XTDMake Documentation

Release 1.0.3

Xavier MARCELET

Dec 19, 2018

Contents:

CHAPTER 1

Introduction

XTDMake is a set of CMake packages that provides easy-to-use targets that generate code quality measurements reports.

- Documentation (using Doxygen)
- Documentation coverage (using Doxygen and Lcov)
- Count lines of code (using Cloc)
- C++ static code analysis (using CppCheck)
- Unit tests (using CMake's test facility)
- Code coverage (using Lcov)
- Memory leak of unit tests (using Valgrind)
- Code duplication analysis (using Pmd)
- C++ include sanitizing (using Iwyu)

Each target generates both a locally readable and machine processable reports. Local report targets the developer while the machine-processable reports can be used in your Continuous Integration (CI) process.

1.1 Locally runnable

Key Point Indicators (KPIs) measurement tools are often built in the CI work flow and therefore cannot be run on the developer's local environment. This usually lead to discovering regressions (failed tests, a lower coverage or what-so-ever) only after pushing code to distant repository. Developer's being responsible for the KPIs, they should be able to run the measurement tools before pushing new code.

1.2 Per module

Because code of industrial applications is usually divided in different modules, each with a different purpose and levels of criticity, XTDMake's KPIs reports are generated per module, allowing a finer interpretation of the indicators.

1.3 Incremental execution

C++ compilation is already slow enough. XTDMake's targets are designed to be fully incremental with a fine dependency tracking.

CHAPTER 2

Installation

2.1 From PPA Package

Project homepage : https://launchpad.net/~psycofdj/+archive/ubuntu/xtdmake

1. Add PPA repository to apt

```
sudo add-apt-repository ppa:psycofdj/xtdmake
```

2. Update apt

```
sudo apt-get update
```

3. Install XTDMake

```
sudo apt-get install --install-suggests xtdmake
```

2.2 From source

Project homepage : https://github.com/psycofdj/xtdmake

Note: Each packages requires a set of programs. You're not forced to install everything if you don't need all XTDMake's modules.

1. Install suggested dependencies

```
# Doxygen (Generate documentation from source code)
sudo apt-get install doxygen
# Dot (Generate pictures from graphs)
sudo apt-get install graphviz
```

(continues on next page)

(continued from previous page)

```
# xsltproc (Transform XML files from XSLT style-sheets)
sudo apt-get install xsltproc
# lcov (Generate HTML results from code-coverage information)
sudo apt-get install lcov
# coverxygen (Generate documentation-coverage information from doxygen␣
↪documentation)
sudo pip install coverxygen
# cloc (Count line of codes)
sudo apt-get install cloc
# cppcheck (C++ static code analysis tool)
sudo apt-get install cppcheck
# valgrind instrumentation framework for dynamic analysis
sudo apt-get install valgrind
# jq, awk for json
sudo apt-get install jq
# java 8
sudo apt-get install openjdk-8-jre
# PMD
wget https://github.com/pmd/pmd/releases/download/pmd_releases%2F5.7.0/pmd-
↪bin-5.7.0.zip
sudo unzip -d /usr/share pmd-bin-5.7.0.zip
# Include what you use
sudo apt-get install iwyu
```

2. Download latest release

```
# fetch latest release version
tag=$(curl -s https://api.github.com/repos/psycofdj/xtdmake/tags | \
  jq -r '[ .[] | .["name"] ] | sort | last')

# download archive
wget https://github.com/psycofdj/xtdmake/archive/${tag}.tar.gz -O xtdmake-$
↪{tag}.tar.gz

# uncompress archive
tar xvzf xtdmake-${tag}.tar.gz
```

3. Install XTDMake

```
cd xtdmake-${tag}.tar.gz
mkdir .build
cd .build
cmake ..
sudo make install
```

CHAPTER 3

Quick Start

In your root CMakeLists.txt

```
# ----------
# cmake init
# ----------

cmake_minimum_required(VERSION 2.6)
project(<project_name>)

# enabled_testing() must be called at top-level for module CheckRule to work
# properly
enable_testing()

# project's versions must be set for module StaicShared to work properly
set(PROJECT_VERSION_MAJOR 0)
set(PROJECT_VERSION_MINOR 1)
set(PROJECT_VERSION_PATCH 1)

# ------------
# load XTDmake
# ------------

# All XTDMake global default parameters must be set before calling init function.
# Ex:
#   -> list(APPEND CheckRule_DEFAULT_LINKS "${Boost_LIBRARIES}")

# this function load desisred XTDMake module, each one may or may not be REQUIRED
xtdmake_init(
  StaticShared    REQUIRED
  DocRule         REQUIRED
  DocCoverageRule REQUIRED
  CppcheckRule    REQUIRED
  CheckRule       REQUIRED
  ClocRule        REQUIRED
```

(continues on next page)

(continued from previous page)

```
  Tracking        REQUIRED
  Cppunit         REQUIRED
  CovRule         REQUIRED
  MemcheckRule    REQUIRED
  CodeDupRule     REQUIRED
  Reports         REQUIRED)

# make XTDMake aware of current cmake project
xtdmake_init_project(<project_name> ${PROJECT_BINARY_DIR})

# (optional) configure XTDMake to injects dependency tracking informations in␣
↪binaries and libraries
enable_tracking()

# ---------------------------
# rest of your CMakeLists.txt
# ---------------------------
```

3.1 Load code quality targets

In your module CMakelists.txt, example core/CMakeLists.txt :

```
include_directories(
  ${Boost_INCLUDE_DIRS}
  ${core_INCLUDE_DIRS}
  )

# Create both statis and shared libraries using a single call
add_shared_static_library(core
  src/types.cc
  src/log/Appender.cc
  src/log/ColoredFormatter.cc
  src/log/ConfLoader.cc
  src/log/Formatter.cc
  src/log/helpers.cc
  src/log/Logger.cc
  src/log/MemoryAppender.cc
  src/log/StreamAppender.cc
  src/log/Stream.cc
  src/log/SyslogAppender.cc
  src/log/FormatModifiers.cc
  src/tty.cc
  src/text.cc
  src/Application.cc
  src/config/Parser.cc
  src/config/Grammar.cc
  )

# enable doxygen documentation
add_doc(core)

# enable documentation coverage report
add_doc_coverage(core)
```

(continues on next page)

(continued from previous page)

```
# enable count lines of code report
add_cloc(core)

# enable cppcheck report
add_cppcheck(core)

# enable unittests report
# link all test to static version on library libcore
add_check(core
  INCLUDES ./src}
  LINKS    core_s)

# enable test coverage report
add_cov(core)

# enable test memory check report
add_memcheck(core)

# enable code duplication report
add_codedup(core)

# enable code duplication report
add_iwyu(core)
```

3.2 Adds some unittests

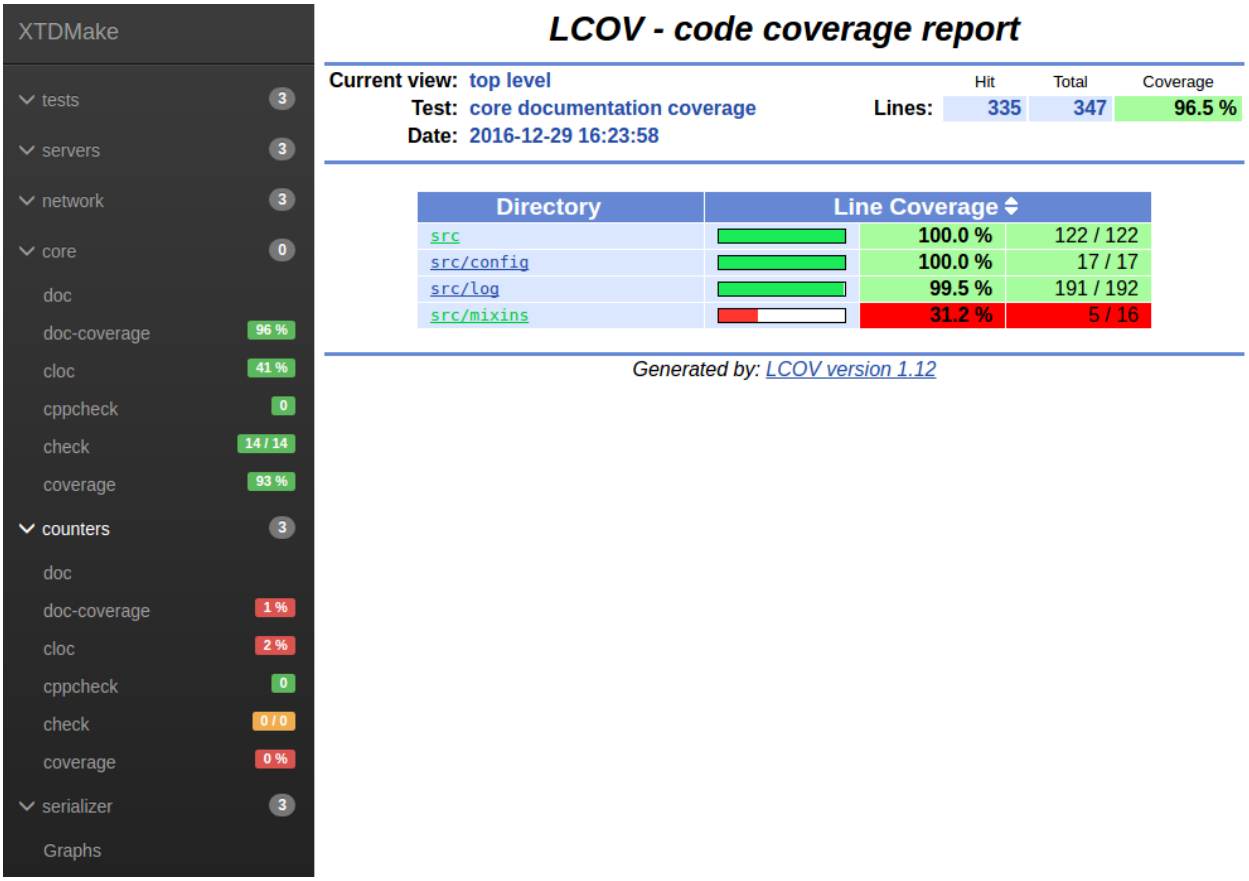
in core/unit/TestMyClass.cc

3.3 Run targets

```
$ cd path-to-build-dir
$ make reports
...
...
...
[100%] Built target

$ make reports-show
(browser opens on report interface)
```

3.4 Binaries RSC keywords



CHAPTER 4

Code quality modules

4.1 DocRule

This module generates a report from result of cppcheck static analysis.

4.1.1 Prerequisites

doxygen Code documentation generator for C/C++. Available from ubuntu packages or from source at http://www.doxygen.org/

graphviz Graph drawing tools, Available from ubuntu packages or from source at http://www.graphviz.org/

Plantuml UML diagrams drawing tool. Available from ubuntu packages (>= xenial) or from source at http://plantuml.com/

4.1.2 Functions

```
add_doc(module,
  [INPUT               <dir>     [ <dir>     ... ]]
  [FILE_PATTERNS       <pattern> [ <pattern> ... ]]
  [EXCLUDE             <file>    [ <file>    ... ]]
  [EXCLUDE_PATTERNS    <pattern> [ <pattern> ... ]]
  [PREDEFINED          <name>    [ <name>    ... ]]
  [EXPAND_AS_DEFINED   <name>    [ <name>    ... ]]
  [EXAMPLE             <dir>  ]
  [PLANTUML]           <jar>  ]
  [IMAGE]              <dir>  ]
  [CONFIGURE_TEMPLATE <file> ]
  [WERROR              { YES | NO } ]
  [CALL_GRAPHS         { YES | NO } ]
)
```

This function generates cmake targets that produce doxygen documentation for a given module. Generated targets are added as dependency of the global `doc` and `doc-clean` targets.

4.1.3 Parameters

module Name of the module. It determines the name of the generated cmake targets and the directory where targets generate the report.

INPUT List of directories where target should search source files to process. Ultimatly this paramter will be given to doxygen `INPUT` configuration (see https://www.stack.nl/~dimitri/doxygen/manual/config.html#cfg_input).

Default value is given by *DocRule_DEFAULT_INPUT*

FILE_PATTERNS List of wildcards search files in given input directories. Ultimatly this paramter will be given to doxygen `FILE_PATTERNS` configuration. (see https://www.stack.nl/~dimitri/doxygen/manual/config.html#cfg_input). Together with `INPUT`, this paramter will determine the files dependency of generated target.

Default value is given by *DocRule_DEFAULT_FILE_PATTERNS*

EXCLUDE List of files to exclude from doxygen generation. Ultimatly this paramter will be given to doxygen `EXCLUDE` configuration. (see https://www.stack.nl/~dimitri/doxygen/manual/config.html#cfg_exclude).

Default value is given by *DocRule_DEFAULT_EXCLUDE*

EXCLUDE List of patterns to exclude from doxygen generation. Ultimatly this paramter will be given to doxygen `EXCLUDE_PATTERNS` configuration. (see https://www.stack.nl/~dimitri/doxygen/manual/config.html#cfg_exclude_patterns).

Default value is given by *DocRule_DEFAULT_EXCLUDE_PATTERNS*

EXCLUDE List of predefined macro given to doxygen in `PREDEFINED` configuration (see https://www.stack.nl/~dimitri/doxygen/manual/config.html#cfg_predefined).

Default value is given by *DocRule_DEFAULT_PREDEFINED*

EXPAND_AS_DEFINED List of predefined macro given to doxygen in `EXPAND_AS_DEFINED` configuration (see https://www.stack.nl/~dimitri/doxygen/manual/config.html#cfg_expand_as_defined).

Default value is given by *DocRule_DEFAULT_EXPAND_AS_DEFINED*

EXAMPLE Directory containing examples files given to doxygen as `EXAMPLE_PATH` configuration (see https://www.stack.nl/~dimitri/doxygen/manual/config.html#cfg_example_path).

Default value is given by *DocRule_DEFAULT_EXAMPLE*

IMAGE Directory containing images files given to doxygen as `IMAGE_PATH` configuration (see https://www.stack.nl/~dimitri/doxygen/manual/config.html#cfg_image_path).

Default value is given by *DocRule_DEFAULT_IMAGE*

PLANTUML Path to plantuml jar file given to doxygen as `PLANTUML_JAR_PATH` configuration (see https://www.stack.nl/~dimitri/doxygen/manual/config.html#cfg_plantuml_jar_path).

Default value is given by *DocRule_DEFAULT_PLANTUML*

> **Warning:** Plantml integration is not supported in doxygen version prior to 1.8.11. In that case this parameter has no effect.

WERROR If YES, doxygen warning are threated as errors (see https://www.stack.nl/~dimitri/doxygen/manual/config.html#cfg_warn_as_error).

Default value is given by *DocRule_DEFAULT_WERROR*

CALL_GRAPHS If YES, doxygen will generate call graph and caller graph. This option requires grahviz to be installed.

Default value is given by *DocRule_DEFAULT_CALL_GRAPHS*

CONFIGURE_TEMPLATE Path to doxygen configuration template to use. If empty, the function chooses one if its own default templates based on currently installed doxygen version.

Default value is given by *DocRule_DEFAULT_CONFIG*

> **Warning:** For XTDMake to work correclty with your manually defined configure template, you must insure that :
>
> - `GENERATE_XML` is `YES` (required by *DocCoverageRule* module)
> - `OUTPUT_DIRECTORY` is `@DocRule_OUTPUT@`.

