaining tests where there are missing results. Differing from finish-run, recover-run can be used if the system had crashed during the testing process and the test results from before the crash happened can generally be recovered.

run *[Test | Suite | OpenBenchmarking.org ID | Test Result] ...*
This option will run the selected test(s).

run-tests-in-suite
This option can be used if you wish to run all of the tests found in a supplied suite, but you wish to re-configure each of the test options rather than using the defaults supplied by the suite.

Batch Testing

batch-benchmark *[Test | Suite | OpenBenchmarking.org ID | Test Result] ...*
This option and its arguments are equivalent to the benchmark option, but the process will be run in the Phoronix Test Suite batch mode.

batch-install *[Test | Suite | OpenBenchmarking.org ID | Test Result] ...*
If you wish to run the install process in the Phoronix Test Suite batch mode but do not wish to run any tests at this time. Running the install process in the batch mode will use the default values and not prompt the user of any possible options, to ensure the process is fully automated.

batch-run *[Test | Suite | OpenBenchmarking.org ID | Test Result] ...*
This option and its arguments are equivalent to the run option, but the process will be run in the Phoronix Test Suite batch mode.

batch-setup
This option is used to configure the batch mode options for the Phoronix Test Suite, which is subsequently written to the user configuration file. Among the options are whether to automatically upload the test results to OpenBenchmarking.org and prompting for the saved file name.

default-benchmark *[Test | Suite | OpenBenchmarking.org ID | Test Result] ...*
This option will install the selected test(s) (if needed) and will proceed to run the test(s) in the defaults mode. This option is equivalent to running phoronix-test-suite with the install option followed by the default-run option.

default-run *[Test | Suite | OpenBenchmarking.org ID | Test Result] ...*
This option will run the selected test(s). The name of the test or suite must be supplied or the OpenBenchmarking.org ID or saved local file name. Multiple arguments can be supplied to run additional tests at the same time and save the results in a suite-like fashion. Unlike the normal run option, the default-run will not prompt the user to select from the available test options but will instead use the default options as automatically set by pts-core or the test profile. Use batch-run to automatically test all of the available options.

OpenBenchmarking.org

clone-result *[OpenBenchmarking.org ID] ...*
This option will download a local copy of a file that was saved to OpenBenchmarking.org, as long as a

valid public ID is supplied. More than one ID can be specified and the results will then be merged.

openbenchmarking-launcher

This option is called automatically with the .openbenchmarking MIME file extension support for launching OpenBenchmarking.org operations.

openbenchmarking-login

This option is used for controlling your Phoronix Test Suite client options for OpenBechmarking.org and syncing the client to your account.

openbenchmarking-refresh

This option is used for refreshing the stored OpenBenchmarking.org repostory information and other data. The Phoronix Test Suite will automatically refresh this data every three days or when other thresholds are exceeded, but this command can be used to manually refresh/updates the data.

upload-result *[Test Result]*

This option is used for uploading a test result to OpenBenchmarking.org.

System

diagnostics

This option will print information that is useful to developers when debugging problems with the Phoronix Test Suite and/or test profiles and test suites.

interactive

A simple text-driven interactive interface to the Phoronix Test Suite.

system-info

Display the installed system hardware and software information as detected by the Phoronix Test Suite Phodevi Library.

system-sensors

Display the installed system hardware and software sensors in real-time as detected by the Phoronix Test Suite Phodevi Library.

Information

info *[Test | Suite | OpenBenchmarking.org ID | Test Result]*

This option will show details about the supplied test, suite, virtual suite, or result file.

list-available-suites

This option will list all available test suites that are available from the enabled OpenBenchmarking.org repositories.

list-available-tests

This option will list all available test profiles that are available from the enabled OpenBenchmarking.org repositories.

list-available-virtual-suites

This option will list all available virtual test suites that can be dynamically created based upon the available tests from enabled OpenBenchmarking.org repositories.

list-installed-dependencies

This option will list all of the packages / external test dependencies that are already installed on the system that the Phoronix Test Suite may potentially depend upon by test profiles.

list-installed-suites

This option will list all suites that are currently installed on the system.

list-installed-tests

This option will list all test profiles that are currently installed on the system.

list-missing-dependencies

This option will list all of the packages / external test dependencies that are missing from the system that the Phoronix Test Suite may potentially need by select test profiles.

list-possible-dependencies

This option will list all of the packages / external test dependencies that are are potentially used by the Phoronix Test Suite.

list-saved-results

This option will list all of the saved test results found on the system.

list-test-usage

This option will list various details about installed tests and their usage.

