Verifysoft TECHNOLOGY

Software Testing Solutions for your Productivity and Quality

- Code Coverage
- Software Complexity Measurement
- Static Code Analysis
- Dynamic Code Analysis
- Safety-critical Embedded

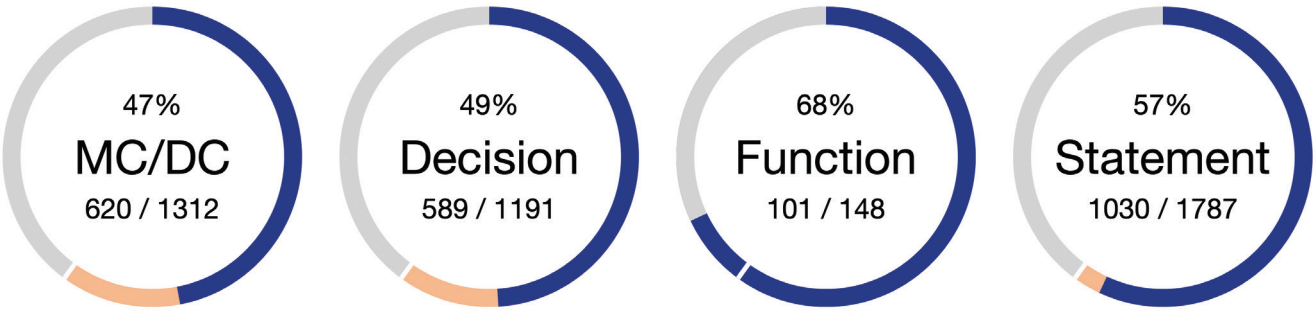
Testwell CTC++ Code Coverage Analyser

Code coverage for the highest requirements of safety standards

Testwell CTC++ analyzes which parts of your source code have been tested. Testwell CTC++ supports all coverage levels and is used by leading companies for safety-critical projects.

Coverage levels

Testwell CTC++ provides all coverage levels required by standards for safety-critical software development: function coverage, statement coverage, decision or equivalently branch coverage and modified condition/decision coverage (MC/DC). Condition and multicondition coverage can also be determined.

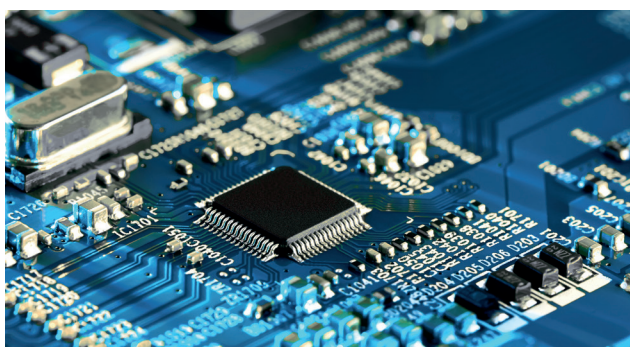


Desktop applications

- ✓ Low impact on the build process
- ✓ Scalable for large projects
- ✓ Compiler-independent
- ✓ For Windows, Linux, macOS

Embedded software

- ✓ Low memory requirements
- ✓ Testing on any target
- ✓ With any cross compiler
- ✓ Customizable runtime layer



Easy to use

- ✓ Generic build integration
- ✓ Very fast execution
- ✓ Seamless integration into many IDEs
- ✓ Ease of integration by modular architecture

Programming languages

- ✓ C, C++
- ✓ Add-Ons for Java and C#

Working method

Coverage measurement is performed with Testwell CTC++ in three independent phases:

Instrumentation

During compilation, Testwell CTC++ automatically instruments a copy of the source code by injecting measurement code. This creates an instrumented version of the program or test executable – fully automatically during the build process or on the basis of a simple, one-time build configuration.

Testing

Any type of tests can be executed as usual: Unit tests, integration tests or complete system tests. The coverage measurement data are written to a file. When performing tests on a target, this write-out is fully adjustable, e.g. the data can be transferred directly to the host computer.