Tip: The following variables are given to the configure template :

- `@CMAKE_PROJECT_NAME@`
- `@DocRule_MODULE@`
- `@DocRule_OUTPUT@`
- `@DocRule_WERROR@`
- `@DocRule_INPUT@`
- `@DocRule_FILE_PATTERNS@`
- `@DocRule_EXCLUDE@`
- `@DocRule_EXAMPLE@`
- `@DocRule_IMAGE@`
- `@DocRule_PREDEFINED@`
- `@DocRule_EXPAND_AS_DEFINED@`
- `@DocRule_CALL_GRAPHS@`
- `@DocRule_PLANTUML@`

4.1.4 Global variables

DocRule_DEFAULT_EXCLUDE
""

DocRule_DEFAULT_EXCLUDE_PATTERNS
""

DocRule_DEFAULT_FILE_PATTERNS
"*.cc;*.hh;*.hpp"

DocRule_DEFAULT_PREDEFINED
""

DocRule_DEFAULT_EXPAND_AS_DEFINED
""

DocRule_DEFAULT_EXAMPLE
"${CMAKE_CURRENT_SOURCE_DIR}/doc/example"

DocRule_DEFAULT_IMAGE
"${CMAKE_CURRENT_SOURCE_DIR}/doc/image"

DocRule_DEFAULT_PLANTUML
"/usr/share/plantuml/plantuml.jar"

DocRule_DEFAULT_INPUT
"${CMAKE_CURRENT_SOURCE_DIR}/src;${CMAKE_CURRENT_SOURCE_DIR}/doc"

DocRule_DEFAULT_WERROR
"YES"

DocRule_DEFAULT_CALL_GRAPHS
"YES"

DocRule_DEFAULT_CONFIG
""

4.1.5 Generated targets

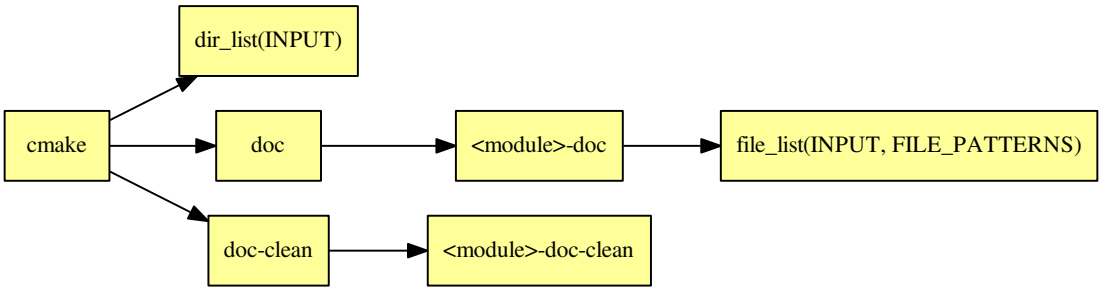
doc generate doc reports for all modules

doc-clean removes doc reports for all modules

<module>-doc generate doc report for module <*module*>

<module>-doc-clean removes doc report for module <*module*>

4.1.6 Dependencies



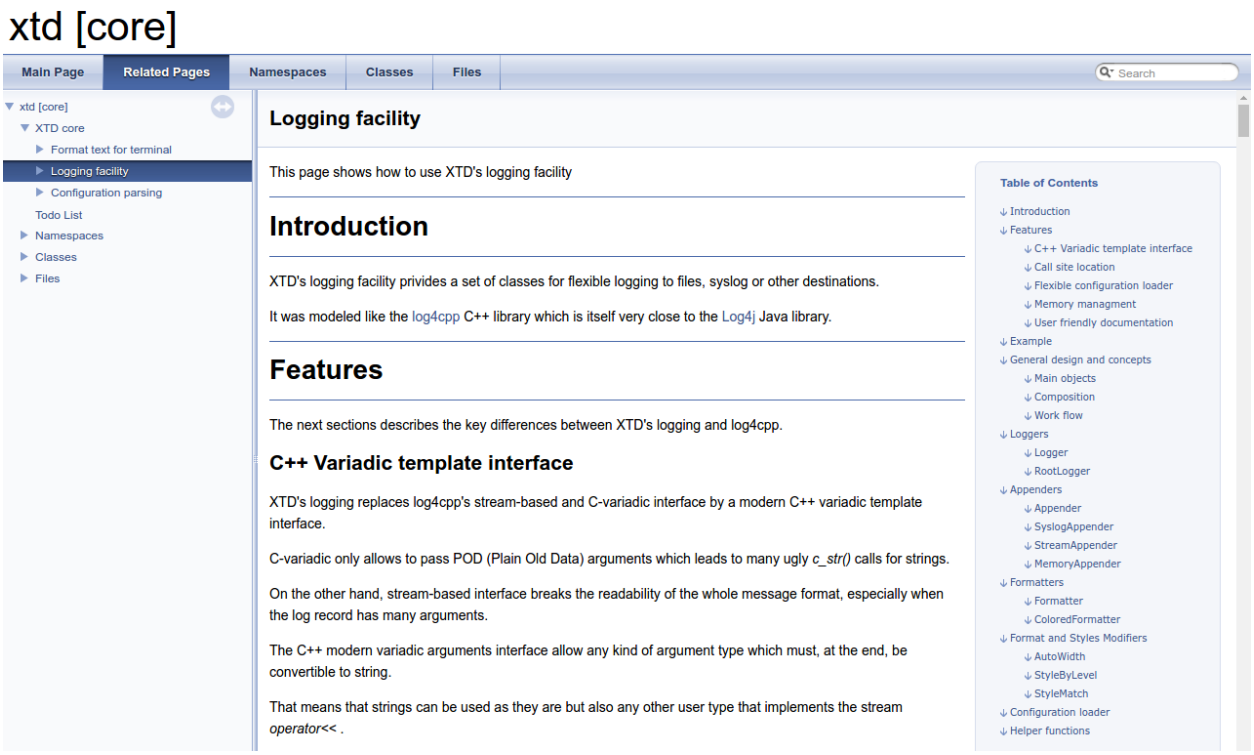
Warning: The dependency of cmake build system to the modification time of `INPUT` directories doesn't work with cmake versions prior to 3.0. This mean you must re-run cmake after adding new sources files in order to properly update the target files dependencies

4.1.7 Generated reports

XML : `reports/doc/xml/<module>/index.xml`

HTML : `reports/doc/html/<module>/index.html`

Bellow an example of generated html report :



4.2 DocCoverageRule

This modules generate a report about documentation's coverage.

4.2.1 Prerequisites

lcov Generates html report from coverage statistics. Available from ubuntu packages or from http://ltp.sourceforge.net/coverage/lcov.php

coverxygen Generate coverage statictics from doxygen xml output. Available from :

- Ubuntu PPA at https://launchpad.net/~psycofdj/+archive/ubuntu/coverxygen
- Python Package index : https://pypi.python.org/pypi/coverxygen/
- Source at https://github.com/psycofdj/coverxygen

DocRule This module must be enabled in order to load DocCoverageRule.

4.2.2 Functions

```
add_doc_coverage(<module>
  [ KIND  <kind>  [<kind>  ...]]
  [ SCOPE <scope> [<scope> ...]]
  [ MIN_PERCENT   <value> ]
  [ PREFIX        <path>  ]
)
```

This function generates cmake targets that produce reports that show your documentation's coverage. Generated targets are added as dependency of the global `doc-coverage` and `doc-coverage-clean` targets.

4.2.3 Parameters

module Name of the module. It determines the name of the generated cmake targets and the directory where targets generate the report.

KIND List of symbols to consider for coverage measurements. Available values are described by the `--kind` parameter of the `coverxygen` tools at https://github.com/psycofdj/coverxygen.

Default value is given by *`DocCoverageRule_DEFAULT_KIND`*.

SCOPE List of scope of symbol to consider for coverage measurements. Available values are described by the `--scope` parameter of the `coverxygen` tools at https://github.com/psycofdj/coverxygen.

Default value is given by *DocCoverageRule_DEFAULT_SCOPE*.

MIN_PERCENT Minimal percent of line coverage to consider target as successful. The target itself won't fail but generated JSON status will be tagged as failure.

Default value is given by *DocCoverageRule_DEFAULT_MIN_PERCENT*.

PREFIX Path prefix to remove from files in coverage interface.

Default value is given by *DocCoverageRule_DEFAULT_PREFIX*.

4.2.4 Global variables

DocCoverageRule_DEFAULT_KIND
"enum;typedef;variable;function;class;struct;define"

DocCoverageRule_DEFAULT_SCOPE
"public;protected"

DocCoverageRule_DEFAULT_MIN_PERCENT
"30"

DocCoverageRule_DEFAULT_PREFIX
"${CMAKE_CURRENT_SOURCE_DIR}/src"

4.2.5 Generated targets

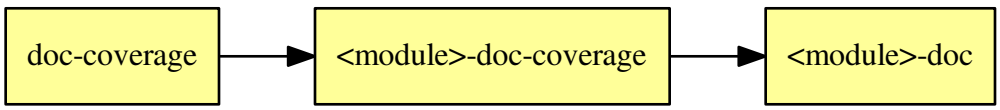
doc-coverage generate documentation coverage reports for all modules

doc-coverage-clean removes documentation coverage reports for all modules

<module>-doc-coverage generate documentation coverage report for module <*module*>

<module>-doc-coverage-clean removes documentation coverage report for module <*module*>

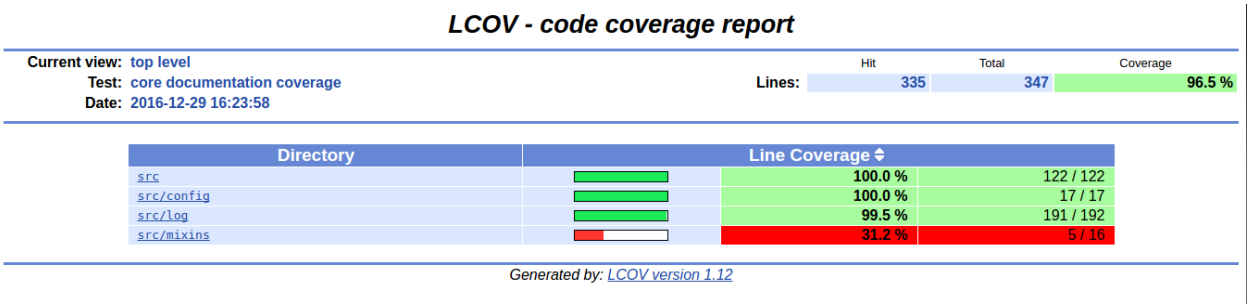
4.2.6 Dependencies



4.2.7 Generated reports

HTML : `reports/doc-coverage/<module>/index.html`

Bellow an example of generated html report :



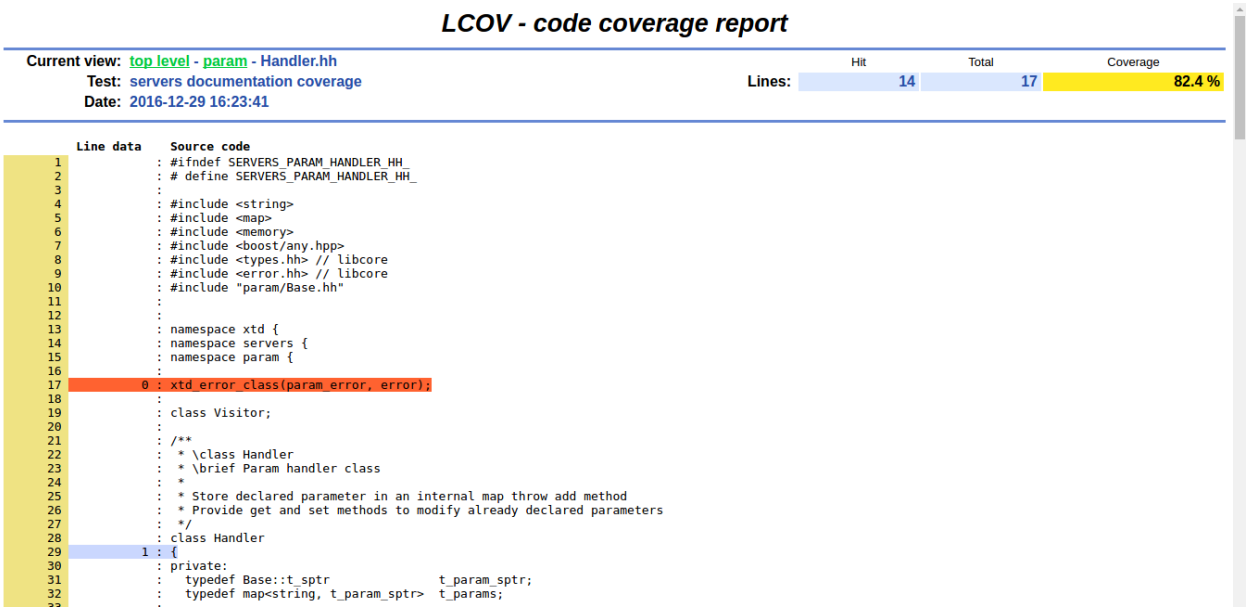
LCOV - code coverage report

Current view: top level
Test: core documentation coverage
Date: 2016-12-29 16:23:58

	Hit	Total	Coverage
Lines:	335	347	96.5 %

Directory	Line Coverage	
src	100.0 %	122 / 122
src/config	100.0 %	17 / 17
src/log	99.5 %	191 / 192
src/mixins	31.2 %	5 / 16

Generated by: LCOV version 1.12



LCOV - code coverage report

Current view: top level - param - Handler.hh
Test: servers documentation coverage
Date: 2016-12-29 16:23:41

	Hit	Total	Coverage
Lines:	14	17	82.4 %

```
      Line data    Source code
    1             : #ifndef SERVERS_PARAM_HANDLER_HH_
    2             : # define SERVERS_PARAM_HANDLER_HH_
    3             :
    4             : #include <string>
    5             : #include <map>
    6             : #include <memory>
    7             : #include <boost/any.hpp>
    8             : #include <types.hh> // libcore
    9             : #include <error.hh> // libcore
   10             : #include "param/Base.hh"
   11             :
   12             :
   13             : namespace xtd {
   14             : namespace servers {
   15             : namespace param {
   16             :
   17           0 : xtd_error_class(param_error, error);
   18             :
   19             : class Visitor;
   20             :
   21             : /**
   22             :  * \class Handler
   23             :  * \brief Param handler class
   24             :  *
   25             :  * Store declared parameter in an internal map throw add method
   26             :  * Provide get and set methods to modify already declared parameters
   27             :  */
   28             : class Handler
   29           1 : {
   30             : private:
   31             :   typedef Base::t_sptr              t_param_sptr;
   32             :   typedef map<string, t_param_sptr>  t_params;
```

JSON : `reports/doc-coverage/<module>/data.json`

```
[
  {
    "<path_to_file1>": [
      {
        "line": 53,
        "documented": true,
        "file": "/home/psyco/dev/xtdcpp/core/src/log/Appender.hh",
        "symbol": "xtd::log::Appender::Appender"
      },
      "..."
    ]
  },
  {
    "<path_to_file2>": [
      "..."
    ]
  }
]
```

JSON : `reports/doc-coverage/<module>/status.json`

```
{
  "status": "success",
  "graphs": [
```

(continues on next page)

(continued from previous page)

```
    {
      "data": {
        "labels": [],
        "datasets": [
          {
            "borderColor": "rgba(51, 204, 51, 0.5)",
            "pointBorderColor": "rgba(31, 122, 31, 1)",
            "yAxisID": "absolute",
            "label": "documented lines",
            "backgroundColor": "rgba(51, 204, 51, 0)",
            "pointBackgroundColor": "rgba(31, 122, 31, 1)",
            "data": "%(documented)d"
          },
          {
            "borderColor": "rgba(179, 0, 0, 0.5)",
            "pointBorderColor": "rgba(102, 0, 0, 1)",
            "yAxisID": "absolute",
            "label": "total lines",
            "backgroundColor": "rgba(179, 0, 0, 0)",
            "pointBackgroundColor": "rgba(102, 0, 0, 1)",
            "data": "%(total)d"
          },
          {
            "borderColor": "rgba(102, 153, 255, 0.5)",
            "pointBorderColor": "rgba(0, 60, 179, 1)",
            "yAxisID": "percent",
            "label": "% covered lines",
            "backgroundColor": "rgba(102, 153, 255, 0)",
            "pointBackgroundColor": "rgba(0, 60, 179, 1)",
            "data": "int((float(%(documented)d) / float(%(total)d)) * 100)"
          }
        ]
      },
      "type": "line",
      "options": {
        "scales": {
          "xAxes": [
            {
              "ticks": {
                "fontSize": 12,
                "minRotation": 80
              }
            }
          ],
          "yAxes": [
            {
              "position": "left",
              "ticks": {
                "fontSize": 24,
                "beginAtZero": true
              },
              "type": "linear",
              "id": "absolute",
              "display": true
            },
            {
              "position": "right",
```

(continues on next page)

(continued from previous page)


```
              "ticks": {
                "max": 100,
                "fontSize": 24,
                "beginAtZero": true
              },
              "type": "linear",
              "id": "percent"
            }
          ]
        },
        "title": {
          "text": "%(module)s : doc-coverage",
          "display": true
        }
      }
    }
  ],
  "data": {
    "documented": 335,
    "total": 347
  },
  "label": "96 %"
}
```


4.3 ClocRule

This module generates a report counting the number of code, blank and comments lines of your module.

4.3.1 Prerequisites

cloc Count line of code tool. Available from ubuntu packages (>= trusty) or from source at http://cloc.sourceforge.net/

xsltproc XSL Template rendering tool. Available from ubuntu packages or from source at http://xmlsoft.org/

4.3.2 Functions

```
add_cloc(module,
  [ INTPUT        <dir>     [ <dir>     ... ]],
  [ FILE_PATTERNS <pattern> [ <pattern> ... ]],
  [ MIN_PERCENT   <value> ]
)
```

This function generates cmake targets that produce cloc reports for a given module. Generated targets are added as dependency of the global `cloc` and `cloc-clean` targets.