Asset Creation

debug-install *[Test | Suite | OpenBenchmarking.org ID | Test Result] ...*

This option is intended for use by test profile writers and is identical to the install option but will yield more information during the run process that can be used to debug issues with a test profile installer or to verify the test profile is functioning correctly.

debug-run *[Test | Suite | OpenBenchmarking.org ID | Test Result] ...*

This option is intended for use by test profile writers and is identical to the run option but will yield more information during the run process that can be used to debug issues with a test profile or to verify the test profile is functioning correctly.

debug-test-download-links *[Test | Suite]*

This option will check all download links within the specified test profile(s) to ensure there are no broken URLs.

download-test-files *[Test | Suite] ...*

This will download the selected test file(s) to the Phoronix Test Suite download cache but will not install the tests.

force-install *[Test | Suite | OpenBenchmarking.org ID | Test Result] ...*

This option will force the installation (or re-installation) of a test or suite. The arguments and process is

similar to the install option but even if the test is installed, the entire installation process will automatically be executed. This option is generally used when debugging a test installation problem.

result-file-to-suite *[Test Result]*

This option will guide the user through the process of generating their own test suite, which they can then run, that is based upon an existing test results file.

validate-result-file

This option can be used for validating a Phoronix Test Suite result file as being compliant against the OpenBenchmarking.org specification.

validate-test-profile

This option can be used for validating a Phoronix Test Suite test profile as being compliant against the OpenBenchmarking.org specification.

validate-test-suite

This option can be used for validating a Phoronix Test Suite test suite as being compliant against the OpenBenchmarking.org specification.

Result Management

copy-run-in-result-file *[Test Result]*

This option is used if you wish to change an existing test run within a saved results file.

extract-from-result-file *[Test Result]*

This option will extract a single set of test results from a saved results file that contains multiple test results that have been merged. The user is the prompted to specify a new result file name and select which result identifier to extract.

merge-results *[Test Result]* ...

This option will manually merge multiple sets of test results generated by the Phoronix Test Suite.

refresh-graphs *[Test Result]*

This option will re-render and save all result graphs within a saved file. This option can be used when making modifications to the graphing code or its color/option configuration file and testing the changes.

remove-from-result-file *[Test Result]*

This option is used if there is a set of test results you wish to remove/delete from a saved results file. The user must specify a saved results file and then they will be prompted to select the results identifier associated with the results they wish to remove.

remove-result *[Test Result]*

This option will permanently remove the saved file set that is set as the first argument.

rename-identifier-in-result-file *[Test Result]*

This option is used if you wish to change the name of the identifier in a test results file that is shown in the Phoronix Test Suite Results Viewer and the contained graphs.

reorder-result-file *[Test Result]*

This option is used if you wish to change the order in which test results are shown in the Phoronix Test Suite Results Viewer and the contained graphs. The user must specify a saved results file and then they will be prompted to select the results identifiers one at a time in the order they would like them to be displayed from left to right.

result-file-to-csv *[Test Result]*

This option will read a saved test results file and output the system hardware and software information along with the results to a CSV output. The CSV (Comma Separated Values) output can then be loaded into a spreadsheet for easy viewing.

show-result *[Test Result]*

Open up the test results in the Phoronix Test Suite Result Viewer or on OpenBenchmarking.org.

Result Analytics

analyze-all-runs *[Test Result]*

This option will generate a candlestick graph showing the distribution of results from all trial runs. The candlestick graph is similar to the Japanese candlestick charts used by the financial industry, except instead of representing stock data it is numerical result data from all trial runs.

The tip of the upper-wick represents the highest value of the test runs with the tip of the lower-wick representing the lowest value of all test runs. The upper-edge of the candle body represents the first or last run value and the lower-edge represents the first or last run value. Lastly, if the last run value is less than the first run value, the candle body is the same color as the graph background, otherwise the last run value is greater.

analyze-batch *[Test Result]*

This option will analyze a batch results file and plot out the performance impact from the different options onto a line graph (i.e. to see the impact that changing the video resolution has on the system's performance).

analyze-image-delta *[Test Result]*

This option will analyze a test result file if it contains any test results that produced an image quality comparison (IQC) and will render image deltas illustrating the difference between images from two test results.

Other

build-suite

This option will guide the user through the process of generating their own test suite, which they can then run. Optionally, passed as arguments can be the test(s) or suite(s) to add to the suite to be created, instead of being prompted through the process.

help

This option will display the list of available Phoronix Test Suite commands.

network-setup

This option allows the user to configure how the Phoronix Test Suite connects to OpenBenchmarking.org and other web-services. Connecting through an HTTP proxy can be configured

through this option.

rebuild-composite-xml *[Test Result]*
This option can be used to rebuild your composite XML file in the event that the main result file was corrupted and the test-X.xml files are still present in the result file directory.

result-file-to-pdf *[Test Result]*
No summary is available.

result-file-to-text *[Test Result]*
This option will read a saved test results file and output the system hardware and software information to the terminal. The test results are also outputted.

user-config-reset
This option can be used for resetting the Phoronix Test Suite user configuration file to its default state.

user-config-set
This option can be used for setting an XML value in the Phoronix Test Suite user configuration file.

version
This option will display the Phoronix Test Suite client version.

Modules

list-modules
This option will list all of the available Phoronix Test Suite modules on this system.

module-info *[Phoronix Test Suite Module]*
This option will show detailed information on a Phoronix Test Suite module such as the version, developer, and a description of its purpose.

module-setup *[Phoronix Test Suite Module]*
This option will allow you to configure all available end-user options for a Phoronix Test Suite module. These options are then stored within the user's configuration file. Not all modules may have options that can be configured by the end-user.

test-module *[Phoronix Test Suite Module]*
This option can be used for debugging a Phoronix Test Suite module.

