NAME

kyua **-** Testing framework for infrastructure software

SYNOPSIS

kyua [**--config** *file*] [**--logfile** *file*] [**--loglevel** *level*] [**--variable** *name=value*] *command* [*command_options*] [*command_arguments*]

DESCRIPTION

If you are here looking for details on how to run the test suite in /usr/tests (*or /usr/tests*)*, please start by reading the* tests(7) *manual page that should be supplied by your system*.

Kyua is a testing framework for infrastructure software, originally designed to equip BSD-based operating systems with a test suite. This means that Kyua is lightweight and simple, and that Kyua integrates well with various build systems and continuous integration frameworks.

Kyua features an expressive test suite definition language, a safe runtime engine for test suites and a powerful report generation engine.

Kyua is for both developers and users, from the developer applying a simple fix to a library to the system administrator deploying a new release on a production machine.

Kyua is able to execute test programs written with a plethora of testing libraries and languages. The test program library of choice is ATF, which **kyua**'s design originated from. However, framework-less test programs and TAP-compliant test programs can also be executed through **kyua**

Overview

As can be observed in the synopsis, the interface of **kyua** implements a common subcommand-based interface. The arguments to the tool specify, in this order: a set of common options that all the commands accept, a required *command* name that specifies what **kyua** should do, and a set of possibly-optional *command_options* and *command_arguments* that are specific to the chosen command.

The following options are recognized by all the commands. Keep in mind that these must always be specified before the command name.

--config *path*, **-c** *path*

Specifies the configuration file to process, which must be in the format described in kyua.conf(5). The special value ‘none’ explicitly disables the loading of any configuration file.

Defaults to *~/.kyua/kyua.conf* if it exists, otherwise to */etc/kyua/kyua.conf* if it exists, or else to ‘none’.

--logfile *path*

Specifies the location of the file to which **kyua** will log run time events useful for postmortem debugging.

The default depends on different environment variables as described in *Logging*, but typically the file will be stored within the user's home directory.

--loglevel *level*

Specifies the maximum logging level to record in the log file. See *Logging* for more details.

The default is 'info'.

--variable *name=value*, **-v** *name=value*

Sets the *name* configuration variable to *value*. The values set through this option have preference over the values set in the configuration file.

The specified variable can either be a builtin variable or a test-suite specific variable. See kyua.conf(5) for more details.

The following commands are generic and do not have any relation to the execution of tests or the inspection of their results:

about Shows general program information. See kyua-about(1).

config Inspects the values of the configuration variables. See kyua-config(1).

db-exec Executes an arbitrary SQL statement on a results file and prints the resulting table. See kyua-db-exec(1).

help Shows usage information. See kyua-help(1).

The following commands are used to generate reports based on the data previously recorded in a results file:

report Generates a plaintext report. Combined with its **--verbose** flag and the ability to only display specific test cases, this command can also be used to debug test failures post-facto on the console. See kyua-report(1).

report-html Generates an HTML report. See kyua-report-html(1).

report-junit Generates a JUnit report. See kyua-report-junit(1).

The following commands are used to interact with a test suite:

debug Executes a single test case in a controlled environment for debugging purposes. See kyua-debug(1).

list Lists test cases defined in a test suite by a kyuafile(5) and, optionally, displays their metadata. See kyua-list(1).

test Runs tests defined in a test suite by a kyuafile(5). See kyua-test(1).

Logging

kyua has a logging facility that collects all kinds of events at run time. These events are always logged to a file so that the log is available when it is most needed: right after a non-reproducible problem happens. The only way to disable logging is by sending the log to */dev/null*.

The location of the log file can be manually specified with the **--logfile** option, which applies to all commands. If no file is explicitly specified, the location of the log files is chosen in this order:

1. *${HOME}/.kyua/logs/* if *HOME* is defined.

2. *${TMPDIR}/* if *TMPDIR* is defined.

3. */tmp/*.