4.3.3 Parameters

module Name of the module. It determines the name of the generated cmake targets and the directory where targets generate the report.

INPUT List of directories where target should search source files process. Default value is given by *ClocRule_DEFAULT_INPUT*

FILE_PATTERNS List of wildcards search files in given input directories. Default value is given by *ClocRule_DEFAULT_FILE_PATTERNS*

MIN_PERCENT Minimal percent of comment lines to consider target as successful. The target itself won't fail but generated JSON status will be tagged as failure.

Default value is given by *ClocRule_DEFAULT_MIN_PERCENT*.

4.3.4 Global variables

ClocRule_DEFAULT_INPUT
"${CMAKE_CURRENT_SOURCE_DIR}/src"

ClocRule_DEFAULT_FILE_PATTERNS
"*.cc;*.hh;*.hxx"

ClocRule_DEFAULT_MIN_PERCENT
"30"

4.3.5 Generated target

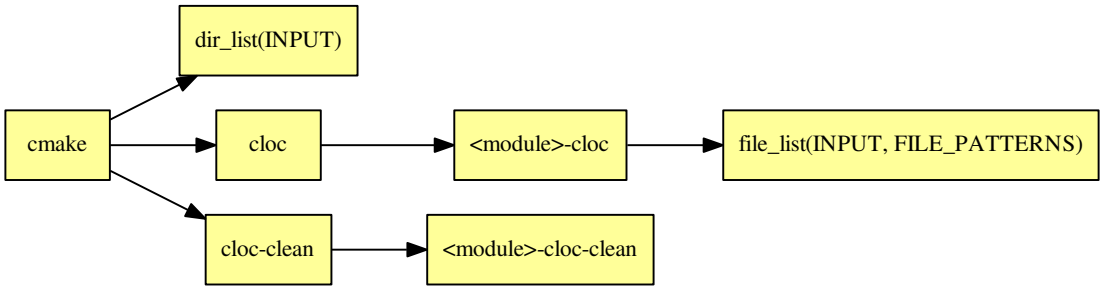
cloc generate cloc reports for all modules

cloc-clean removes cloc reports for all modules

<module>-cloc generate cloc report for module <*module*>

<module>-cloc-clean removes cloc report for module <*module*>

4.3.6 Dependencies



Warning: The dependency of cmake build system to the modification time of `INPUT` directories doesn't work with cmake versions prior to 3.0. This mean you must re-run cmake after adding new sources files in order to properly update the rule files dependencies

4.3.7 Generated reports

XML : `reports/cloc/<module>/cloc.xml`

```
<?xml version="1.0"?>
<results>
<header>
  <cloc_url>http://cloc.sourceforge.net</cloc_url>
  <cloc_version>1.60</cloc_version>
  <elapsed_seconds>0.14513897895813</elapsed_seconds>
  <n_files>43</n_files>
  <n_lines>6476</n_lines>
  <files_per_second>296.267758728031</files_per_second>
  <lines_per_second>44619.302454017</lines_per_second>
  <report_file>/home/psyco/dev/xtdcpp/.release/reports/core/cloc/cloc.xml</report_
↪file>
</header>
<files>
  <file name="/home/psyco/dev/xtdcpp/core/src/Application.cc" blank="73" comment="19"␣
↪ code="349" language="C++" />
  <!-- <file ...> -->
  <total blank="927" comment="2283" code="3266" />
</files>
<languages>
  <language name="C++" files_count="17" blank="410" comment="50" code="1981" />
  <language name="C/C++ Header" files_count="26" blank="517" comment="2233" code="1285
↪" />
  <total sum_files="43" blank="927" comment="2283" code="3266" />
</languages>
</results>
```

HTML : `reports/cloc/<module>/index.html`

Bellow an example of generated html report :

Summary

Language	File count	Blank lines	Comments	Code
C++	6	247	94	1095
C/C++ Header	6	221	332	752
	12	468	426	1847

File details

File	Language	Blank lines	Comments	Code
/home/psyco/dev/xtdcpp/servers/src/app/HttpServer.cc	C++	125	79	540
/home/psyco/dev/xtdcpp/servers/src/app/HtmlOArchive.cc	C++	81	9	388
/home/psyco/dev/xtdcpp/servers/src/param/Base.hh	C/C++ Header	68	156	187
/home/psyco/dev/xtdcpp/servers/src/app/HtmlOArchive.hh	C/C++ Header	33	0	169
/home/psyco/dev/xtdcpp/servers/src/app/HttpServer.hh	C/C++ Header	40	12	152
/home/psyco/dev/xtdcpp/servers/src/param/Handler.cc	C++	26	2	122
/home/psyco/dev/xtdcpp/servers/src/param/Handler.hh	C/C++ Header	46	108	108
/home/psyco/dev/xtdcpp/servers/src/app/Server.hh	C/C++ Header	20	1	87
/home/psyco/dev/xtdcpp/servers/src/param/Visitor.hh	C/C++ Header	14	55	49
/home/psyco/dev/xtdcpp/servers/src/param/Visitor.cc	C++	6	0	21
/home/psyco/dev/xtdcpp/servers/src/param/Base.cc	C++	5	0	14
/home/psyco/dev/xtdcpp/servers/src/app/Server.cc	C++	4	4	10
		468	426	1847

JSON : `reports/cloc/<module>/status.json`

```
{
  "status": "success",
  "graphs": [
    {
      "data": {
        "labels": [],
        "datasets": [
          {
            "borderColor": "rgba(51, 204, 51, 0.5)",
            "pointBorderColor": "rgba(31, 122, 31, 1)",
            "yAxisID": "absolute",
            "label": "comment lines",
            "backgroundColor": "rgba(51, 204, 51, 0)",
```

(continues on next page)

(continued from previous page)

```
            "pointBackgroundColor": "rgba(31, 122, 31, 1)",
            "data": "%(comment)d"
          },
          {
            "borderColor": "rgba(179, 0, 0, 0.5)",
            "pointBorderColor": "rgba(102, 0, 0, 1)",
            "yAxisID": "absolute",
            "label": "code lines",
            "backgroundColor": "rgba(179, 0, 0, 0)",
            "pointBackgroundColor": "rgba(102, 0, 0, 1)",
            "data": "%(code)d"
          },
          {
            "borderColor": "rgba(102, 153, 255, 0.5)",
            "pointBorderColor": "rgba(0, 60, 179, 1)",
            "yAxisID": "percent",
            "label": "% comment lines",
            "backgroundColor": "rgba(102, 153, 255, 0)",
            "pointBackgroundColor": "rgba(0, 60, 179, 1)",
            "data": "int(float(%(comment)d) / (float(%(comment)d) + float(%(code)d))␣
↪* 100)"
          }
        ]
      },
      "type": "line",
      "options": {
        "scales": {
          "xAxes": [
            {
              "ticks": {
                "fontSize": 12,
                "minRotation": 80
              }
            }
          ],
          "yAxes": [
            {
              "position": "left",
              "ticks": {
                "fontSize": 24,
                "beginAtZero": true
              },
              "type": "linear",
              "id": "absolute",
              "display": true
            },
            {
              "position": "right",
              "ticks": {
                "max": 100,
                "fontSize": 24,
                "beginAtZero": true
              },
              "type": "linear",
              "id": "percent"
            }
          ]
```

(continues on next page)

(continued from previous page)

```
        },
        "title": {
          "text": "%(module)s : cloc",
          "display": true
        }
      }
    }
  ],
  "data": {
    "comment": 2283,
    "code": 3266
  },
  "label": "41 %"
}
```

4.4 CppcheckRule

This module generates a report from result of cppcheck static analysis.

4.4.1 Prerequisites

cppcheck Static C++ code analyzer tool. Available from ubuntu packages or from source at http://cppcheck.sourceforge.net/

xsltproc XSL Template rendering tool. Available from ubuntu packages or from source at http://xmlsoft.org/

4.4.2 Functions

```
add_cppcheck(module,
  [INTPUT        <dir>     [ <dir>     ... ]],
  [FILE_PATTERNS <pattern> [ <pattern> ... ]]
)
```

This function generates cmake targets that produce cppcheck reports for a given module. Generated targets are added as dependency of the global `cppcheck` and `cppcheck-clean` targets.

4.4.3 Parameters

module Name of the module. It determines the name of the generated cmake targets and the directory where targets generate the report.

INPUT List of directories where target should search source files process. Default value is given by *CppcheckRule_DEFAULT_INPUT*

FILE_PATTERNS List of wildcards search files in given input directories. Default value is given by *CppcheckRule_DEFAULT_FILE_PATTERNS*

4.4.4 Global variables

CppcheckRule_DEFAULT_INPUT
"${CMAKE_CURRENT_SOURCE_DIR}/src"

CppcheckRule_DEFAULT_FILE_PATTERNS
"*.cc;*.hh;*.hxx"

4.4.5 Generated targets

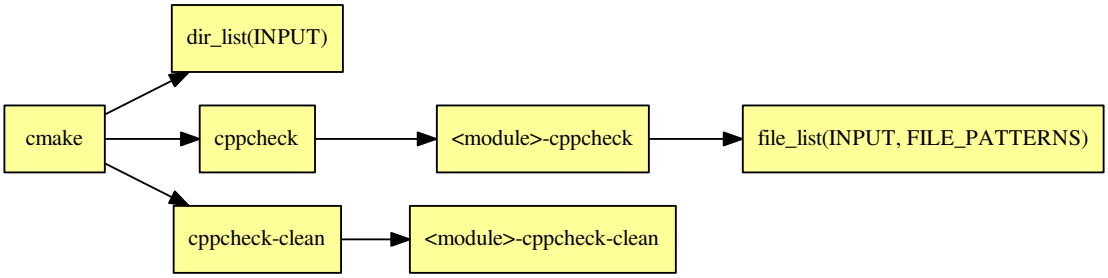
cppcheck generate cppcheck reports for all modules

cppcheck-clean removes cppcheck reports for all modules

<module>-cppcheck generate cppcheck report for module <*module*>

<module>-cppcheck-clean removes cppcheck report for module <*module*>

4.4.6 Dependencies

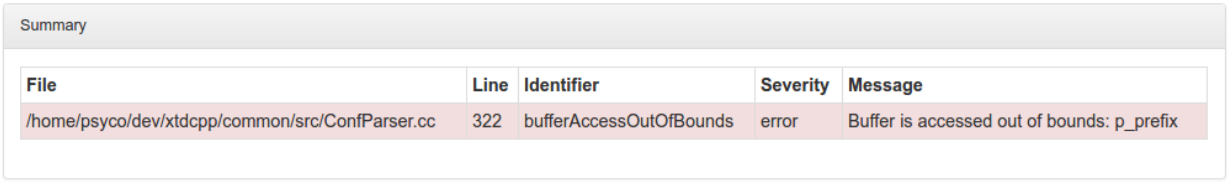


Warning: The dependency of cmake build system to the modification time of `INPUT` directories doesn't work with cmake versions prior to 3.0. This mean you must re-run cmake after adding new sources files in order to properly update the rule files dependencies

4.4.7 Generated reports

HTML : `reports/cppcheck/<module>/index.html`

Bellow an example of generated html report :



Summary

File	Line	Identifier	Severity	Message
/home/psyco/dev/xtdcpp/common/src/ConfParser.cc	322	bufferAccessOutOfBounds	error	Buffer is accessed out of bounds: p_prefix

XML : `reports/cppcheck/<module>/cppcheck.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<results version="2">
  <cppcheck version="1.72"/>
  <errors>
    <error id="duplicateExpression" severity="style" msg="Same expression on both␣
↪sides of '<='." verbose="Finding the same expression on both sides of an operator␣
↪is suspicious and might indicate a cut and paste or logic error. Please examine␣
↪this code carefully to determine if it is correct.">
      <location file="functions.hh" line="12"/>
      <location file="functions.hh" line="12"/>
    </error>
    <error id="bitwiseOnBoolean" severity="style" msg="Boolean variable 'test1' is␣
↪used in bitwise operation. Did you mean '&&'?" verbose="Boolean variable 'test1' is␣
↪used in bitwise operation. Did you mean '&&'?" inconclusive="true">
      <location file="functions.hh" line="22"/>
    </error>
  </errors>
</results>
```

JSON : `reports/cppcheck/<module>/status.json`

```
{
  "status": "success",
  "graphs": [
    {
      "data": {
        "labels": [],
        "datasets": [
          {
            "borderColor": "rgba(179, 0, 0, 0.5)",
            "pointBorderColor": "rgba(102, 0, 0, 1)",
            "yAxisID": "absolute",
            "label": "cppcheck error count",
            "backgroundColor": "rgba(179, 0, 0, 0.5)",
            "pointBackgroundColor": "rgba(102, 0, 0, 1)",
            "data": "%(total)d"
          }
        ]
      },
      "type": "line",
      "options": {
        "scales": {
```

(continues on next page)

(continued from previous page)

```
          "xAxes": [
            {
              "ticks": {
                "fontSize": 12,
                "minRotation": 80
              }
            }
          ],
          "yAxes": [
            {
              "position": "left",
              "ticks": {
                "fontSize": 24,
                "beginAtZero": true
              },
              "type": "linear",
              "id": "absolute",
              "display": true
            }
          ]
        },
        "title": {
          "text": "%(module)s : cppcheck",
          "display": true
        }
      }
    }
  ],
  "data": {
    "total": 0
  },
  "label": "0"
}
```

4.5 CheckRule

This module create targets that runs and generate reports about unit-tests.

- *About debugger*
- *Target Dependencies*
- *Generated reports*

4.5.1 Prerequisites

enable_testing() This module requires that `enable_testing()` is called at top level CMakeLists.txt.

xsltproc XSL Template rendering tool. Available from ubuntu packages or from source at http://xmlsoft.org/

4.5.2 Functions

```
add_check(<module>
  [ PATTERNS  <pattern>     [<pattern>   ...]]
  [ INCLUDES  <dir>         [<dir>       ...]]
  [ LINKS     <lib>         [<lib>       ...]]
  [ ENV       <key>=<value> [<key=value> ...]]
  [ ARGS      <arg>         [<arg>       ...]]
  [ DIRECTORY      <dir>  ]
  [ PREFIX         <str>  ]
  [ JOBS           <int>  ]
  [ CMAKEVARS_NAME <name> ]
  [ NO_DEFAULT_ENV        ]
  [ NO_DEFAULT_ARGS       ]
  [ NO_DEFAULT_INCLUDES   ]
  [ NO_DEFAULT_LINKS      ]
)
```

This function automatically detects tests source files, creates binary targets and generate test report.

4.5.3 Parameters

module Name of the module. It determines the name of the generated cmake targets and the directory where targets generate the report.

PATTERNS List of file extensions to match while searching for tests. See *details* about how tests are automatically detected by this module.

Default value is given by *CheckRule_DEFAULT_PATTERNS*

> **Warning:** Items given in `PATTERNS` list are not wildcards but only file extensions (ie: no asterix)

INCLUDES List of include directories to add when compiling test sources. Each item will be added through cmake `target_include_directories` directive.

> **Warning:** When using cmake version prior to 2.8.12, test include directories are added through cmake `include_directories`. Therefore, they will also be added to your CMakeLists.txt targets.

Default value is given by *CheckRule_DEFAULT_INCLUDES* unless NO_DEFAULT_INCLUDES option is given.

LINKS List of libraries to add when linking test binaries. Each item will be added through cmake target_link_directories directive.

Default value is given by *CheckRule_DEFAULT_LINKS* unless NO_DEFAULT_LINKS option is given.

ENV List of environment variable to defined before running each test.

Default value is given by *CheckRule_DEFAULT_ENV* unless NO_DEFAULT_ENV option is given.

ARGS List of command-line options to pass when running test binaries.

Default value is given by *CheckRule_DEFAULT_ARGS* unless NO_DEFAULT_ARGS option is given.

Tip: This option is a convenient way to give your tests some informations about source and build directory tree.

Default value is given *CheckRule_DEFAULT_ARGS*

DBG_ARGS List of command-line options to pass when running test through debugger. It Usually sets arguments to command line to prevent your test framework to protect run with forks, allowing to get a usable frame-stack to investigate crashes.

Default value is given by *CheckRule_DEFAULT_DBG_ARGS* unless NO_DEFAULT_ARGS option is given.

Default value is given *CheckRule_DEFAULT_DBG_ARGS*

DIRECTORY Directory to search tests source files. See *details* about how tests are automatically detected by this module.

Default value is given *CheckRule_DEFAULT_DIRECTORY*

PREFIX Filename prefix of test source files. See *details* about how tests are automatically detected by this module.

Default value is given *CheckRule_DEFAULT_PREFIX*

JOBS Number of simultaneous test to run when target is called.

Default value is given *CheckRule_DEFAULT_JOBS*

CMAKEVARS_NAME Path to header file generated by check rule. See *details* about how getting information about source/build tree in your test code.

Default value is given *CheckRule_DEFAULT_CMAKEVARS_NAME*

NO_DEFAULT_ENV If option is given, don't use *CheckRule_DEFAULT_ENV*

NO_DEFAULT_ARGS If option is given, don't use *CheckRule_DEFAULT_ARGS*

NO_DEFAULT_INCLUDES If option is given, don't use *CheckRule_DEFAULT_INCLUDES*

NO_DEFAULT_LINKS If option is given, don't use *CheckRule_DEFAULT_LINKS*

4.5.4 Global variables

CheckRule_DEFAULT_PATTERNS
".c;.cc;.cpp"

CheckRule_DEFAULT_INCLUDES
""

CheckRule_DEFAULT_LINKS