Generating reports

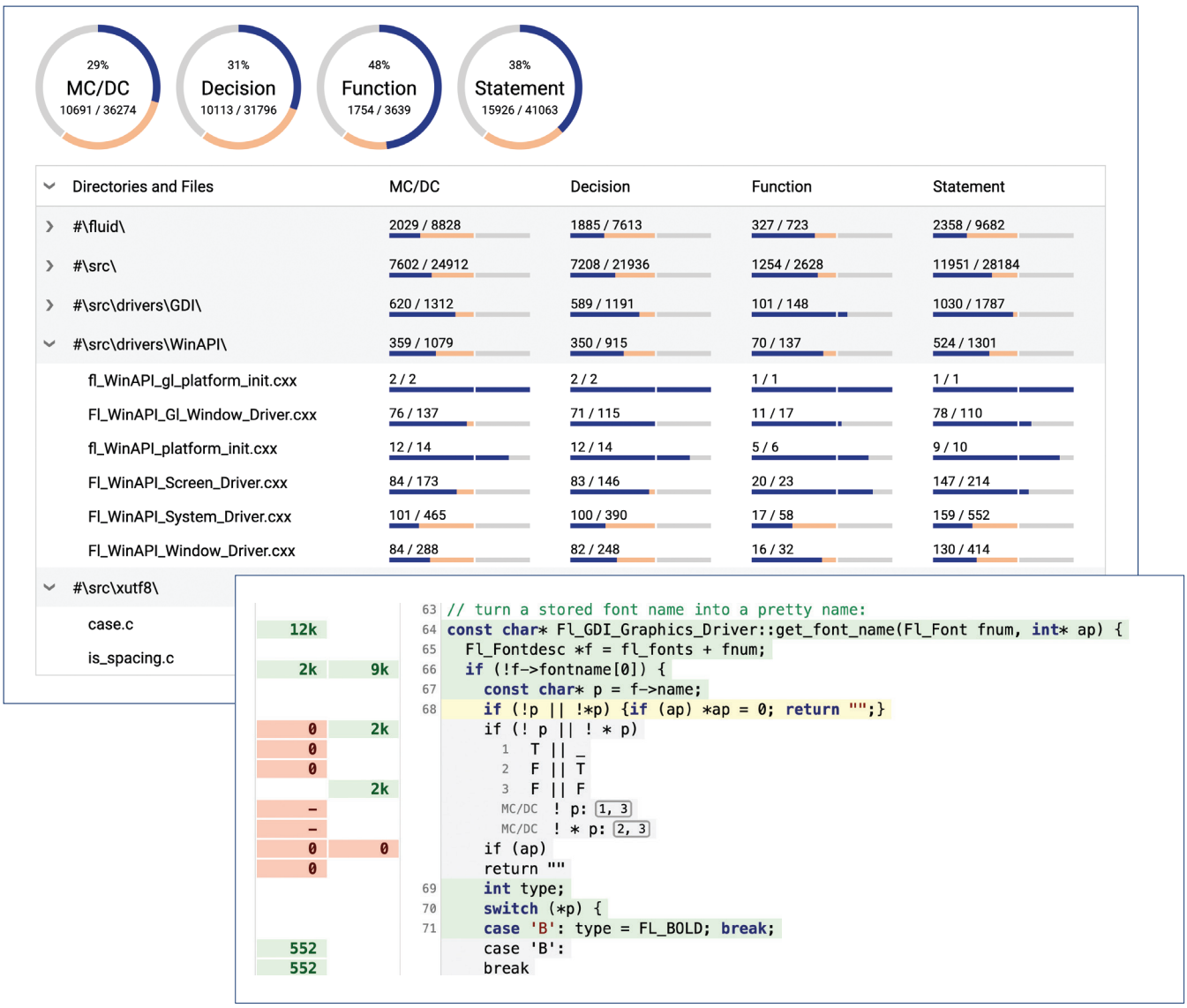
In the third phase, Testwell CTC++ generates coverage reports based on the raw data. Data from different builds and different tests can be combined. A structured HTML report and any text-based exchange formats are available as output formats.