Installation Instructions

Overview

The Phoronix Test Suite supports Linux, Apple Mac OS X, Microsoft Windows, Solaris, BSD, and other operating system environments. The only Linux distribution-specific code deals with the external dependencies support feature that are set by individual test profiles. If you are not running one of the supported Linux distributions, Solaris, BSD, or Mac OS X, you may need to install a package manually (as instructed by the Phoronix Test Suite) in order for a test to run. An example of an external dependency would be GCC and the OpenGL Utility Toolkit being needed for test profiles that build an OpenGL benchmark from source-code.

Distributions where the Phoronix Test Suite has been officially tested include Ubuntu, Fedora, Mandriva, Gentoo, PCLinuxOS, Arch Linux, Pardus, OpenSuSE, Optware, webOS, Zenwalk, CentOS, Red Hat Enterprise Linux, Scientific Linux, Debian, Mint, MEPIS, PTS Desktop Live, MeeGo, and Amazon Linux EC2.

Tested BSD distributions include FreeBSD, PC-BSD, NetBSD, OpenBSD, Debian GNU/kFreeBSD, and DragonflyBSD.

Tested Solaris distributions include OpenSolaris, Solaris Express 11, OpenIndiana, and Nexenta.

Dependencies

The only required dependency for the Phoronix Test Suite is PHP 5.2 or newer. On Linux distributions, the needed package is commonly called *php5-cli* or *php-cli* or *php*. It's important to note that only PHP for the command-line is needed and not a web server (Apache) or other packages commonly associated with PHP and its usage by web-sites. The PHP5 version required is PHP 5.2 or newer and can also be found at **www.php.net**.

As part of the PHP requirement, the following PHP extensions are required and/or highly recommended in order to take advantage of the Phoronix Test Suite capabilities:

PHP DOM is needed for XML operations and must be installed for the Phoronix Test Suite to function.

PHP ZIP is needed for file compression and decompression and specifically dealing with test profiles and suites obtained via OpenBenchmarking.org or when uploading such tests and suites.

PHP OpenSSL is used for enabling HTTPS communication with Phoronix Test Suite / OpenBenchmarking.org servers.

PHP GD is highly recommended for rendering of test data to JPEG and PNG image formats and is selectively used in other image operations.

PHP Zlib is highly recommended for greater data compression when dealing with remote OpenBenchmarking.org assets.

PHP PCNTL is used for multi-threaded system sensor monitoring support during the testing process and other threaded tasks by the Phoronix Test Suite module framework.

PHP POSIX is used for reliably obtaining more system information in an efficient manner.

PHP CURL is supported as an alternative networking library for improved network performance in downloading test files and other operations.

PHP FPDF is used to generate PDF reports of test data.

Without all of these extensions, some capabilities of the Phoronix Test Suite will not be available. Many of these packages are enabled by default and do not require any additional installation steps on most Linux distributions, otherwise they are often found in the package vendor's repository.

Notes

General

You may need to modify the *php.ini* file on your system in order to support uploading results to OpenBenchmarking.org or logging into your OpenBenchmarking.org account. The *allow_url_fopen*, *file_uploads*, and *allow_url_include* options must be set to true in the PHP configuration. See additional information in this **Phoronix Forums support thread**.

The Phoronix Test Suite has a feature known as "External Dependencies" where the Phoronix Test Suite can attempt to automatically install some of the test-specific dependencies on supported distributions. If running on a distribution where there is currently no External Dependencies profile, the needed package name(s) are listed. To see all possible External Dependencies, run *phoronix-test-suite list-possible-dependencies*. Running *phoronix-test-suite list-missing-dependencies* will attempt to report the external dependencies that are not installed on the given system.

Major updates to the Phoronix Test Suite are released on a quarterly basis. The latest stable and development versions of the Phoronix Test Suite are available at **Phoronix-Test-Suite.com**. The Git repository where the latest Phoronix Test Suite code is provided is hosted at **Phorogit.com** and can be cloned/pulled from the *http://www.phorogit.com/repo/phoronix-test-suite.git* repository. The latest upstream development code is housed in the master tree while older Phoronix Test Suite releases are available in their respective Git branches based upon the release's code-name.

File Structure

If manually changing the location of the *phoronix-test-suite* launcher file, the *PTS_USER_PATH* environmental variable must be adjusted inside the file to reflect the absolute location that leads to the root directory of the *pts* and *pts-core* folders. The *pts-core* directory contains the "engine" of the Phoronix Test Suite.

Running Locally

The Phoronix Test Suite can be simply extracted from the downloaded *.tar.gz* or *.zip* file or it can also be installed system-wide. If you just wish to run the Phoronix Test Suite without installing it, open a terminal and run *./phoronix-test-suite <options>* from the same directory.

Generic Installation

Running *install-sh* from the root directory of the Phoronix Test Suite will install the software for system-wide access. By default the *phoronix-test-suite* executable is in */usr/bin/*, the Phoronix Test Suite files in */usr/share/phoronix-test-suite/*, and the documentation in */usr/share/doc/phoronix-test-suite/*. Root access is required. The default installation prefix is /usr/ but can be adjusted as the first argument (example: *install-sh /home/user/* to install the Phoronix Test Suite in your home directory).

Debian/Ubuntu Installation

Debian/Ubuntu users are able to follow the Generic Installation instructions or can obtain a Debian Package from the Phoronix Test Suite web-site. The package contains the *phoronix-test-suite* executable in */usr/bin/*, the Phoronix Test Suite files in */usr/share/phoronix-test-suite/*, and the documentation in */usr/share/doc/phoronix-test-suite/*.