```
""

CheckRule_DEFAULT_ENV
""

CheckRule_DEFAULT_DIRECTORY
"${CMAKE_CURRENT_SOURCE_DIR}/unit"

CheckRule_DEFAULT_PREFIX
"Test"

CheckRule_DEFAULT_JOBS
"1"

CheckRule_DEFAULT_ARGS
""

CheckRule_DEFAULT_DBG_ARGS
""

CheckRule_DEFAULT_CMAKEVARS_NAME
"${CMAKE_CURRENT_BINARY_DIR}/cmakevars.h"

CheckRule_DEFAULT_TIMEOUT
"120"
```

4.5.5 Getting path informations in tests

Tests often need to read sample files located in either source or build directory. Because source and build trees are not relative to each others, only CMake knows where both root directories are located.

XTDMake's CheckRule provides two ways to forward this informations to your tests :

1. Using `ARGS` and/or `CheckRule_DEFAULT_ARGS` to add command line parameters built with CMake variables such as :

   - `CMAKE_SOURCE_DIR` : top source directory
   - `CMAKE_BINARY_DIR` : top build directory
   - `CMAKE_CURRENT_SOURCE_DIR` : current module's source directory
   - `CMAKE_CURRENT_BINARY_DIR` : current module's build directory

   One possible value for `CheckRule_DEFAULT_ARGS` could be:

   ```
   --topsrc-dir=\${CMAKE_PROJECT_SOURCE_DIR} \
   --topbuild-dir=\${CMAKE_PROJECT_BINARY_DIR} \
   --src-dir=\${CMAKE_SOURCE_DIR} \
   --build-dir=\${CMAKE_BINARY_DIR}
   ```

2. Using generated header file. CheckRule automatically creates for each module an header file named `CheckRule_DEFAULT_CMAKEVARS` or `CMAKEVARS` arguments. This file is generated from the given template :

   ```
   #define   TOP_SRCDIR       "@CMAKE_SOURCE_DIR@"
   #define       SRCDIR       "@CMAKE_CURRENT_SOURCE_DIR@"
   #define TOP_BUILDDIR       "@PROJECT_BINARY_DIR@"
   #define     BUILDDIR       "@CMAKE_CURRENT_BINARY_DIR@"
   #define PROJECT_SOURCE_DIR "@PROJECT_SOURCE_DIR@"
   #define PROJECT_BINARY_DIR "@PROJECT_BINARY_DIR@"
   ```

Your test code can simply include the generated header and use defined variables to build path to your assets files located in source or build tree.

4.5.6 Finding the test sources

This module scans given `DIRECTORY` for source files prefixed by `PREFIX` and matches one of file extensions given by `PATTERNS`. Each matched file is considered as a standalone executable test.

4.5.7 Deducing the target name

This function deduces the name of the test from its source file by stripping `DIRECTORY`, `PREFIX` and match extension. Example :

file ./unit/TestApplication.cc

DIRECTORY ./unit

PATTERNS .cc;.cpp.c

Deduced name Application

4.5.8 Generated targets

check generate doc reports for all modules

check-clean removes doc reports for all modules

<module>-check generate unittests report for module <*module*>

<module>-check-build build all test binaries for module <*module*>

<module>-check-run run tests for module <*module*> that are not up-to-date

<module>-check-run-verbose run tests for module <*module*> that are not up-to-date with ctest verbose output

<module>-check-run-forced run all tests for module <*module*>

<module>-check-clean clean test targets for module <*module*>

For each test <*name*>, the function also produces :

t<name> build individual test binary target <*name*>

<module>-check-ut-<name> run individual test <*name*>

<module>-check-ut-<name>-dbg run individual test <*name*> wrapped in debugger

<module>-check-ut-<name>-cmd prints individual test command <*name*>

4.5.9 Adding test manually

To integrate manually defined tests with CheckRule module, you must use the following function.

> **Warning:** This function must be called **before** `add_check`

```
add_check_test(module name
  COMMAND <command> [ <arg> ... ]
  [ ENVIRONMENT <var>=<value> [ <var>=<value> ... ]
)
```

module name of targeted module

name name of the test target

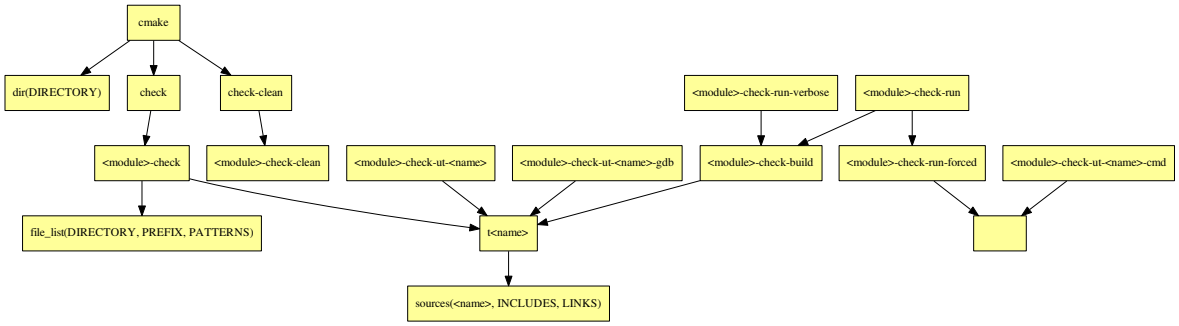
COMMAND command line to run for this test

ENVIRONMENT environment variable to define before running the test

4.5.10 About debugger

By default, CheckRule debugger target wraps test execution in `GNU gdb`. If `USE_CLANG` variable is defined, debugger is switched to `lldb`.

4.5.11 Target Dependencies

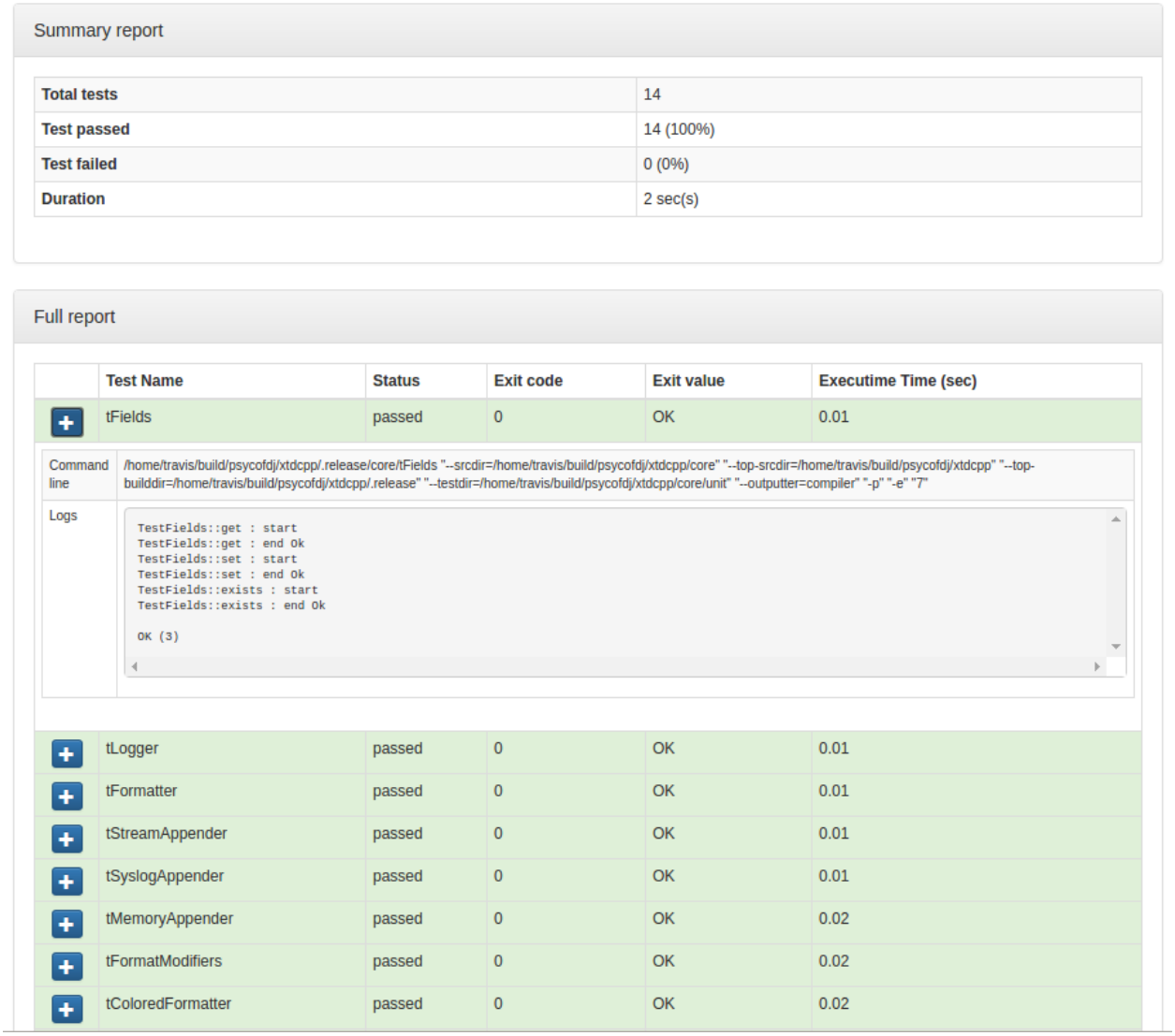


Warning: The dependency of cmake build system to the modification time of `DIRECTORY` doesn't work with cmake versions prior to 3.0. This mean you must re-run cmake after adding new sources files in order to properly update the rule files dependencies.

4.5.12 Generated reports

HTML : `reports/check/<module>/index.html`

Bellow an example of generated html report :



Summary report

Total tests	14
Test passed	14 (100%)
Test failed	0 (0%)
Duration	2 sec(s)

Full report

	Test Name	Status	Exit code	Exit value	Executime Time (sec)
+	tFields	passed	0	OK	0.01
Command line	/home/travis/build/psycofdj/xtdcpp/.release/core/tFields "--srcdir=/home/travis/build/psycofdj/xtdcpp/core" "--top-srcdir=/home/travis/build/psycofdj/xtdcpp" "--top-builddir=/home/travis/build/psycofdj/xtdcpp/.release" "--testdir=/home/travis/build/psycofdj/xtdcpp/core/unit" "--outputter=compiler" "-p" "-e" "7"				
Logs	TestFields::get : start TestFields::get : end Ok TestFields::set : start TestFields::set : end Ok TestFields::exists : start TestFields::exists : end Ok OK (3)				
+	tLogger	passed	0	OK	0.01
+	tFormatter	passed	0	OK	0.01
+	tStreamAppender	passed	0	OK	0.01
+	tSyslogAppender	passed	0	OK	0.01
+	tMemoryAppender	passed	0	OK	0.02
+	tFormatModifiers	passed	0	OK	0.02
+	tColoredFormatter	passed	0	OK	0.02

XML : `reports/check/<module>/index.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<Site BuildName="(empty)"
      BuildStamp="20161231-1237-Experimental"
      Name="(empty)"
      Generator="ctest-3.5.1"
      CompilerName=""
      CompilerVersion=""
      OSName="Linux"
      Hostname="PSYCO-INTEL"
      OSRelease="4.4.0-57-generic"
      OSVersion="#78-Ubuntu SMP Fri Dec 9 23:50:32 UTC 2016"
      OSPlatform="x86_64"
      Is64Bits="1"
      VendorString="GenuineIntel"
      VendorID="Intel Corporation"
      FamilyID="6"
      ModelID="79"
```

(continues on next page)

(continued from previous page)

```
      ProcessorCacheSize="20480"
      NumberOfLogicalCPU="16"
      NumberOfPhysicalCPU="1"
      TotalVirtualMemory="93"
      TotalPhysicalMemory="64340"
      LogicalProcessorsPerPhysical="16"
      ProcessorClockFrequency="1898.75"
      >
      <Testing>
              <StartDateTime>Dec 31 13:37 CET</StartDateTime>
              <StartTestTime>1483187874</StartTestTime>
              <TestList>
                      <Test>./tApplication</Test>
              </TestList>
              <Test Status="passed">
                      <Name>tConfigParser</Name>
                      <Path>.</Path>
                      <FullName>./tConfigParser</FullName>
                      <FullCommandLine>/home/psyco/dev/xtdcpp/.release/core/
↪tConfigParser "--srcdir=/home/psyco/dev/xtdcpp/core" "--top-srcdir=/home/psyco/dev/
↪xtdcpp" "--top-builddir=/home/psyco/dev/xtdcpp/.release" "--testdir=/home/psyco/dev/
↪xtdcpp/core/unit" "--outputter=compiler" "-p" "-e" "7"</FullCommandLine>
                      <Results>
                              <NamedMeasurement type="numeric/double" name="Execution␣
↪Time">
                                      <Value>0.0134299</Value>
                              </NamedMeasurement>
                              <NamedMeasurement type="text/string" name="Completion␣
↪Status">
                                      <Value>Completed</Value>
                              </NamedMeasurement>
                              <NamedMeasurement type="text/string" name="Command Line
↪">
                                      <Value>/home/psyco/dev/xtdcpp/.release/core/
↪tConfigParser "--srcdir=/home/psyco/dev/xtdcpp/core" "--top-srcdir=/home/psyco/dev/
↪xtdcpp" "--top-builddir=/home/psyco/dev/xtdcpp/.release" "--testdir=/home/psyco/dev/
↪xtdcpp/core/unit" "--outputter=compiler" "-p" "-e" "7"</Value>
                              </NamedMeasurement>
                              <Measurement>
                                      <Value>
          TestConfParser::Constructor : start
          TestConfParser::Constructor : end Ok
          TestConfParser::parse : start
          TestConfParser::parse : end Ok
          TestConfParser::get : start
          TestConfParser::get : end Ok
          TestConfParser::search : start
          TestConfParser::search : end Ok
          TestConfParser::setParams : start
          TestConfParser::setParams : end Ok
          TestConfParser::parseFile : start
          TestConfParser::parseFile : end Ok
          OK (6)
        </Value>
                              </Measurement>
                      </Results>
              </Test>
```

(continues on next page)

(continued from previous page)

```
            <EndDateTime>Dec 31 13:37 CET</EndDateTime>
            <EndTestTime>1483187875</EndTestTime>
            <ElapsedMinutes>0</ElapsedMinutes>
      </Testing>
</Site>
```

JSON : `reports/check/<module>/status.json`

```
{
  "status": "success",
  "graphs": [
    {
      "data": {
        "labels": [],
        "datasets": [
          {
            "borderColor": "rgba(51, 204, 51, 0.5)",
            "pointBorderColor": "rgba(31, 122, 31, 1)",
            "yAxisID": "absolute",
            "label": "success tests",
            "backgroundColor": "rgba(51, 204, 51, 0)",
            "pointBackgroundColor": "rgba(31, 122, 31, 1)",
            "data": "%(success)d"
          },
          {
            "borderColor": "rgba(179, 0, 0, 0.5)",
            "pointBorderColor": "rgba(102, 0, 0, 1)",
            "yAxisID": "absolute",
            "label": "failure tests",
            "backgroundColor": "rgba(179, 0, 0, 0)",
            "pointBackgroundColor": "rgba(102, 0, 0, 1)",
            "data": "%(failures)d"
          }
        ]
      },
      "type": "line",
      "options": {
        "scales": {
          "xAxes": [
            {
              "ticks": {
                "fontSize": 12,
                "minRotation": 80
              }
            }
          ],
          "yAxes": [
            {
              "position": "left",
              "ticks": {
                "fontSize": 24,
                "beginAtZero": true
              },
              "type": "linear",
              "id": "absolute",
              "display": true
            }
```

(continues on next page)

(continued from previous page)

```
          ]
        },
        "title": {
          "text": "%(module)s : unittests",
          "display": true
        }
      }
    }
  ],
  "data": {
    "failures": 0,
    "success": 14
  },
  "label": "14 / 14"
}
```

4.6 CovRule

This module generates a report from result of cppcheck static analysis.

