

NAME

kyua report-junit - Generates a JUnit report with the results of a test suite run

SYNOPSIS

kyua report-junit [--output *path*] [--results-file *file*]

DESCRIPTION

The **kyua report-junit** command provides a simple mechanism to generate JUnit reports of the execution of a test suite. The command processes a results file and then generates a single XML file that complies with the JUnit XSchema.

The JUnit output is static and self-contained, so it can easily be plugged into any continuous integration system, like Jenkins.

The following subcommand options are recognized:

--output *directory*

Specifies the file into which to store the JUnit report.

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

Specifies the results file to operate on. Defaults to 'LATEST', which causes **kyua report-junit** 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.

Caveats

Because of limitations in the JUnit XML schema, not all the data collected by Kyua can be properly represented in JUnit reports. However, because test data are extremely useful for debugging purposes, the **kyua report-junit** command shovels these data into the JUnit output. In particular:

- The test case metadata values are prepended to the test case's standard error output.
- Test cases that report expected failures as their results are recorded as passed. The fact that they failed as expected is recorded in the test case's standard error output along with the corresponding reason.

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-uuuuuu' 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.

EXIT STATUS

The **kyua report-junit** command always returns 0.

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-junit  
cd /usr/tests && kyua report-junit --results-file=LATEST  
kyua report-junit --results-file=/usr/tests  
kyua report-junit --results-file=usr_tests  
kyua report-junit --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-junit \  
  --results-file=usr_tests.20140731-150500-196784  
kyua report-junit \  
  --results-file=~/kyua/store/results.usr_tests.20140731-150500-196784.db
```

SEE ALSO

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