A phoronix-test-suite package can also be found in the official Ubuntu universe repository beginning with the Ubuntu 9.04 (Jaunty) release, but it may be out of date compared to the latest upstream version.

Fedora / Red Hat Installation

The Phoronix Test Suite can be installed on Fedora, Red Hat Enterprise Linux, and CentOS systems using the generic installation method. Alternatively, a *phoronix-test-suite* package is available in recent versions of the Fedora repository and in the EPEL (Extra Packages for Enterprise Linux) repository for Red Hat Enterprise Linux. At this time, however, the Fedora-supplied package is vastly out of date.

Gentoo Installation

Gentoo ebuild's for the Phoronix Test Suite have been created by users and can be currently found in the **Gentoo BugZilla**.

Mandriva Installation

Phoronix Test Suite packages are available through Mandriva Cooker starting with the Mandriva Linux 2009 release.

OpenSolaris Installation

The Phoronix Test Suite core (pts-core) is supported on Solaris and OpenSolaris platforms, but some features may not fully function. Additionally, the PTS External Dependencies feature is supported on OpenSolaris using IPS (the Image Packaging System). However, not all test profiles are supported on Solaris / OpenSolaris, and it may be in violation of your end-user agreement to perform such benchmarks on the Sun Solaris operating system. Using the OpenSolaris Image Packaging System, installing *SUNWphp524* will provide PHP, PHP GD, and related dependencies. The Phoronix Test Suite has also been tested on Oracle Solaris 11 Express, OpenIndiana, StormOS, and Nexenta Core Platform.

BSD Installation

The Phoronix Test Suite also supports *BSD operating systems. However, like the Solaris support, not all test profiles are compatible with BSD operating systems, but pts-core has been tested against FreeBSD 7.0/7.2/8.0, NetBSD, PC-BSD 7/8/9, and OpenBSD 4.4/4.6. The Phoronix Test Suite will also work with Debian GNU/kFreeBSD.

Mac OS X Installation

The Phoronix Test Suite is fully supported on Apple's Mac OS X operating system as of Mac OS X 10.5 with improved support under Mac OS X 10.6. PHP ships with Mac OS X by default so it's simply a matter of downloading the Phoronix Test Suite package, extracting it, and running the executable. For tests that rely upon a compiler, Apple's XCode with GCC and LLVM can be utilized.

Windows Installation

The Phoronix Test Suite support on Windows is in development and will function just fine, but is not yet feature complete compared to the other operating systems support. At this time the Phoronix Test Suite client is dependent upon PHP being installed to *C:\Program Files (x86)\PHP\php*. The Phoronix Test Suite also uses **CPU-Z** for much of the hardware detection support on Windows and is dependent upon CPU-Z being installed to *C:\Program Files\CPUID\CPU-Z\cpuz.exe* if you wish to utilize this hardware reporting feature. In terms of the Microsoft Windows support, right now the Phoronix Test Suite is developed and solely tested against Windows 7 x64.

Configuration

User Files & Folders

~/.phoronix-test-suite/user-config.xml

This is a per-user configuration file. Among the information stored here is the test options, locations for storing files, and batch mode options. This file is formatted in XML.

~/.phoronix-test-suite/graph-config.xml

This is a per-user configuration file for storing graph attributes. The adjustable options include HTML hex color codes for different areas of the graph, dimensions of the graph, and font sizes. This file is formatted in XML.

~/.phoronix-test-suite/download-cache/

This directory contains test packages that have been downloaded for test profiles. For more information on the download cache.

~/.phoronix-test-suite/installed-tests/

This directory is where tests are installed by default. Each test has its own directory within a sub-directory of *installed-tests/* based upon its OpenBenchmarking.org repository. In the test's folder is a *pts-install.xml* file used for managing the installation.

~/.phoronix-test-suite/test-results/

This directory is where tests results are saved by default. Each saved file has its own directory. In the saved directory is then a *composite.xml* file containing the useful results while in the *test-X.xml* files are back-ups of the results.

~/.phoronix-test-suite/modules-data/

This is the directory where any Phoronix Test Suite modules should save any files to, within a sub-directory of the module's name. The module configuration settings are also stored within this directory.

~/.phoronix-test-suite/test-profiles/

This is the directory where test profiles are stored.

~/.phoronix-test-suite/test-suites/

This is the directory where test suites are stored.

Environmental Variables

TEST_RESULTS_NAME

When this variable is set, the value will be used as the name for automatically saving the test results.

TEST_RESULTS_IDENTIFIER

When this variable is set, the value will be used as the test identifier when automatically saving the test results.

TEST_RESULTS_DESCRIPTION

When this variable is set, the value will be used as the test results description when saving the test results.

PRESET_OPTIONS

For setting any test option(s) from an environmental variable rather than being prompted for the options when running a test. Example: *PRESET_OPTIONS="stream.run-type=Add" ./phoronix-test-suite benchmark stream*

SKIP_TESTS

If there are any test(s) to exempt from the testing process, specify them in this variable. Multiple tests can be waived by delimiting each test identifier by a comma. A test hardware type (i.e. Graphics) can also be supplied for skipping a range of tests.