4.6.1 Prerequisites

lcov Generates html report from coverage statistics. Available from ubuntu packages or from http://ltp.sourceforge.net/coverage/lcov.php

CheckRule This module must be enabled in order to load CovRule.

4.6.2 Functions

```
add_cov(<module>
  [ EXCLUDE_PATTERNS <pattern> [ <pattern> .. ]]
  [ MIN_PERCENT      <value> ]
)
```

This function generates cmake targets that produce reports that show your code coverage. Generated targets are added as dependency of the global `cov` and `doc-clean` targets.

4.6.3 Parameters

module Name of the module. It determines the name of the generated cmake targets and the directory where targets generate the report.

EXCLUDE_PATTERNS List of files patterns to exclude from for coverage measurements.

Default value is given by *CovRule_DEFAULT_EXCLUDE_PATTERNS*.

MIN_PERCENT Minimal percent of line coverage to consider target as successful. The target itself won't fail but generated JSON status will be tagged as failure.

Default value is given by *CovRule_DEFAULT_MIN_PERCENT*.

4.6.4 Global variables

CovRule_DEFAULT_EXCLUDE_PATTERNS
"Test*.*"

CovRule_DEFAULT_MIN_PERCENT
"30"

4.6.5 Generated targets

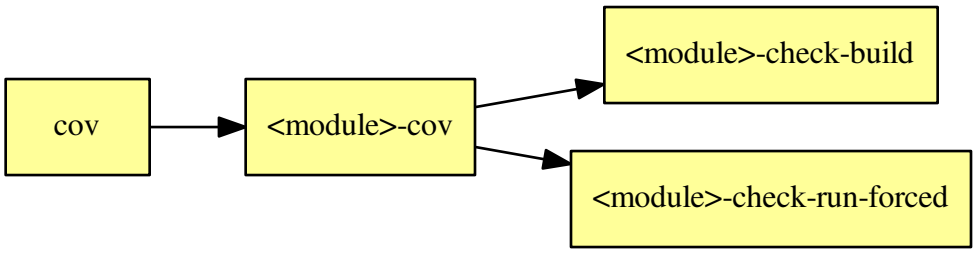
cov generate coverage reports for all modules

cov-clean removes coverage reports for all modules

<module>-cov generate coverage report for module <*module*>

<module>-cov-clean removes coverage report for module <*module*>

4.6.6 Dependencies



4.6.7 Generated reports

HTML : `reports/coverage/<module>/index.html`

Bellow an example of generated html report :

LCOV - code coverage report

			Hit	Total	Coverage
Current view:	top level	Lines:	1403	1494	93.9 %
Test:	core unit test coverage	Functions:	525	659	79.7 %
Date:	2016-12-29 16:24:50				
Legend:	Rating: low: < 75 % medium: >= 75 % high: >= 90 %				

Directory	Line Coverage		Functions	
src	82.1 %	416 / 507	83.6 %	188 / 225
src/config	100.0 %	248 / 248	89.6 %	60 / 67
src/log	100.0 %	733 / 733	75.2 %	273 / 363
src/mixins	100.0 %	6 / 6	100.0 %	4 / 4

Generated by: LCOV version 1.12

```
      :
      : bool
    3 : attrs::from_string(const string& p_value, attrs& p_attrs)
      : {
      :   const static map<string, attrs> l_values = {
      :     { "",           unset      },
      :     { "unset",      unset      },
      :     { "reset",      reset      },
      :     { "dim",        dim        },
      :     { "bold",       bold       },
      :     { "underlined", underlined },
      :     { "blink",      blink      },
      :     { "reverse",    reverse    },
      :     { "hidden",     hidden     }
    3 :   };
      :
    3 :   auto c_item = l_values.find(p_value);
    3 :   if (l_values.end() == c_item)
    1 :     return false;
      :
    2 :   p_attrs = c_item->second;
    2 :   return true;
      : }
      :
      : string
    0 : attrs::to_string(const attrs& p_attrs)
      : {
      :   const static map<typename std::underlying_type<s>::type, string> l_values = {
    0 :     { valueof(s::dim),        "dim"        },
    0 :     { valueof(s::bold),       "bold"       },
    0 :     { valueof(s::underlined), "underlined" },
    0 :     { valueof(s::blink),      "blink"      },
    0 :     { valueof(s::reverse),    "reverse"    },
    0 :     { valueof(s::hidden),     "hidden"     }
    0 :   };
      :
    0 :   if (0 == p_attrs.m_attrs.size())
    0 :     return "unset";
      :
    0 :   vector<string> l_parts;
    0 :   for (auto c_attr : p_attrs.m_attrs) {
    0 :     auto c_item = l_values.find(c_attr);
    0 :     if (l_values.end() != c_item)
    0 :       l_parts.push_back(c_item->second);
      :   }
    0 :   return boost::join(l_parts, " | ");
      : }
```

XML : `reports/coverage/<module>/coverage.xml`

```
<?xml version="1.0" ?>
<!DOCTYPE coverage
  SYSTEM 'http://cobertura.sourceforge.net/xml/coverage-04.dtd'>
<coverage branch-rate="0.0" branches-covered="0" branches-valid="0" complexity="0"␣
↪line-rate="0.939089692102" lines-covered="1403" lines-valid="1494" timestamp=
↪"1483189103" version="2.0.3">
```

(continues on next page)

(continued from previous page)

```
      <sources>
              <source>.</source>
      </sources>
      <packages>
              <package branch-rate="0.0" complexity="0" line-rate="1.0" name="......
↪core.src.config">
                      <classes>
                              <class branch-rate="0.0" complexity="0" filename="../../
↪core/src/config/Grammar.hxx" line-rate="1.0" name="......core.src.config.Grammar.hxx
↪">
                                      <methods>
                                              <method branch-rate="0.0" line-rate="0.0
↪" name="xtd::config::impl::Grammar&lt;std::istream_iterator&lt;char, char,␣
↪std::char_traits&lt;char&gt;, long&gt; &gt;::handleError(boost::spirit::line_pos_
↪iterator&lt;std::istream_iterator&lt;char, char, std::char_traits&lt;char&gt;, long&
↪gt; &gt;, boost::spirit::line_pos_iterator&lt;std::istream_iterator&lt;char, char,␣
↪std::char_traits&lt;char&gt;, long&gt; &gt;, boost::spirit::line_pos_iterator&lt;
↪std::istream_iterator&lt;char, char, std::char_traits&lt;char&gt;, long&gt; &gt;,␣
↪boost::spirit::info const&amp;)" signature="">
                                                      <lines>
                                                              <line branch="false"␣
↪hits="0" number="124"/>
                                                      </lines>
                                              </method>
                                              <method branch-rate="1.0" line-rate="1.0
↪" name="xtd::config::impl::Grammar&lt;__gnu_cxx::__normal_iterator&lt;char*, std::__
↪cxx11::basic_string&lt;char, std::char_traits&lt;char&gt;, std::allocator&lt;char&
↪gt; &gt; &gt; &gt;::Grammar()" signature="">
                                                      <lines>
                                                              <line branch="false"␣
↪hits="20" number="10"/>
                                                      </lines>
                                              </method>
                                      </methods>
                                      <lines>
                                              <line branch="false" hits="23" number=
↪"10"/>
                                              <line branch="false" hits="23" number=
↪"11"/>
                                              <line branch="false" hits="46" number=
↪"26"/>
                                              <line branch="false" hits="46" number=
↪"27"/>
                                              <line branch="false" hits="23" number=
↪"28"/>
                                      </lines>
                              </class>
                      </classes>
              </package>
      </packages>
</coverage>
```

JSON : `reports/coverage/<module>/status.json`

```
{
  "status": "success",
  "graphs": [
```

(continues on next page)

(continued from previous page)

```
    {
      "data": {
        "labels": [],
        "datasets": [
          {
            "borderColor": "rgba(51, 204, 51, 0.5)",
            "pointBorderColor": "rgba(31, 122, 31, 1)",
            "yAxisID": "absolute",
            "label": "covered lines",
            "backgroundColor": "rgba(51, 204, 51, 0)",
            "pointBackgroundColor": "rgba(31, 122, 31, 1)",
            "data": "%(covered)d"
          },
          {
            "borderColor": "rgba(179, 0, 0, 0.5)",
            "pointBorderColor": "rgba(102, 0, 0, 1)",
            "yAxisID": "absolute",
            "label": "total lines",
            "backgroundColor": "rgba(179, 0, 0, 0)",
            "pointBackgroundColor": "rgba(102, 0, 0, 1)",
            "data": "%(total)d"
          },
          {
            "borderColor": "rgba(102, 153, 255, 0.5)",
            "pointBorderColor": "rgba(0, 60, 179, 1)",
            "yAxisID": "percent",
            "label": "% covered lines",
            "backgroundColor": "rgba(102, 153, 255, 0)",
            "pointBackgroundColor": "rgba(0, 60, 179, 1)",
            "data": "int((float(%(covered)d) / float(%(total)d)) * 100)"
          }
        ]
      },
      "type": "line",
      "options": {
        "scales": {
          "xAxes": [
            {
              "ticks": {
                "fontSize": 12,
                "minRotation": 80
              }
            }
          ],
          "yAxes": [
            {
              "position": "left",
              "ticks": {
                "fontSize": 24,
                "beginAtZero": true
              },
              "type": "linear",
              "id": "absolute",
              "display": true
            },
            {
              "position": "right",
```

(continues on next page)

(continued from previous page)

```
              "ticks": {
                "max": 100,
                "fontSize": 24,
                "beginAtZero": true
              },
              "type": "linear",
              "id": "percent"
            }
          ]
        },
        "title": {
          "text": "%(module)s : coverage",
          "display": true
        }
      }
    }
  ],
  "data": {
    "covered": 1403,
    "total": 1494,
    "percent": "int((float(%(covered)d) / float(%(total)d)) * 100)"
  },
  "label": "93 %"
}
```

4.7 MemcheckRule

This module generates a report that shows memory defects detected by valgrind for available unit tests.

4.7.1 Prerequisites

valgrind Instrumentation framework for building dynamic analysis tools. Available from ubuntu packages or from source at http://valgrind.org/

CheckRule This module must be enabled in order to load MemcheckRule.

4.7.2 Functions

```
add_memcheck(<module>
[SUPPRESSIONS <file> [<file> ... ]]
[EXTRA_ARGS   <args>]
)
```

This function generates cmake targets that produce reports that show memory flaws detected by valgrind on module's test suite. Generated targets are added as dependency of the global `memcheck` and `memcheck-clean` targets.

4.7.3 Parameters

module Name of the module. It determines the name of the generated cmake targets and the directory where targets generate the report.

SUPPRESSIONS List of existing files to add as valgrind supression stacks. See http://valgrind.org/docs/manual/manual-core.html#manual-core.suppress

EXTRA_ARGS List of additional arguments to pass to valgrind. Use with caution, parameters must be compatible with `--tool=memcheck`.

4.7.4 Generated targets

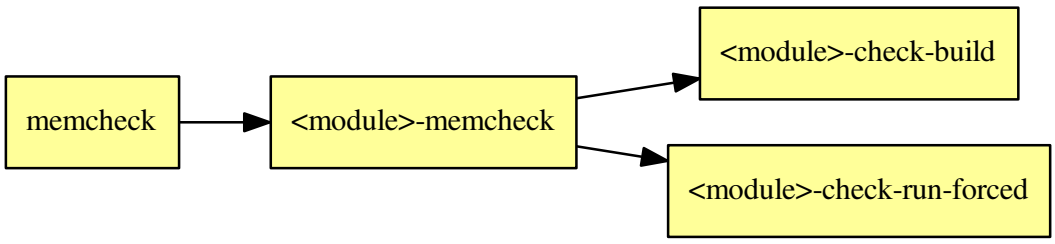
`memcheck` generate memory reports for all modules

`memcheck-clean` removes memory reports for all modules

`<module>-memcheck` generate memory report for module <*module*>

`<module>-memcheck-clean` removes memory report for module <*module*>

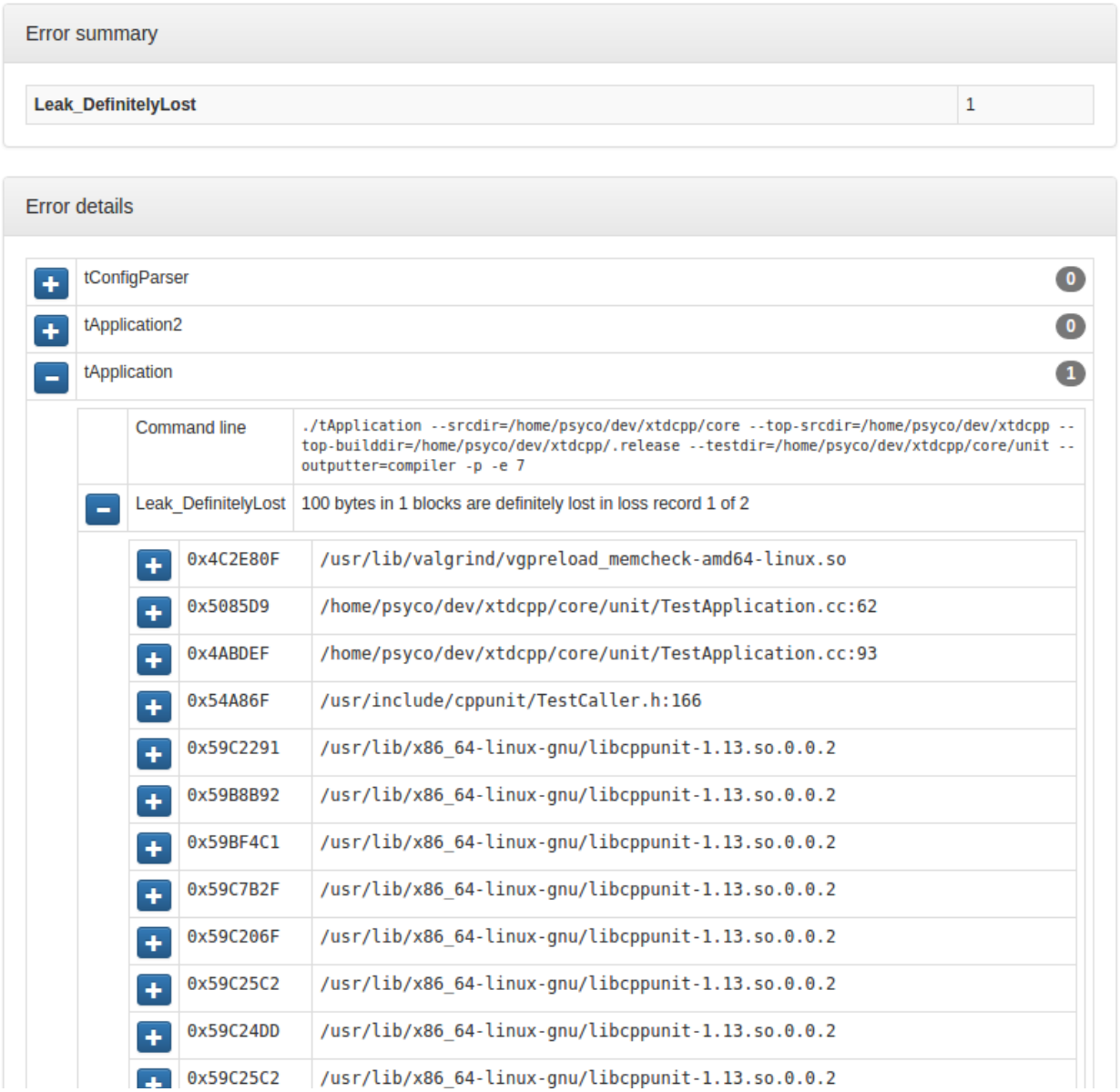
4.7.5 Dependencies



4.7.6 Generated reports

HTML : `reports/memcheck/<module>/index.html`

Bellow an example of generated html report :



Error summary

Leak_DefinitelyLost	1

Error details

+	tConfigParser	0
+	tApplication2	0
−	tApplication	1
	Command line	./tApplication --srcdir=/home/psyco/dev/xtdcpp/core --top-srcdir=/home/psyco/dev/xtdcpp --top-builddir=/home/psyco/dev/xtdcpp/.release --testdir=/home/psyco/dev/xtdcpp/core/unit --outputter=compiler -p -e 7
−	Leak_DefinitelyLost	100 bytes in 1 blocks are definitely lost in loss record 1 of 2
+	0x4C2E80F	/usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so
+	0x5085D9	/home/psyco/dev/xtdcpp/core/unit/TestApplication.cc:62
+	0x4ABDEF	/home/psyco/dev/xtdcpp/core/unit/TestApplication.cc:93
+	0x54A86F	/usr/include/cppunit/TestCaller.h:166
+	0x59C2291	/usr/lib/x86_64-linux-gnu/libcppunit-1.13.so.0.0.2
+	0x59B8B92	/usr/lib/x86_64-linux-gnu/libcppunit-1.13.so.0.0.2
+	0x59BF4C1	/usr/lib/x86_64-linux-gnu/libcppunit-1.13.so.0.0.2
+	0x59C7B2F	/usr/lib/x86_64-linux-gnu/libcppunit-1.13.so.0.0.2
+	0x59C206F	/usr/lib/x86_64-linux-gnu/libcppunit-1.13.so.0.0.2
+	0x59C25C2	/usr/lib/x86_64-linux-gnu/libcppunit-1.13.so.0.0.2
+	0x59C24DD	/usr/lib/x86_64-linux-gnu/libcppunit-1.13.so.0.0.2
+	0x59C25C2	/usr/lib/x86_64-linux-gnu/libcppunit-1.13.so.0.0.2

