

03 December 2021



Change Documentation for

***Testwell CTC++***

Version 9.1.3

## Features and Changes

### Determination of data file name for test execution

With the new environment variable `CTC_DATA_NAME`, the name of the data file written out during execution of an instrumented program can be set. This name substitutes the name set by default ("**MON.dat**") or at instrumentation time with the parameter `DATAFILE`.

With that possibility, it is easier to distinguish different test runs with the same executable.

A differing path for the data file is still determined by the environment variable `CTC_DATA_PATH`.

Example (Windows-oriented): Unit tests and integration tests share an instrumented library

```
set CTC_DATA_NAME=UnitTests.dat
... execute unit tests ...
set CTC_DATA_NAME=IntegrationTests.dat
... execute integration tests ...
```

### HOTA and BITCOV package

On Linux, HOTA / BITCOV is now also part of the standard installation procedure (like already on Windows). The documentation of BITCOV is moved to the Testwell CTC++ Help.

For Linux, the improved error message of **dmp2txt** (see release 9.1.2) is deployed with this update.

### Cygwin Integration

With the installation on Windows, files for Cygwin integration are copied to the Testwell CTC++ directory in folder `\Cygwin_integ`. There is a makefile to perform the actual installation of this integration.

This package has been improved to work better regarding installation, configuration, and parallel usage with gcc compiler from MinGW.

## Bug Fixes

### Recognition of types

User-defined types introduced with `using namespace ns_name;` or `using ns_name::name;` were not always recognized correctly by `ctc` during instrumentation. This led to different issues:

- Declarations inside `if` statements were not recognized.
- Static casts like `static_cast<string&&>` were not handled correctly.

In both cases, the instrumentation resulted in uncompileable code.

### Nested `< >` brackets in static casts

A static cast with many nested angular brackets could lead to a syntax error of `ctc` for the end of the source file to be instrumented.

### Crash of `ctc` with long source code paths

Paths with more than 260 characters could cause a crash of `ctc`. This is fixed by cutting the beginning of the path.

## Header presentation in HTML report

A special situation could lead to a wrong presentation in the source code view of the HTML report:

- several instrumented header files are included in a source file,
- one of them is reported as a part of its including source file by **ctcpost** because there was a different variant of the header file extracted before.

To fix the presentation, the line numbers associated by **ctc** with the `#include` directives are changed and harmonized. Hence a full rebuild with a new symbol file **MON.sym** is necessary for this change to take effect for a project.

## Recognition of constant expressions in C++

For C++ code, non-constant expressions in if-statements like

```
if (A & 1);
```

were wrongly recognized as constant, if they started with one non-constant operand followed by certain operators and several constant operands.