PTS_MODULE_SETUP

This variable can be used to load Phoronix Test Suite module settings automatically when using the *module-setup* option. An example would be:
PTS_MODULE_SETUP="phoromatic.remote_host=http://www.phoromatic.com/; phoromatic.remote_account=123456; phoromatic.remote_verifier=ABCD" phoronix-test-suite module-setup phoromatic.

PTS_MODULES

If there are any Phoronix Test Suite modules to additionally load, they can be specified here. Multiple modules can be supplied by delimiting them with a comma. The more appropriate way of loading Phoronix Test Suite modules for longer periods of time is by using the *~/.phoronix-test-suite/user-config.xml* configuration.

NO_PHODEVI_CACHE

This is a debugging option to disable the Phodevi cache from being loaded of cached software/hardware information. Instead, all software/hardware will be polled from the Phodevi library without caching.

EXTERNAL_PHODEVI_CACHE

This option can be used for loading an external Phodevi cache. Such as loading the native hardware/software information from within a Windows Wine client from a native system host.

PTS_DISPLAY_MODE

If you wish to load a non-default display mode for a single instance, specify the mode in this variable.

TOTAL_LOOP_TIME

When running any test(s), if you would like the test(s) to continue running as a loop until a certain time has been reached, this variable can be used. The value should be the number of minutes to run the testing process before the loop is ended.

TOTAL_LOOP_COUNT

When running any test(s), if you would like the test(s) to continue running for a number of times, this variable can be used. The value should be the number of times to loop the testing process before ending.

FORCE_TIMES_TO_RUN

If you wish to override the number of times to run each test -- rather than the Phoronix Test Suite using the number of times specified in each test profile -- this variable can be used.

FORCE_MIN_TIMES_TO_RUN

This is similar to the FORCE_TIMES_TO_RUN option but will only be used if the test profile's run count is less than this defined value.

NO_MD5_CHECKS

To disable MD5 check-sums from being used when downloading test files, set this variable to 1.

PTS_DOWNLOAD_CACHE

While non-standard Phoronix Test Suite download caches can be specified within the *user-config.xml* file, an additional directory to look for potential Phoronix Test Suite download files can be specified by this variable.

GRAPH_HIGHLIGHT

If this variable is set with a valid test identifer from a result file whether you are using the *refresh-graphs* command or any other related to the rendering of test results on a bar graph, the specified test identifier's result will be highlighted in a different color than the other rendered test results. Multiple identifiers can be specified when delimited by a comma.

VIDEO_MEMORY

If Phodevi fails to detect the system's video memory capacity or is incorrectly detected, the video

memory capacity (in MB) can be specified by this variable.

OVERRIDE_VIDEO_MODES

If Phodevi fails to detect all of the system's monitor video modes or a separate set of modes would be preferred, the modes can be specified in this variable. Example:
OVERRIDE_VIDEO_MODES=800x600,1024x768,1280x1024 phoronix-test-suite benchmark nexuiz.

DEFAULT_VIDEO_MODE

If Phodevi fails to detect the system's monitor standard / default resolution, the mode can be specified in this variable. Example: *DEFAULT_VIDEO_MODE=1680x1050 phoronix-test-suite benchmark nexuiz*
.

SKIP_EXTERNAL_DEPENDENCIES_CHECK

To skip the Phoronix Test Suite external dependency checking/installation when installing a test, set this environmental variable to *1*.

General Information

Frequently Asked Questions

Q: May I use the Phoronix Test Suite when running benchmarks for my own publication or blog? Are there any publishing restrictions?

A: Anyone is more than welcome to use the Phoronix Test Suite for their own publication or purpose. While the Phoronix Test Suite came out of our internal test tools for carrying out Linux hardware reviews at **Phoronix.com**, we invite other hardware review web-sites, technology journals, and independent publications to use our software too. While not required, we would just kindly ask that you mention in your review/article that the *Phoronix Test Suite* was used for carrying out your testing, and ideally to link to **www.phoronix-test-suite.com** so that your readers will know where to obtain the software if they are interested in running the tests. You are also more than welcome to upload your results to **OpenBenchmarking.org** so that others may compare their results against yours in an easy manner.

We also try to make the Phoronix Test Suite easy-to-use by independent publications. For example, if you would like to watermark your web-site's URL into the graphs containing your test results, that can be easily modified in *~/.phoronix-test-suite/graph-config.xml*. The colors and other graph settings are also stored in this XML file. If you are a publication and run into any issues with the Phoronix Test Suite or have a feature request, please let us know.

A partial list of other publications using the Phoronix Test Suite can be found in **this forum thread**.

Q: Why does the Phoronix Test Suite not use my distribution's package management system for acquiring all needed packages?

A: The tests themselves are generally downloaded from source and built locally on the machine, rather than fetching any distribution-specific packages. This is done to ensure more comparable results across operating systems / releases, etc. The distribution packager could be applying a number of unknown patches to the software, building the software with unique build options, or making other changes to the software that could skew the results.

Q: Besides being a developer, documentation writer, or having any other unique technical abilities, how else can I contribute to the Phoronix Test Suite?

A: Independent code contributions are very welcome as well as creating your own test profiles and suites. We also appreciate any feedback, comments, or other ideas either by emailing us, posting on the forums, or sending a message to the mailing list. You can also see this (outdated) **forum thread**.