JSON : `reports/memcheck/<module>/memcheck.json`

```
{
   "tests" : [
      {
         "args" : {
            "args" : [
               "--srcdir=/home/psyco/dev/xtdcpp/core",
               "--top-srcdir=/home/psyco/dev/xtdcpp",
               "--top-builddir=/home/psyco/dev/xtdcpp/.release",
               "--testdir=/home/psyco/dev/xtdcpp/core/unit",
               "--outputter=compiler",
               "-p",
               "-e",
               "7"
            ],
```

(continues on next page)

(continued from previous page)


```
            "bin" : "./tApplication"
         },
         "errors" : [
            {
               "descr" : "100 bytes in 1 blocks are definitely lost in loss record 1␣
↪of 2",
               "kind" : "Leak_DefinitelyLost",
               "stack" : [
                  {
                     "line" : "",
                     "ip" : "0x4C2E80F",
                     "fn" : "operator new[](unsigned long)",
                     "obj" : "/usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so",
                     "file" : "",
                     "dir" : ""
                  },
                  {
                     "dir" : "/home/psyco/dev/xtdcpp/core/unit",
                     "obj" : "/home/psyco/dev/xtdcpp/.release/core/tApplication",
                     "file" : "TestApplication.cc",
                     "line" : "62",
                     "ip" : "0x5085D9",
                     "fn" : "MyApp::MyApp(bool)"
                  },
                  {
                     "obj" : "/home/psyco/dev/xtdcpp/.release/core/tApplication",
                     "file" : "TestApplication.cc",
                     "line" : "93",
                     "ip" : "0x4ABDEF",
                     "fn" : "TestApplication::handleSignal()",
                     "dir" : "/home/psyco/dev/xtdcpp/core/unit"
                  }
               ]
            }
         ]
      }
   ],
   "stats" : {
      "Leak_DefinitelyLost" : 1
   }
}
```


JSON : `reports/memcheck/<module>/status.json`


```
{
  "status": "failure",
  "graphs": [
    {
      "data": {
        "labels": [],
        "datasets": [
          {
            "borderColor": "rgba(179, 0, 0, 0.5)",
            "pointBorderColor": "rgba(102, 0, 0, 1)",
            "yAxisID": "absolute",
            "label": "memcheck error count",
            "backgroundColor": "rgba(179, 0, 0, 0.5)",
```


(continues on next page)

(continued from previous page)

```
            "pointBackgroundColor": "rgba(102, 0, 0, 1)",
            "data": "%(total)d"
          }
        ]
      },
      "type": "line",
      "options": {
        "scales": {
          "xAxes": [
            {
              "ticks": {
                "fontSize": 12,
                "minRotation": 80
              }
            }
          ],
          "yAxes": [
            {
              "position": "left",
              "ticks": {
                "fontSize": 24,
                "beginAtZero": true
              },
              "type": "linear",
              "id": "absolute",
              "display": true
            }
          ]
        },
        "title": {
          "text": "%(module)s : memcheck",
          "display": true
        }
      }
    }
  ],
  "data": {
    "total": 1
  },
  "label": "1"
}
```

4.8 CodeDupRule

- *Prerequisites*
- *Functions*
- *Parameters*
- *Global variables*
- *Suppression file*

This module generates a report that shows detected code duplication blocks.

4.8.1 Prerequisites

Java Java runtime environment. Available from ubuntu packages or from source at http://cppcheck.sourceforge.net/

> **Warning:** module requires Java 8 minimum version.

Pmd PMD is a source code analyzer. Available from source or binaries at http://pmd.sourceforge.net/

xsltproc XSL Template rendering tool. Available from ubuntu packages or from source at http://xmlsoft.org/

4.8.2 Functions

```
add_codedup(module,
  [INTPUT           <dir>     [ <dir>     ... ]],
  [FILE_PATTERNS    <pattern> [ <pattern> ... ]],
  [EXCLUDE_PATTERNS <regexp>  [ <regexp>  ... ]],
  [SUPPRESSIONS     <file>]
  [MIN_TOKENS       <int>]
  [ARGS             <int>]
)
```

This function generates cmake targets that produce codedup report for a given module. Generated targets are added as dependency of the global `codedup` and `codedup-clean` targets.

4.8.3 Parameters

module Name of the module. It determines the name of the generated cmake targets and the directory where targets generate the report.

INPUT List of directories where target should search source files process. Default value is given by *CodeDupRule_DEFAULT_INPUT*

FILE_PATTERNS List of wildcards search files in given input directories. Default value is given by *CodeDupRule_DEFAULT_FILE_PATTERNS*

EXCLUDE_PATTERNS List of regular expressions to exclude matched input files. Default value is given by *CodeDupRule_DEFAULT_EXCLUDE_PATTERNS*

SUPPRESSIONS Path to suppression list. Default value is given by *CodeDupRule_DEFAULT_SUPPRESSION*

4.8.4 Global variables

CodeDupRule_DEFAULT_PMD_VERSION
"5.7.0"

CodeDupRule PDM installed version.

CodeDupRule_DEFAULT_PMD_HOME
"/usr/share/pmd-bin-${CodeDupRule_PMD_VERSION}"

CodeDupRule location of PDM installation.

CodeDupRule_DEFAULT_INPUT
"${CMAKE_CURRENT_SOURCE_DIR}/src"

CodeDupRule default list of input source directories

CodeDupRule_DEFAULT_FILE_PATTERNS
"*.cc;*.hh;*.hxx"

CodeDupRule default list of wildcard patterns to search in INPUT directories

CodeDupRule_DEFAULT_EXCLUDE_PATTERNS
"${CMAKE_CURRENT_SOURCE_DIR}/unit/.*"

CodeDupRule default list of regexp to exclude from analysis

CodeDupRule_DEFAULT_MIN_TOKENS
"100"

CodeDupRule default minimum token length which should be reported as a duplicate

CodeDupRule_DEFAULT_ARGS
"--skip-lexical-errors"

CodeDupRule default additional arguments to give to PMD

CodeDupRule_DEFAULT_SUPPRESSION
"${CMAKE_CURRENT_SOURCE_DIR}/src/codedup.suppr"

CodeDupRule default path to suppression file

4.8.5 Suppression file

You may want to squelch some of the duplicated blocks detected by PMD. To do so can provide a `json` file with the following format:

```
[
  <suppression_1>,
  <suppression_2>,
  ...
]
```

where each `<suppression>` structure gives instruction to squelch one bloc with the following format:

```
[
  {
    "file" : "<path-to-file>",
    "from" : <start_line>,
    "to"   : <end_line>
  },
  {
    "file" : "<path-to-file>",
    "from" : <start_line>,
    "to"   : <end_line>
  },
```

(continues on next page)

(continued from previous page)

```
  ...
]
```

Duplicated code block detected by PMD is compared to each `<suppression>`. When bloc if found is all given `files` between `from` and `to` lines, the duplication is squelched.

4.8.6 Generated targets

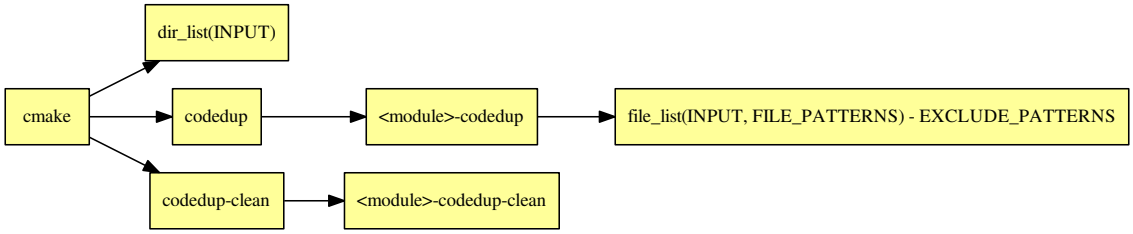
`codedup` generate codedup reports for all modules

`codedup-clean` removes codedup reports for all modules

`<module>-codedup` generate codedup report for module <*module*>

`<module>-codedup-clean` removes codedup report for module <*module*>

4.8.7 Dependencies



Warning: The dependency of cmake build system to the modification time of `INPUT` directories doesn't work with cmake versions prior to 3.0. This mean you must re-run cmake after adding new sources files in order to properly update the rule files dependencies

4.8.8 Generated reports

HTML : `reports/codedup/<module>/index.html`

Bellow an example of generated html report :

Summary report

Detected blocks	1
Duplicated lines	18

Full report

	Block ID	Number of lines	Number of tokens	Associated files
+	ID	18	121	2

File	Line
/home/psyco/dev/xtdcpp/core/src/log/ColoredFormatter.cc	16
/home/psyco/dev/xtdcpp/core/src/log/ColoredFormatter.cc	34

```
  Formatter()
{
  using namespace tty;

  setStyles({
      { "name",     style(color::green)              },
      { "threadid", style(color::yellow)             },
      { "message",  style(color::white)              },
      { "module",   style(color::lyellow)            },
      { "time",     style(color::cyan)               },
      { "slevel",   style(color::lred, attrs::bold)  },
      { "location", style(color::lblack)             },
      { "pid",      style(color::lblue)              },
      { "ppid",     style(color::lblue, attrs::bold) }
    });
}

ColoredFormatter::ColoredFormatter(const Formatter& p_base) :
```

XML : `reports/codedup/<module>/codedup.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<pmd-cpd>
   <duplication lines="18" tokens="121">
      <file line="16" path="/home/psyco/dev/xtdcpp/core/src/log/ColoredFormatter.cc"/>
      <file line="34" path="/home/psyco/dev/xtdcpp/core/src/log/ColoredFormatter.cc"/>
```

(continues on next page)

(continued from previous page)

```
      <codefragment><![CDATA[  Formatter()
{
  using namespace tty;

  setStyles({
      { "name",     style(color::green)               },
      { "threadid", style(color::yellow)              },
      { "message",  style(color::white)               },
      { "module",   style(color::lyellow)             },
      { "time",     style(color::cyan)                },
      { "slevel",   style(color::lred, attrs::bold)   },
      { "location", style(color::lblack)              },
      { "pid",      style(color::lblue)               },
      { "ppid",     style(color::lblue, attrs::bold)  }
    });
}

ColoredFormatter::ColoredFormatter(const Formatter& p_base) :]]></codefragment>
   </duplication>
</pmd-cpd>
```

JSON : `reports/codedup/<module>/status.json`

```
{
  "status": "failure",
  "index": "index.html",
  "module": "core",
  "label": "1",
  "graphs": [
    {
      "data": {
        "labels": [],
        "datasets": [
          {
            "borderColor": "rgba(179, 0, 0, 0.5)",
            "pointBorderColor": "rgba(102, 0, 0, 1)",
            "yAxisID": "absolute",
            "label": "codedup: # error count",
            "backgroundColor": "rgba(179, 0, 0, 0.5)",
            "pointBackgroundColor": "rgba(102, 0, 0, 1)",
            "data": "%(total)d"
          }
        ]
      },
      "type": "line",
      "options": {
        "scales": {
          "xAxes": [
            {
              "ticks": {
                "fontSize": 12,
                "minRotation": 80
              }
            }
          ],
          "yAxes": [
            {
```

(continues on next page)

(continued from previous page)

```
              "position": "left",
              "ticks": {
                "fontSize": 24,
                "beginAtZero": true
              },
              "type": "linear",
              "id": "absolute",
              "display": true
            }
          ]
        },
        "title": {
          "text": "%(module)s : codedup",
          "display": true
        }
      }
    }
  ],
  "kpi": "codedup",
  "data": {
    "total": 1
  }
}
```

4.9 IwyuRule

This module generates a report that shows detected code duplication blocks.

4.9.1 Prerequisites

include-what-you-use LLVM-based include analyzer. Available from ubuntu packages or from source at https://include-what-you-use.org/

Mako Python template renderer. Available from ubuntu packages or from source at http://www.makotemplates.org/

4.9.2 Functions

```
add_iwyu(module,
  DEPENDS target1 [target2 ... ],
  [EXCLUDE_PATTERN  <glob>],
  [JOBS             <int>],
  [MAPPING          <file>],
  [VERBOSE]
)
```

This function generates cmake targets that produce a report about includes dependencies for a given module. Generated targets are added as dependency of the global `iwyu` and `iwyu-clean` targets.

4.9.3 Parameters

module Name of the module. It determines the name of the generated cmake targets and the directory where targets generate the report.

INPUT List of directories where target should search source files process. Default value is given by *IwyuRule_DEFAULT_INPUT*

FILE_PATTERNS List of wildcards search files in given input directories. Default value is given by *IwyuRule_DEFAULT_FILE_PATTERNS*

EXCLUDE_PATTERNS List of regular expressions to exclude matched input files. Default value is given by *IwyuRule_DEFAULT_EXCLUDE_PATTERNS*

SUPPRESSIONS Path to suppression list. Default value is given by *IwyuRule_DEFAULT_SUPPRESSION*

4.9.4 Global variables

IwyuRule_DEFAULT_PMD_VERSION
"5.7.0"

IwyuRule PDM installed version.

IwyuRule_DEFAULT_PMD_HOME
"/usr/share/pmd-bin-${IwyuRule_PMD_VERSION}"

IwyuRule location of PDM installation.

IwyuRule_DEFAULT_INPUT
"${CMAKE_CURRENT_SOURCE_DIR}/src"

IwyuRule default list of input source directories

IwyuRule_DEFAULT_FILE_PATTERNS
"*.cc;*.hh;*.hxx"

IwyuRule default list of wildcard patterns to search in INPUT directories

IwyuRule_DEFAULT_EXCLUDE_PATTERNS
"${CMAKE_CURRENT_SOURCE_DIR}/unit/.*"

IwyuRule default list of regexp to exclude from analysis

IwyuRule_DEFAULT_MIN_TOKENS
"100"

IwyuRule default minimum token length which should be reported as a duplicate

IwyuRule_DEFAULT_ARGS
"--skip-lexical-errors"

IwyuRule default additional arguments to give to PMD

IwyuRule_DEFAULT_SUPPRESSION
"${CMAKE_CURRENT_SOURCE_DIR}/src/codedup.suppr"

IwyuRule default path to suppression file

4.9.5 Suppression file

You may want to squelch some of the duplicated blocks detected by PMD. To do so can provide a `json` file with the following format:

```
[
  <suppression_1>,
  <suppression_2>,
  ...
]
```

where each `<suppression>` structure gives instruction to squelch one bloc with the following format:

```
[
  {
    "file" : "<path-to-file>",
    "from" : <start_line>,
    "to"   : <end_line>
  },
  {
    "file" : "<path-to-file>",
    "from" : <start_line>,
    "to"   : <end_line>
  },
  ...
]
```

Duplicated code block detected by PMD is compared to each `<suppression>`. When bloc if found is all given `files` between `from` and `to` lines, the duplication is squelched.

4.9.6 Generated targets

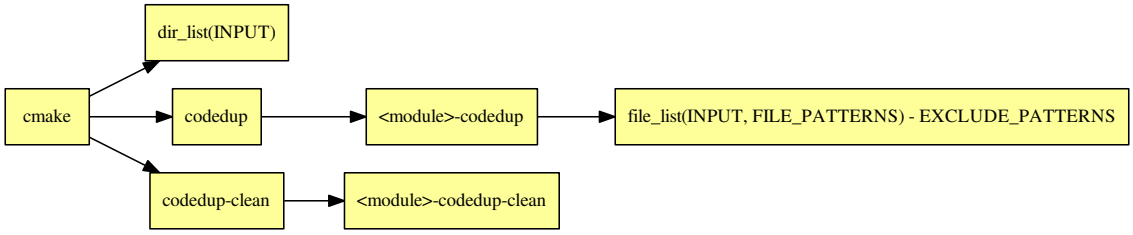
codedup generate codedup reports for all modules

codedup-clean removes codedup reports for all modules

<module>-codedup generate codedup report for module <*module*>

<module>-codedup-clean removes codedup report for module <*module*>

4.9.7 Dependencies



Warning: The dependency of cmake build system to the modification time of `INPUT` directories doesn't work with cmake versions prior to 3.0. This mean you must re-run cmake after adding new sources files in order to properly update the rule files dependencies

4.9.8 Generated reports

HTML : `reports/codedup/<module>/index.html`

Bellow an example of generated html report :

Summary report

Detected blocks	1
Duplicated lines	18

Full report

	Block ID	Number of lines	Number of tokens	Associated files
+	ID	18	121	2

File	Line
/home/psyco/dev/xtdcpp/core/src/log/ColoredFormatter.cc	16
/home/psyco/dev/xtdcpp/core/src/log/ColoredFormatter.cc	34