Functional safety

- ✓ Suitable for safety-critical development according to:
  - ✓ ISO 26262
  - ✓ DO 178-C
  - ✓ EN 50657 / EN 50128
  - ✓ IEC 61508
  - ✓ IEC 62304
  - ✓ IEC 60880
  - ✓ ISO 25119 / DIN EN 16590
- ✓ Qualification support
- ✓ TÜV-certified
  - ✓ ISO 26262
  - ✓ IEC 61508
  - ✓ EN 50128
  - ✓ IEC 62304

Coverage reports

Testwell CTC++ provides a comprehensive HTML report that is adaptable to the user's needs and to the type and size of the project.



Configurable report layout

- ✓ Desired coverage levels in any combination
- ✓ Selectable report levels with drill-down: directories, source files, functions

Optional source code view

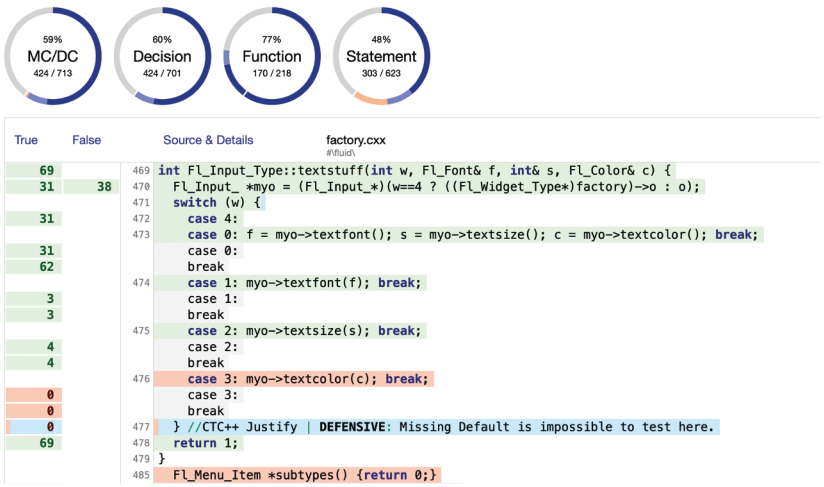
- ✓ Highlighting of executed and not executed lines
- ✓ Display of all coverage counters
- ✓ Compact visualization of complex coverage measures like MC/DC
- ✓ Visibility of missing test cases

Justification of missing coverage

Justifications can be used to record the reasons when full coverage cannot be achieved.

Testwell CTC++ derives which code parts are covered by a justification.

- ✓ Own categorization of justifications
- ✓ Recording in comments or in companion files
- ✓ Clear distinction between tested and justified code
- ✓ Recognition of over-justification



Coverage data in any form

Create template-based reports in any form. With a simple template language for data export, Testwell CTC++ supports structured reports such as HTML reports as well as the export of single text files.

	A	B	C	D
1		MC/DC		
2	Function	Hits	Total	Ratio
3	hasAdmission	6	8	75%
4	calcPrice	9	10	90%
5	main	6	6	100%
6	TicketApp::showInstruction	3	6	50%
7	TicketApp::switch2BatchMode	2	2	100%
8	TicketApp::ask4Input	3	4	75%
9	TicketApp::reportPrice	3	4	75%

Classic exchange formats like CSV, XML, JSON

Overall result as a badge or on dashboards

```
# Coverage Report: Coaster as Markdown

This report was generated at 2024-03-04 09:44:08 using
    `ctcreport -template example_markdown -o coverage.md`

## Coverage in Total
- 54% MC/DC (48 / 88)
- 85% Statement (60 / 70)

## Coverage per Source File
### Directory C:\CoasterCode\
#### PassengerScan.cpp:
- 75% MC/DC (6 / 8)
- 100% Statement (3 / 3)

#### PriceCalculation.cpp:
- 90% MC/DC (9 / 10)
- 100% Statement (5 / 5)
```

Text reports, e.g. in Markdown, for easy archiving and management in a repository