Q: Do you offer technical support for the Phoronix Test Suite

A: Paid, professional support is available and is done via **our commercial services**. We also offer Phoromatic licenses for use within a corporate intranet and other custom services. Free, community support is offered via our **mailing list**, IRC channel (*#phoronix* on *FreeNode.net*, and the **Phoronix Forums**.

Q: May I put the Phoronix Test Suite logo on my company's web-site or on my product packaging?

A: Contact us for licensing information and details regarding the Phoronix Certification & Qualification Suite.

Q: How often is the Phoronix Test Suite updated?

A: We provide major updates on a quarterly basis with an occasional point release to fix outstanding bugs or address other issues. The latest work going into the Phoronix Test Suite is accessible via our Git repository at **Phorogit.com**.

Tips & Tricks

General

- The desktop's screensaver will automatically be shutdown when a test is running and will be restored to its previous state upon the test's completion. This is supported for GNOME, KDE, and other XDG-supportive desktop environments.

- If you have many computers you wish to benchmark, once all of your tests have been downloaded, run *phoronix-test-suite make-download-cache* to generate a copy of these files at *~/.phoronix-test-suite/download-cache/*. A download cache is used for conserving time and bandwidth by eliminating the need for the Phoronix Test Suite to download files that have already been downloaded once. Copy this folder to the other systems or copy it to a DVD or USB hard drive, connect it to the next test system, and the Phoronix Test Suite will automatically use this local download cache. Or store these files on a local HTTP/FTP server and then update your *~/.phoronix-test-suite/user-config.xml* file to reflect the location of this download cache directory.

Running

- When running a test in batch mode (through the use of the *batch-run* or *batch-benchmark* options) that normally has end-user options (such as the sub-test to run or resolution), the Phoronix Test Suite will run the test with each unique combination of options possible, if configured appropriately.

- When running a test where you are prompted to enter any test options, multiple selections can be performed -- which will result in multiple test runs for each combination of the selected option(s) -- by separating each intended test option / number with a comma.

- When being prompted for the test identifier or the file name for saving the results, several user variables are supported. These include *$VIDEO_RESOLUTION*, *$VIDEO_CARD*, *$OPERATING_SYSTEM*, *$PROCESSOR*, *$MOTHERBOARD*, *$CHIPSET*, and *$KERNEL_VERSION*. If any of these variables are entered, the Phoronix Test Suite will replace them with their respective values before saving the results.

- If *RemoveDownloadFiles* is set to *TRUE* within the *user-config.xml* file, once a test has been installed the originally downloaded files for that test will be automatically removed. This conserves disk space but will cause these files to be re-downloaded the next time the test needs to be re-installed. This will also not back up the downloaded files to the Phoronix Test Suite download cache. Enabling this option is just recommended for users with very limited disk space.

- If the amount of video memory for your graphics card is incorrectly reported by the Phoronix Test Suite (you can check by running *phoronix-test-suite diagnostics*), you can use the *VIDEO_MEMORY=* environmental variable for overriding the video memory capacity (in Megabytes) used by the Phoronix Test Suite.

- If the *DISPLAY* environmental variable is not set or *NO_GRAPHICS_TESTS* environmental variable is set, tests of type *Graphics* will not be run. Likewise, if *NO_SYSTEM_TESTS* environmental variable is set, tests of type *System* will not run. This applies to all test types where *NO_<TEST TYPE>_TESTS* is set.

- If while running multiple tests you want to quit the testing prematurely, in a new terminal window type *touch ~/.phoronix-test-suite/halt-testing*. All results for tests that had already run will be saved (permitting you opted to save the results), except for the test currently being run.

- If you wish to stop the current test run prematurely but continue the testing process, in a new terminal window type *touch ~/.phoronix-test-suite/skip-test*.

- If you want the specified test(s) to run in a loop for a set period of time, use the *TOTAL_LOOP_TIME* environmental variable. For instance, running *TOTAL_LOOP_TIME=120 phoronix-test-suite benchmark ffmpeg* would keep running the ffmpeg test profile for 120 minutes.

- If you want the specified test(s) to run in a loop for a set number of times, use the *TOTAL_LOOP_COUNT* environmental variable. For instance, running *TOTAL_LOOP_COUNT=3 phoronix-test-suite benchmark ffmpeg* would keep running the ffmpeg test profile three times.

- When any tests are being installed and when tests are being run, a lock is created in the system's temporary directory with the name *phoronix-test-suite.active* (i.e. */tmp/phoronix-test-suite.active*) and is removed upon completion. Thus if you have any system scripts that you wish to run when tests are not running or being installed as to not impact the results, one simple way to handle this is by having the script check for the existence of this lock.

Configuration
- The user configuration options for the Phoronix Test Suite are stored in *~/.phoronix-test-suite/user-config.xml*. The batch mode options are also stored within this file and those can be adjusted by running *phoronix-test-suite batch-setup*.

- The colors, size, and other attributes for the graphs found within the Phoronix Test Suite Results Viewer can be modified via the file *~/.phoronix-test-suite/graph-config.xml*. No matter the graph type, the same generic *graph-config.xml* configuration is used.

Test / Suite Writing
- The Phoronix Test Suite recursively determines tests/suites and allows a suite to call another suite.

Module Writing
- By writing a Phoronix Test Suite module, you can easily extend the functionality of the Phoronix Test Suite. Some examples are being able to email the results upon completion, controlling the system's screensaver, updating a text LCD panel with the current test status, etc.

