

**NAME**

**kyua report** - Generates reports with the results of a test suite run

**SYNOPSIS**

**kyua report** [--output *path*] [--results-file *file*] [--results-filter *types*] [--verbose]  
[*test\_filter1* .. *test\_filterN*]

**DESCRIPTION**

The **kyua report** command parses a results file and generates a user-friendly, plaintext report for user consumption on the terminal. By default, these reports only display a summary of the execution of the full test suite to highlight where problems may lie.

The output of **kyua report** can be customized to display full details on all executed test cases. Additionally, the optional arguments to **kyua report** are used to select which test programs or test cases to display. These are filters and are described below in *Test filters*.

Reports generated by **kyua report** are *not intended to be machine-parseable*.

The following subcommand options are recognized:

**--output *path***

Specifies the path to which the report should be written to. The special values */dev/stdout* and */dev/stderr* can be used to specify the standard output and the standard error, respectively.

**--results-file *path*, -s *path***

Specifies the results file to operate on. Defaults to 'LATEST', which causes **kyua report** to automatically load the latest results file from the current test suite.

The following values are accepted:

**'LATEST'**

Requests the load of the latest results file available for the test suite rooted at the current directory.

**Directory**

Requests the load of the latest results file available for the test suite rooted at the given directory.

**Test suite name**

Requests the load of the latest results file available for the given test suite.

Results identifier

Requests the load of a specific results file.

Explicit file name (aka everything else)

Load the specified results file.

See *Results files* for more details.

### **--results-filter types**

Comma-separated list of the test result types to include in the report. The ordering of the values is respected so that you can determine how you want the list of tests to be shown.

The valid values are: 'broken', 'failed', 'passed', 'skipped' and 'xfail'. If the parameter supplied to the option is empty, filtering is suppressed and all result types are shown in the report.

The default value for this flag includes all the test results except the passed tests. Showing the passed tests by default clutters the report with too much information, so only abnormal conditions are included.

### **--verbose**

Prints a detailed report of the execution. In addition to all the information printed by default, verbose reports include the runtime context of the test suite run, the metadata of each test case, and the verbatim output of the test cases.

## **Results files**

Results files contain, as their name implies, the results of the execution of a test suite. Each test suite executed by `kyua-test(1)` generates a new results file, and such results files can be loaded later on by inspection commands such as `kyua-report(1)` to analyze their contents.

Results files support identifier-based lookups and also path name lookups. The differences between the two are described below.

The default naming scheme for the results files provides simple support for identifier-based lookups and historical recording of test suite runs. Each results file is given an identifier derived from the test suite that generated it and the time the test suite was run. Kyua can later look up results files by these fields.

The identifier follows this pattern:

`<test_suite>.<YYYYMMDD>--<HHMMSS>--<uuuuuu>`

where ‘test\_suite’ is the path to the root of the test suite that was run with all slashes replaced by underscores and ‘YYYYMMDD-HHMMSS-uuuuu’ is a timestamp with microsecond resolution.

When using the default naming scheme, results files are stored in the `~/.kyua/store/` subdirectory and each file holds a name of the form:

```
~/.kyua/store/results.<identifier>.db
```

Results files are simple SQLite databases with the schema described in the `/usr/share/kyua/store/schema_v?.sql` files. For details on the schema, please refer to the heavily commented SQL file.

### Test filters

A *test filter* is a string that is used to match test cases or test programs in a test suite. Filters have the following form:

```
test_program_name[:test_case_name]
```

Where ‘test\_program\_name’ is the name of a test program or a subdirectory in the test suite, and ‘test\_case\_name’ is the name of a test case.

### EXIT STATUS

The **kyua report** command returns 0 if no filters were specified or if all filters match one or more test cases. If any filter fails to match any test case, the command returns 1.

Additional exit codes may be returned as described in `kyua(1)`.

### EXAMPLES

#### Workflow with results files

If one runs the following command twice in a row:

```
kyua test -k /usr/tests/Kyuafile
```

the two executions will generate two different files with names like:

```
~/.kyua/store/results.usr_tests.20140731-150500-196784.db  
~/.kyua/store/results.usr_tests.20140731-151730-997451.db
```

Taking advantage of the default naming scheme, the following commands would all generate a report for the results of the *latest* execution of the test suite:

```
cd /usr/tests && kyua report
cd /usr/tests && kyua report --results-file=LATEST
kyua report --results-file=/usr/tests
kyua report --results-file=usr_tests
kyua report --results-file=usr_tests.20140731-151730-997451
```

But it is also possible to explicitly load data for older runs or from explicitly-named files:

```
kyua report \
--results-file=usr_tests.20140731-150500-196784
kyua report \
--results-file=~/.kyua/store/results.usr_tests.20140731-150500-196784.db
```

## SEE ALSO

kyua(1), kyua-report-html(1), kyua-report-junit(1)