```
  Formatter()
{
  using namespace tty;

  setStyles({
      { "name",     style(color::green)              },
      { "threadid", style(color::yellow)             },
      { "message",  style(color::white)              },
      { "module",   style(color::lyellow)            },
      { "time",     style(color::cyan)               },
      { "slevel",   style(color::lred, attrs::bold)  },
      { "location", style(color::lblack)             },
      { "pid",      style(color::lblue)              },
      { "ppid",     style(color::lblue, attrs::bold) }
    });
}

ColoredFormatter::ColoredFormatter(const Formatter& p_base) :
```

XML : `reports/codedup/<module>/codedup.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<pmd-cpd>
   <duplication lines="18" tokens="121">
      <file line="16" path="/home/psyco/dev/xtdcpp/core/src/log/ColoredFormatter.cc"/>
      <file line="34" path="/home/psyco/dev/xtdcpp/core/src/log/ColoredFormatter.cc"/>
```

(continues on next page)

(continued from previous page)

```
      <codefragment><![CDATA[  Formatter()
{
  using namespace tty;

  setStyles({
      { "name",     style(color::green)               },
      { "threadid", style(color::yellow)              },
      { "message",  style(color::white)               },
      { "module",   style(color::lyellow)             },
      { "time",     style(color::cyan)                },
      { "slevel",   style(color::lred, attrs::bold)   },
      { "location", style(color::lblack)              },
      { "pid",      style(color::lblue)               },
      { "ppid",     style(color::lblue, attrs::bold)  }
    });
}

ColoredFormatter::ColoredFormatter(const Formatter& p_base) :]]></codefragment>
   </duplication>
</pmd-cpd>
```

JSON : `reports/codedup/<module>/status.json`

```
{
  "status": "failure",
  "index": "index.html",
  "module": "core",
  "label": "1",
  "graphs": [
    {
      "data": {
        "labels": [],
        "datasets": [
          {
            "borderColor": "rgba(179, 0, 0, 0.5)",
            "pointBorderColor": "rgba(102, 0, 0, 1)",
            "yAxisID": "absolute",
            "label": "codedup: # error count",
            "backgroundColor": "rgba(179, 0, 0, 0.5)",
            "pointBackgroundColor": "rgba(102, 0, 0, 1)",
            "data": "%(total)d"
          }
        ]
      },
      "type": "line",
      "options": {
        "scales": {
          "xAxes": [
            {
              "ticks": {
                "fontSize": 12,
                "minRotation": 80
              }
            }
          ],
          "yAxes": [
            {
```

(continues on next page)

(continued from previous page)


```
              "position": "left",
              "ticks": {
                "fontSize": 24,
                "beginAtZero": true
              },
              "type": "linear",
              "id": "absolute",
              "display": true
            }
          ]
        },
        "title": {
          "text": "%(module)s : codedup",
          "display": true
        }
      }
    }
  ],
  "kpi": "codedup",
  "data": {
    "total": 1
  }
}
```


4.10 Reports

This module will gather HTML reports generated by other XTDMake modules in a fancy HTML interface. This interface allows to navigates from report to report for all declared modules.

The generated html code is fully static, allowing user to view it directly in a web browser without any web server installed.

4.10.1 Prerequisites

Although there is no actual prerequisites to use this module, it's designed to work with other XTDMake's module that generates HTML reports. If none of them then are loaded, Report module will work but won't display any valuable information.

4.10.2 Generated Targets

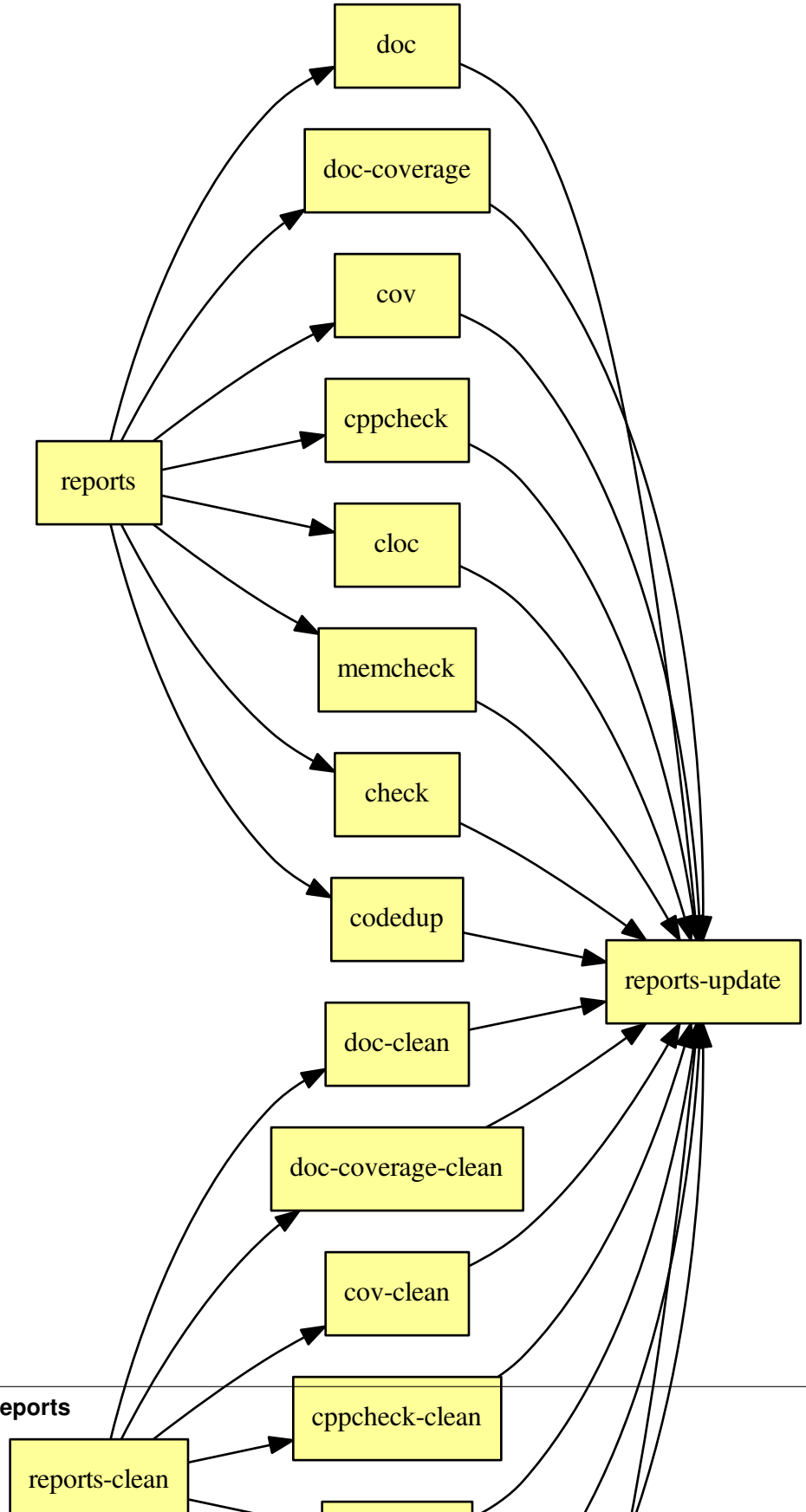
`reports` run all code quality targets for all modules

reports-clean clean all generated code quality reports

reports-update (internal use) update report static interface with available generated code quality targets

reports-show opens report interface in default web-browser (ie: sensible-browser)

4.10.3 Dependencies

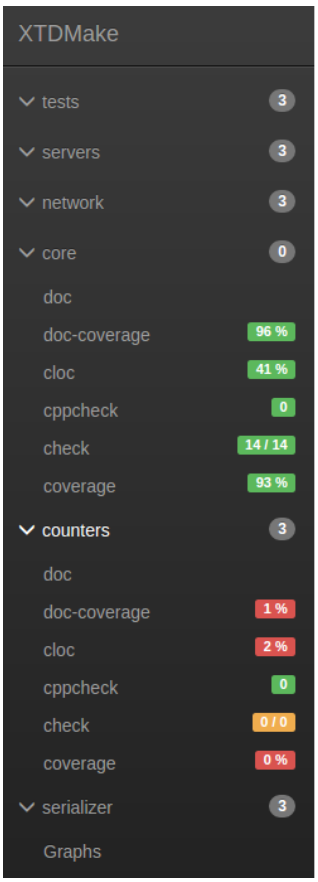


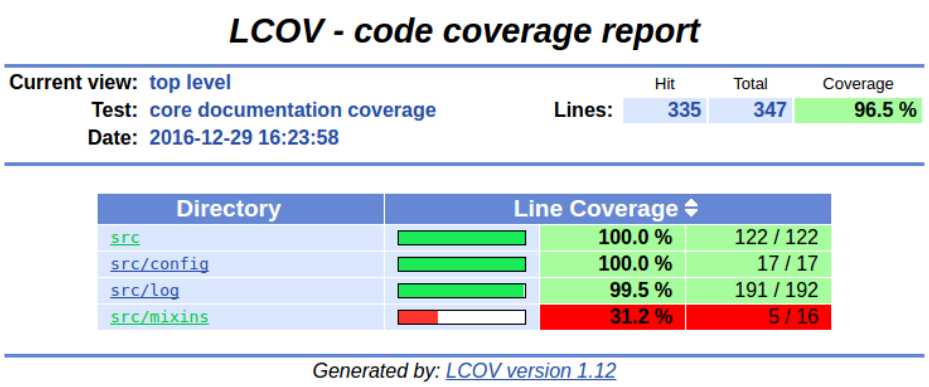
4.10.4 Generated interface

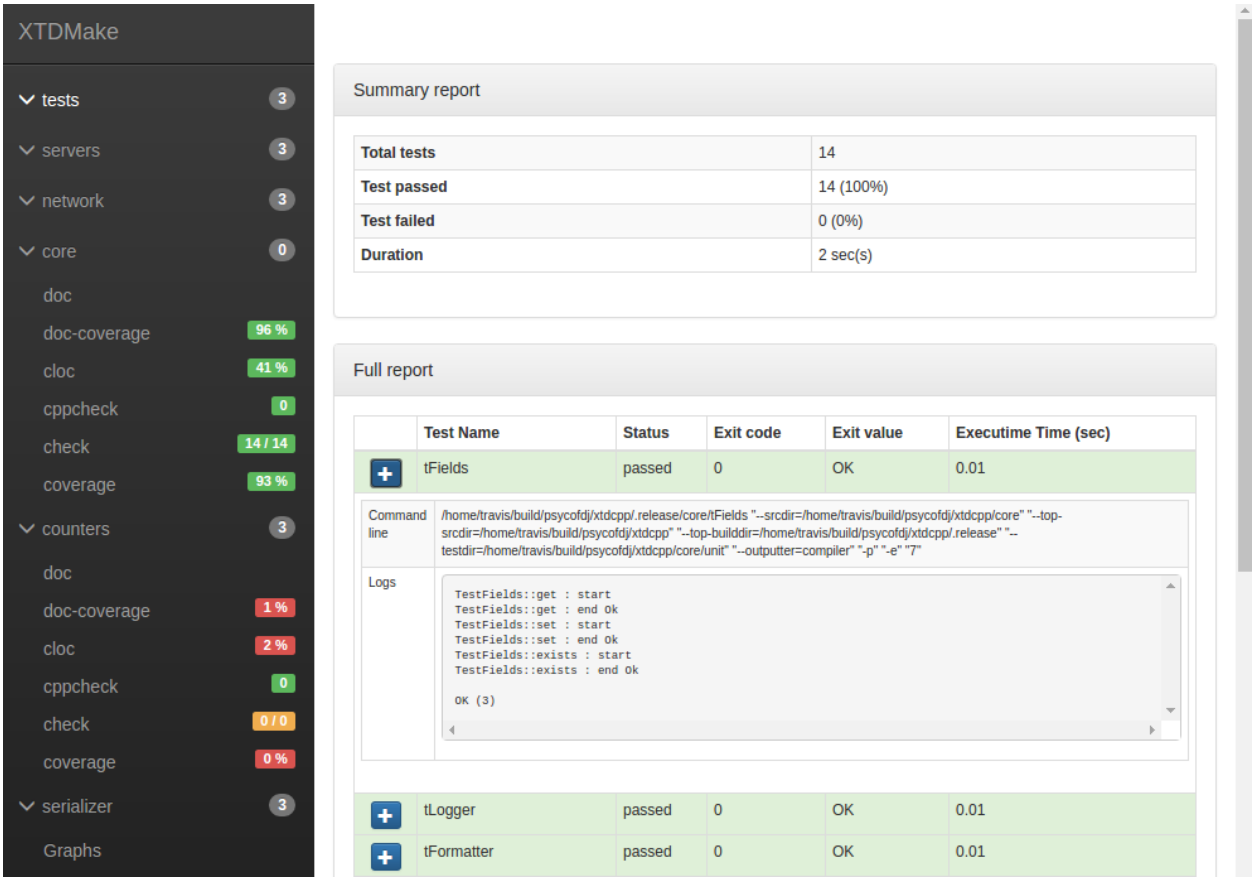
HTML : `reports/interface/index.html`

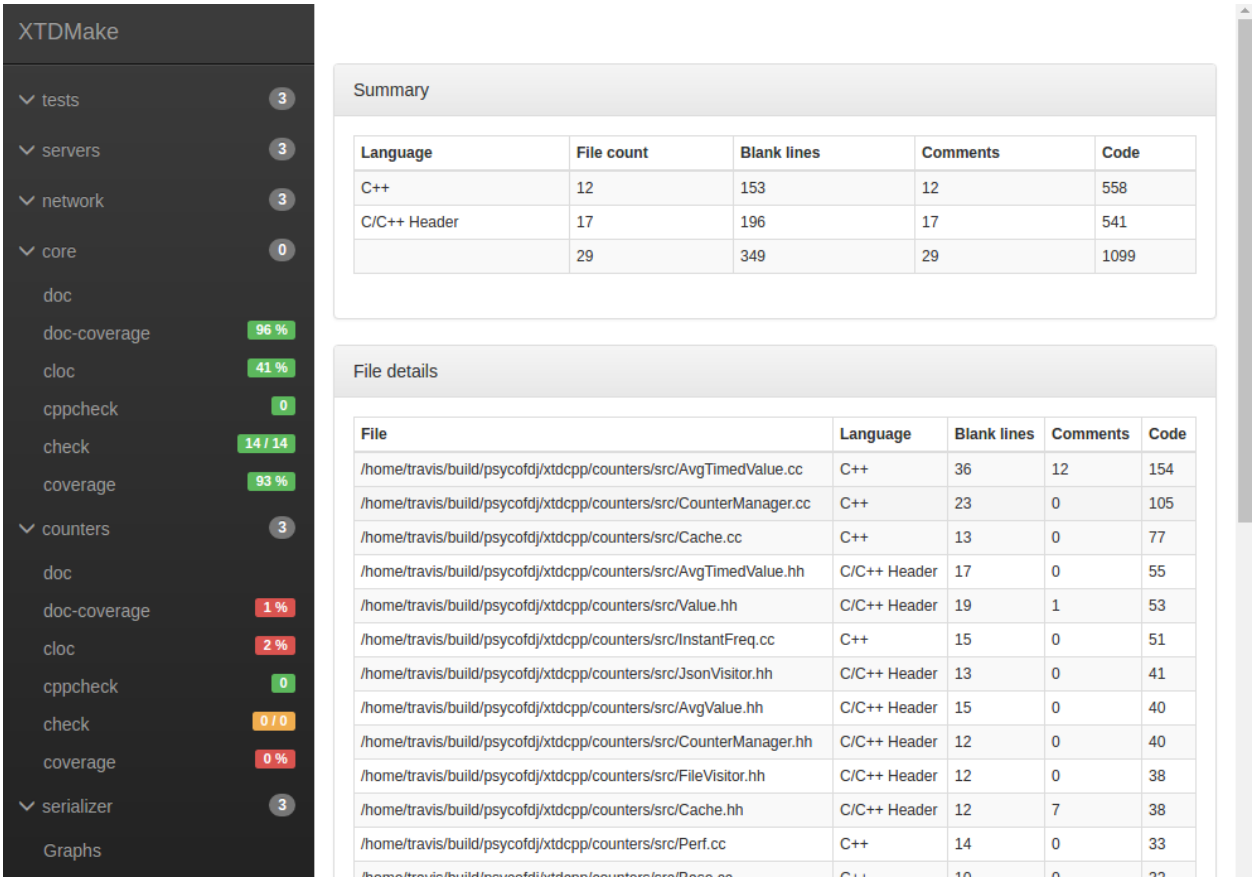
Try live example: https://psycofdj.github.io/xtdcpp/master/