Virtual Test Suites

Virtual test suites are not like a traditional test suite defined by the XML suite specification. Virtual test suites are dynamically generated in real-time by the Phoronix Test Suite client based upon the specified test critera. Virtual test suites can automatically consist of all test profiles that are compatible with a particular operating system or test profiles that meet other critera. When running a virtual suite, the OpenBenchmarking.org repository of the test profiles to use for generating the dynamic suite must be prefixed.

Virtual test suites can be installed and run just like a normal XML test suite and shares nearly all of the same capabilities. However, when running a virtual suite, the user will be prompted to input any user-configuration options for needed test profiles just as they would need to do if running the test individually. When running a virtual suite, the user also has the ability to select individual tests within the suite to run or to run all of the contained test profiles. Virtual test suites are also only supported for an OpenBenchmarking.org repository if there is no test profile or test suite of the same name in the repository. Below is a list of common virtual test suites for the main Phoronix Test Suite repository, but the dynamic list of available virtual test suites based upon the enabled repositories is available by running *phoronix-test-suite list-available-virtual-suites.*

All PTS Tests *pts/all*
This is a collection of all test profiles found within the specified OpenBenchmarking.org repository.

Installed Tests *pts/installed*
This is a collection of test profiles found within the specified OpenBenchmarking.org repository that are already installed on the system under test.

Linux Operating System Tests *pts/linux*
This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being compatible with the Linux Operating System.

Solaris Operating System Tests *pts/solaris*
This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being compatible with the Solaris Operating System.

BSD Operating System Tests *pts/bsd*
This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being compatible with the BSD Operating System.

MacOSX Operating System Tests *pts/macosx*
This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being compatible with the MacOSX Operating System.

Windows Operating System Tests *pts/windows*
This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being compatible with the Windows Operating System.

System Subsystem Tests *pts/system*

This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being a test of the System sub-system.

Processor Subsystem Tests *pts/processor*

This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being a test of the Processor sub-system.

Graphics Subsystem Tests *pts/graphics*

This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being a test of the Graphics sub-system.

Other Subsystem Tests *pts/other*

This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being a test of the Other sub-system.

Utility Tests *pts/utility*

This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being a Utility software test.

Simulator Tests *pts/simulator*

This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being a Simulator software test.

Scientific Tests *pts/scientific*

This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being a Scientific software test.

Benchmark Tests *pts/benchmark*

This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being a Benchmark software test.

Application Tests *pts/application*

This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being a Application software test.

Game Tests *pts/game*

This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified as being a Game software test.

Smp Tests *pts/smp*

This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified via an internal tag as testing smp.

Python Tests *pts/python*

This is a collection of test profiles found within the specified OpenBenchmarking.org repository where the test profile is specified via an internal tag as testing python.

Phoromatic

Overview

Phoromatic is a remote management system for the Phoronix Test Suite. Phoromatic allows the automatic (hence the name Phoromatic) scheduling of tests, remote installation of new tests, and the management of multiple test systems all through an intuitive, easy-to-use web interface. Tests can be scheduled to automatically run on a routine basis across multiple test systems. The test results are then available from this central, secure location.

Basic Instructions

1. Create a free account at **Phoromatic.com**. Phoromatic can also be installed on a LAN / intranet for professional and enterprise support customers of the Phoronix Test Suite.

2. On each test system, install the latest version of the Phoronix Test Suite and then run *phoronix-test-suite module-setup phoromatic*. During this process it will ask the user for an account code and activiation ID that are shown from the *Systems* page of your Phoromatic account.

3. Phoromatic cannot install external dependencies for a test if the Phoronix Test Suite is not running as root, so it would be recommended to now run *phoronix-test-suite install-dependencies all*. Alternatively, you can also run *phoronix-test-suite install all* to install all tests at this time, or begin selectively installing the tests you wish to use. Phoromatic can install tests automatically though, it just cannot install external dependencies when the user is not root.

4. When you are all set, run *phoronix-test-suite phoromatic.start*, which begins the process of the Phoronix Test Suite listening to the Phoromatic server waiting for installation and run commands, etc. The installed system software/hardware is also reported all automatically, etc.

For more information on Phoromatic and a detailed guide, read **Building A Benchmarking Test Farm With Phoromatic**.

PTS Desktop Live

PTS Desktop Live 2010.1 "Anzhofen"

PTS Desktop Live and PTS Netbook Live are Linux-based operating systems designed to provide a free software platform for carrying out system tests and hardware benchmarks in an automated, reproducible, and effective manner. The Phoronix Test Suite operating systems are designed around the **Phoronix Test Suite** with the operating system itself consisting of a stripped down Linux environment, then built upon that is the Phoronix Test Suite with its test profiles and suites. This is not a standard Linux desktop with the normal throng of packages.

Testing with PTS Desktop Live is intended to be via a Live DVD or Live USB (i.e. flash drive or memory card) environment. PTS Netbook Live is similar to PTS Desktop Live but designed for testing on lightweight x86 systems and netbooks. Both PTS Desktop Live and PTS Netbook Live make it very easy to deploy the Phoronix Test Suite across an array of systems, isolates any software differences when seeking to perform hardware comparisons, and is designed to be very easy to carry out these Linux benchmarks even for a first-time Linux user.