And the default naming scheme of the log files is: ‘<progname>.<timestamp>.log’.

The messages stored in the log file have a level (or severity) attached to them. These are:

error Fatal error messages. The program generally terminates after these, either in a clean manner or by crashing.

warning Non-fatal error messages. These generally report a condition that must be addressed but the application can continue to run.

info Informational messages. These tell the user what the program was doing at a general level of operation.

debug Detailed informational messages. These are often useful when debugging problems in the application, as they contain lots of internal details.

The default log level is ‘info’ unless explicitly overridden with **--loglevel**.

The log file is a plain text file containing one line per log record. The format of each line is as follows:

```
timestamp entry_type pid file:line: message
```

entry_type can be one of: ‘E’ for an error, ‘W’ for a warning, ‘I’ for an informational message and ‘D’ for a debug message.

Bug reporting

If you think you have encountered a bug in **kyua**, please take the time to let the developers know about it. This will ensure that the bug is addressed and potentially fixed in the next Kyua release.

The first step in reporting a bug is to check if there already is a similar bug in the database. You can check what issues are currently in the database by going to:

https://github.com/jmmv/kyua/issues/

If there is no existing issue that describes an issue similar to the one you are experiencing, you can open a new one by visiting:

https://github.com/jmmv/kyua/issues/new/

When doing so, please include as much detail as possible. Among other things, explain what operating system and platform you are running **kyua** on, what were you trying to do, what exact messages you saw on the screen, how did you expect the program to behave, and any other details that you may find relevant.

Also, please include a copy of the log file corresponding to the problem you are experiencing. Unless you have changed the location of the log files, you can most likely find them in *~/.kyua/logs/*. If the problem is reproducible, it is good idea to regenerate the log file with an increased log level so as to provide more information. For example:

```
$ kyua --logfile=problem.log --loglevel=debug \
    [rest of the command line]
```

ENVIRONMENT

The following variables are recognized and can be freely tuned by the end user:

COLUMNS The width of the screen, in number of characters. **kyua** uses this to wrap long lines. If not present, the width of the screen is determined from the terminal stdout is connected to, and, if the guessing fails, this defaults to infinity.

HOME Path to the user's home directory. **kyua** uses this location to determine paths to configuration files and default log files.

TMPDIR Path to the system-wide temporary directory. **kyua** uses this location to place the work directory of test cases, among other things.

The default value of this variable depends on the operating system. In general, it is */tmp*.

The following variables are also recognized, but you should not need to set them during normal operation. They are only provided to override the value of built-in values, which is useful when testing **kyua** itself:

KYUA_CONFDIR Path to the system-wide configuration files for **kyua**.

Defaults to */etc/kyua*.

KYUA_DOCDIR Path to the location of installed documentation.

Defaults to */usr/share/doc/kyua*.

KYUA_MISCDIR Path to the location of the installed miscellaneous scripts and data files provided by **kyua**.

Defaults to */usr/share/kyua/misc*.

KYUA_STOREDIR Path to the location of the installed store support files; e.g., the directory containing the SQL database schema.

Defaults to */usr/share/kyua/store*.

FILES

~/.kyua/store/

Default location for the results files.

~/.kyua/kyua.conf

User-specific configuration file.

~/.kyua/logs/

Default location for the collected log files.

/etc/kyua/kyua.conf

System-wide configuration file.

EXIT STATUS

kyua returns 0 on success, 1 on a controlled error condition in the given subcommand, 2 on a general unexpected error and 3 on a usage error.

The documentation of the subcommands in the corresponding manual pages only details the difference between a successful exit (0) and the detection of a controlled error (1). Even though when those manual pages do not describe any other exit statuses, codes above 1 can be returned.

SEE ALSO

kyua.conf(5), kyuafile(5), atf(7), tests(7)

AUTHORS

For more details on the people that made **kyua** possible and the license terms, run:

```
$ kyua about
```