Bellow few screnn shots :






XTDMake
tests 3
servers 3
network 3
core 0
doc
doc-coverage 96 %
cloc 41 %
cppcheck 0
check 14 / 14
coverage 93 %
counters 3
doc
doc-coverage 1 %
cloc 2 %
cppcheck 0
check 0 / 0
coverage 0 %
serializer 3
Graphs
Summary report
Total tests 14
Test passed 14 (100%)
Test failed 0 (0%)
Duration 2 sec(s)
Full report
Test Name Status Exit code Exit value Executime Time (sec)
tFields passed 0 OK 0.01
Command line /home/travis/build/psycofdj/xtdcpp/.release/core/tFields "--srcdir=/home/travis/build/psycofdj/xtdcpp/core" "--top-srcdir=/home/travis/build/psycofdj/xtdcpp" "--top-builddir=/home/travis/build/psycofdj/xtdcpp/.release" "--testdir=/home/travis/build/psycofdj/xtdcpp/core/unit" "--outputter=compiler" "-p" "-e" "7"
Logs
TestFields::get : start
TestFields::get : end Ok
TestFields::set : start
TestFields::set : end Ok
TestFields::exists : start
TestFields::exists : end Ok
OK (3)
tLogger passed 0 OK 0.01
tFormatter passed 0 OK 0.01



4.10.5 Graph history

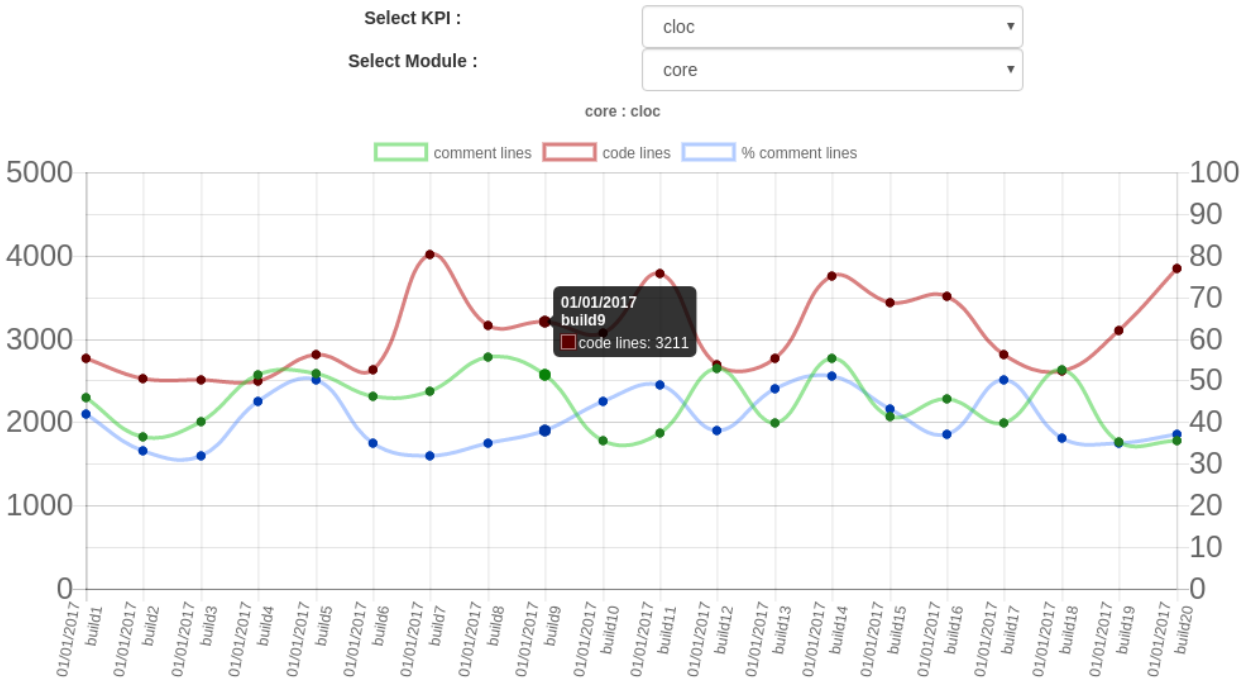
Report module also provides a graph generator tools that allow to keep track of the code quality measurements in time.

```
usage: graph [-h] --report-dir REPORT_DIR --history-dir HISTORY_DIR --output-dir␣
↪OUTPUT_DIR --build-label BUILD_LABEL [--max-items MAX_ITEMS] [--random]

optional arguments:
  -h, --help                show this help message and exit
  --report-dir REPORT_DIR   path to xtdmake reports
  --history-dir HISTORY_DIR path to history output
  --output-dir OUTPUT_DIR   path to javascript output
  --build-label BUILD_LABEL name of current build
  --max-items MAX_ITEMS     maximum number of build to keep in graph
  --random                  internal use
```

Note: This tool not run automatically by XTDMake since it has no way to know when to pin a new "release". It's designed to be run in your continuous integration process.

Example of generated graph :


Select KPI :
cloc
Select Module :
core
core : cloc
comment lines
code lines
% comment lines
5000
4000
3000
2000
1000
0
100
90
80
70
60
50
40
30
20
10
0
01/01/2017
build9
code lines: 3211
01/01/2017 build1
01/01/2017 build2
01/01/2017 build3
01/01/2017 build4
01/01/2017 build5
01/01/2017 build6
01/01/2017 build7
01/01/2017 build8
01/01/2017 build9
01/01/2017 build10
01/01/2017 build11
01/01/2017 build12
01/01/2017 build13
01/01/2017 build14
01/01/2017 build15
01/01/2017 build16
01/01/2017 build17
01/01/2017 build18
01/01/2017 build19
01/01/2017 build20

CHAPTER 5

Utility modules

5.1 StaticShared

This module provides an equivalent of cmake's `add_library` function that builds both static and shared libraries from the same set of object file which improves compilation time.

Warning: Objects are created with `-fPIC` flag which *may* lead to a loss of runtime performance when linking to static library.

5.1.1 Prerequisites

name and version The following variables must be defined :

- `PROJECT_NAME`
- `PROJECT_VERSION_MAJOR`
- `PROJECT_VERSION_MINOR`
- `PROJECT_VERSION_PATH`

cmake This module doesn't work properly with cmake version prior to 3.0. However this module is still compatible with such versions but will create two separate sets of objects for static and shared libraries.

5.1.2 Functions

```
add_shared_static_library(<libname>
  <source> [ <source> ... ]
  [ INSTALL_HEADERS_PATTERNS  <pattern>  [<pattern>  ...]]
  [ INSTALL_HEADERS_DESTINATION <path> ]
  [ INSTALL_LIBS_DESTINATION    <path> ]
  [ INSTALL_HEADERS_DIRECTORY   <dir>  ]
  [ VERSION <version>   ]
  [ SOVERSION <version> ]
  [ NOINSTALL ]
)
```

5.1.3 Parameters

libname Internal name of target libraries. At install time, files will be respectively named `lib${PROJECT_NAME}<name>.so` and `lib${PROJECT_NAME}<name>.a`.

source List of source file to build in libraries.

INSTALL_HEADERS_PATTERNS List of glob pattern to match headers file to install with target libraries.

Default value is given by *StaticShared_DEFAULT_INSTALL_HEADERS_PATTERNS*.

INSTALL_HEADERS_DIRECTORY Directory containing headers to install with target libraries.

Default value is given by `StaticShared_DEFAULT_INSTALL_HEADERS_DIRECTORY`.

INSTALL_HEADERS_DESTINATION Headers target install directory.

Default value is given by *StaticShared_DEFAULT_INSTALL_HEADERS_DESTINATION*.

INSTALL_LIBS_DESTINATION Libraries target install directory

Default value is given by *StaticShared_DEFAULT_INSTALL_LIBS_DESTINATION*.

VERSION Shared library version given to cmake `VERSION` property

SOVERSION Shared library version given to cmake `SOVERSION` property.

NOINSTALL Disables installation configuration for current libraries

5.1.4 Global variables

StaticShared_DEFAULT_INSTALL_LIBS_DESTINATION
"lib"

StaticShared_DEFAULT_INSTALL_HEADERS_DESTINATION
"include/${PROJECT_NAME}/${name}"

StaticShared_DEFAULT_INSTALL_HEADERS_PATTERNS
"*.h;*.hxx;*.hh;*.hpp"

StaticShared_DEFAULT_DIRECTORY
"src/"

StaticShared_DEFAULT_DEFAULT_VERSION
"${PROJECT_VERSION_MAJOR}.${PROJECT_VERSION_MINOR}.${PROJECT_VERSION_PATCH}"

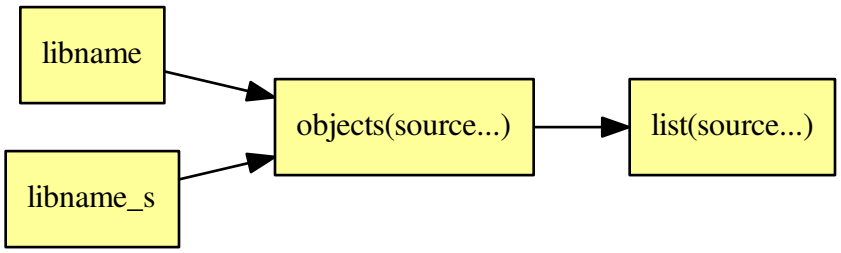
StaticShared_DEFAULT_DEFAULT_SOVERSION
"${PROJECT_VERSION_MAJOR}"

5.1.5 Generated targets

<libname> Target shared library

<libname>_s Target static library

5.1.6 Dependencies



5.2 Tracking

This module module wraps `C` and `CXX` linkers to embed RSC keywords string in your binaries and libraries. RSC keywords ran be later read using the `ident` command from `rcs` Ubuntu package.

Information included for libraries :

$date compile date of binary

$time compile time of binary

$revno current git or bzr revision if any

Information included for binaries :

$date compile date of binary

$time compile time of binary

$name target name of binary

$user shell user used for compilation

$pwd compile build directory

$revno current git or bzr revision if any

$archive [lib_name] (data) compile date of *lib_name* [lib_name] (time) compile time of *lib_name* [lib_name] (revno) git or bzr revno of *lib_name* if any

5.2.1 Functions

```
enable_tracking()
```

You must call this function on top level CMakeLists.txt after loading the Tracking module to enable tracking on your libraries and binaries.

5.2.2 Example

Given a binary `tAppender` compiled with static libraries `libxtdcore_s` and `libxtdtests_s` :

```
$ ident tAppender

$date: 01-01-2017 $
$time: 15:18:03 $
$name: tAppender $
$user: psyco $
$host: psyco-laptop-tux $
$pwd: /home/psyco/dev/xtdcpp/.release/core $
$revno: 9422c4460c24c7e0289f1d4ff0525e14ccabaedb $
$archive: [libxtdcore_s] (time) 15:17:33 $
$archive: [libxtdcore_s] (date) 01-01-2017 $
$archive: [libxtdcore_s] (revno) 9422c4460c24c7e0289f1d4ff0525e14ccabaedb $
$archive: [libxtdtests_s] (time) 15:14:06 $
$archive: [libxtdtests_s] (date) 01-01-2017 $
$archive: [libxtdtests_s] (revno) 9422c4460c24c7e0289f1d4ff0525e14ccabaedb $
```

5.2.3 How is works

Tracking module wraps C and C++ default linker and archive commands with `link_wrapper` and `ar_wrapper` scripts.

`ar_wrapper` silently adds a `.version` file when creating archives. Archives are sort of tars of object files, adding a file to the archive is not harmful.

`link_wrapper` does 3 things. First it searches for `.version` files on linked static archives and adds their content to the list. After gathering all possible information, it silently adds a source file to default link command. This source file declares `char rscid[] = __RCSID__`. Finally, the wrapper adds a `-D__RCSID__=` to linker command that defines the value of rcs keyword.

CHAPTER 6

Other functions

6.1 xtdmake_eval

```
xtdmake_eval(var expr)
```

Evaluates cmake expression `expr` and store it in `var`.

expr cmake expression to evaluate. Example: "${CMAKE_CURRENT_SOURCE_DIR}/toto"

var output variable

6.2 xtdmake_get_directory

```
xtdmake_get_directory(out in)
```

This function extract directory of path given as `in` and stores it in `out` variable. This function is compatible with both cmake (< 3.0) and cmake (>= 3.x).

in input file path

out destination variable

6.3 xtdmake_stringify

```
xtdmake_stringify(var)
```

Transform cmake list is a space-separated string

var input list

6.4 xtdmake_find_program

```
xtdmake_find_program(ns
  NAMES <name> [<name> ...]
  DOC   <string>
  URL   <string>
  REQUIRED <bool>
  [ VERSION_OPT <options> ]
  [ VERSION_POS <int>     ]
  [ MIN_VERSION <version> ]
)
```

Search program matching one of given `NAMES`, try to extract its version using `VERSION_OPT` and `VERSION_POS`, prints a message with STATUS or SEND_ERROR flag depending on `REQUIRED` option value.

Searching results are stores in variables prefixed by namespace `ns` :

- **`<ns>_EXECUTABLE`** name of executable file found among given names
- **`<ns>_FOUND`** 1 if program was found, 0 otherwise
- **`<ns>_VERSION`** version of found program, *unknown* if couldn't find any

ns namespace to store result variables

NAMES possible names of searched program

DOC brief description of searched program, displayed in status message when program is not found

URL url where searched program can be downloaded, displayed in status message when program is not found

REQUIRED when true and program is not found, status message is replace by an error

VERSION_OPT parameter string to pass to program to get its version on stdout, usually `--version`

VERSION_POS position of the version number in the space-delimited string outputted by program with `VERSION_OPT`

MIN_VERSION minimum allowed version of searched program

Example

```
xtdmake_find_program(cloc
  NAMES cloc
  DOC "cloc code line counting tool"
  URL "http://cloc.sourceforge.net/"
  VERSION_OPT "--version"
  VERSION_POS "0"
  MIN_VERSION 1.2
  REQUIRED 0)
```

(continues on next page)

(continued from previous page)

```
if (cloc_FOUND)
  message("cloc executable is ${cloc_EXECUTABLE}")
  message("cloc version ${cloc_VERSION}")
else()
  message("cloc is not available")
endif()
```

6.5 xtdmake_find_python_module

```
xtdmake_find_python_module(ns
 INTERPRETERS <pythonX> [ <pythonX> ... ]
 NAME <name>
 DOC  <string>
 URL  <string>
 REQUIRED <bool>
 VERSION_MEMBER <string>
 VERSION_POS    <string>
)
```

Search python module `NAME` trying given `INTERPRETERS`, try to extract its version using `VERSION_MEMBER` and `VERSION_POS`, prints a message with STATUS or SEND_ERROR flag depending on `REQUIRED` option value.

Searching results are stores in variables prefixed by namespace `ns` :

- **`<ns>_FOUND`** 1 if program was found, 0 otherwise
- **`<ns>_INTERPRETER`** python interpreter where module was found
- **`<ns>_VERSION`** version of found program, *unknown* if couldn't find any
- **`<ns>_NAME`** name of python module

ns namespace to store result variables

INTERPRETERS list of python interpreters to try to find module

NAMES name of python module to load

DOC brief description of searched module, displayed in status message when program is not found

URL url where searched module can be downloaded, displayed in status message when program is not found

REQUIRED when true and program is not found, status message is replace by an error

VERSION_MEMBER module member where version can be parsed, usually `__version__`

VERSION_POS position of the version number in the space-delimited string parsed in version member with `VERSION_MEMBER`

Example

```
xtdmake_find_python_module(coverxygen
  NAME coverxygen
  INTERPRETERS python3 python
  DOC "Tool to generate coverage report from Doxygen documentation"
  URL "https://github.com/psycofdj/coverxygen"
  REQUIRED DocCoverageRule_FIND_REQUIRED
  VERSION_MEMBER "__version__"
  VERSION_POS 0)
```

(continues on next page)

(continued from previous page)

```
if (coverxygen_FOUND)
  message("coverxygen was found using interpreter ${coverxygen_INTERPRETER}")
  message("coverxygen version is ${coverxygen_VERSION}")
  message("coverxygen can be run by the following command : ${coverxygen_INTERPRETER}
↪ -m ${coverxygen_MODULE} <args>")
else()
  message("coverxygen module was not found")
endif()
```

CHAPTER 7

Indices and tables

- genindex

Index