The inaugural release of this operating system was PTS Desktop Live 2009.3 (codenamed "Gernlinden") and was based upon the Phoronix Test Suite 2.0 release. The Gernlinden release shipped with more than 50 open-source test profiles and all needed files being located on the disc image. PTS Desktop Live uses the GTK2 GUI dialog for the Phoronix Test Suite by default, but the CLI options are also accessible. PTS Desktop Live also features optimizations to the Linux kernel and related components for use on the latest high-end hardware and other performance related tweaking. PTS Netbook Live features optimizations for the Intel Atom processor.

The mission of the Phoronix Test Suite and related products will not be achieved until it has made Linux benchmarking incredibly robust and innovative, drop dead easy to perform nearly any kind of benchmark, and when it out-paces other leading operating systems for its benchmarking abilities. These operating systems are being updated quarterly in conjunction with new Phoronix Test Suite releases. The Phoronix Test Suite will continue to be supported under all other Linux distributions, OpenSolaris, Mac OS X, Windows, and *BSD operating systems, but this is just another addition on top of that.

PTS Desktop Live is designed to run on x86_64 compatible systems with 2GB or more of system memory (ideally 4GB+ if you wish to run many of the tests in a live environment) being highly recommended along with a modern CPU and graphics processor. No proprietary graphics drivers are shipped with PTS Desktop Live at this time, so depending upon the hardware you may not have OpenGL acceleration to run the few graphics-related tests that are included with this release. Only graphics drivers are provided for ATI/AMD, NVIDIA, and Intel hardware. An Internet connection is not required but is needed for communication with OpenBenchmarking.org and for utilizing some of the other Phoronix Test Suite features.

PTS Netbook Live is designed for netbooks and nettop computers with an Intel Atom processor.

Quick Start Guide
When the operating system has fully booted, the end-user agreement for the Phoronix Test Suite will be

displayed, followed by initializing the main GTK user interface for the Phoronix Test Suite. The Phoronix Test Suite has more than 100 test profiles while there are about 50 test profiles featured within live environment, which are free software, do not pose significant install size problems, and are suitable for benchmarking from a live environment. By default, PTS Desktop Live and PTS Netbook Live will only show the test profiles and suites where all of the needed test files and dependencies ship with this open-source operating system. However, to access any of the test profiles that may require downloading additional files or external dependencies, use the *View* menu to toggle the different options. The system's hardware and software information can also be viewed in a graphical window from this menu.

From the *Available Tests* tab, any test/suite can be selected to view more information. If you wish to install the test, simply click on the install button while waiting shortly for it to install. Afterwards, the test can be run by selecting it again and clicking on the run/benchmark button. If you have a **OpenBenchmarking.org** account, you can login from the *File* menu while Phoronix Test Suite settings are adjustable from the *Edit* menu.

When the testing has completed, the graphical user interface will reappear. Via the buttons that appear at the top of the interface, the test results can then be opened within the web browser. Depending upon the test profile, it may also be possible to immediately compare the test result(s) against a set of reference systems. If the completed test(s) do have reference results available, a list of the comparable systems are shown, so that you can click on one or more of these reference systems to see how the given system performs in comparison. Results from OpenBenchmarking.org can also be compared with any of the test results.

Assuming you did not launch any commands from the terminal to alter the disk(s) or its contents or carry out any other changes; PTS Desktop Live will have not touched your data or the installed operating system(s). If you wish to archive any of your test results, upload the results to OpenBenchmarking.org. Alternatively, when going to the *Test Results* tab and selecting a test, the *File* menu presents options for exporting results to text, CSV, and PDF files. The contents of the *~/.phoronix-test-suite/test-results/* folder can also be backed up for preserving the test results.

If you wish to use the Phoronix Test Suite from the command-line, open a terminal and run *phoronix-test-suite*.

Development Credits

The Phoronix Test Suite is based upon the extensive testing and internal tools developed by **Phoronix.com** since 2004 along with support from leading tier-one computer hardware and software vendors. The principal architects of the Phoronix Test Suite are **Michael Larabel** and Matthew Tippett. The phoronix-test-suite, pts_Graph, bilde_renderer, Phodevi, tandem_Xml, and nye_Xml are some of the related open-source projects provided by **Phoronix Media**.

Below is a list of individuals and organizations that have contributed upstream patches, test profiles, and/or made other contributions to the Phoronix Test Suite that Phoronix Media would like to thank.

Achim Gottinger

Achim Lange

Andrew Schofield

Apaige

Ben Kero

Chris Campbell

Daniel DÃaz

Daniel Stodden / Citrix

Dean Hilkewich

Fujitsu

Hans Ulrich Niedermann

Jeroen van Rijn

Johannes Obermayr

Juan Aguado

JÃ¶rg Schirottke

Kenney Phillis

Kim Bisgaard

Laminar Research

Matthew Tippett

Michael Larabel

Michael Opdenacker

Michael Ploujnikov

Nils Kneuper

Pekka Panula

Peter Kraus

Rainer König

SOMA Networks, Inc.

Stefan Doesinger

Stepan Hrbek

Stéphane Teletchéa

Sun Microsystems

Tao Zhang

Thomas Klausner

Unigine Corp

Vadim Peretokin

Zachary Powers