



Tampere (Finland) / Offenburg (Germany), 9 March 2012

Please be informed that a new **CTC++ version 7.0.2** has been released.

CTC++ v7.0.2 is a small bug-fix version with some enhancements. Mentioning here some of them:

- new NO\_INCLUDE conf. parameter
- now also wildcards supported in SKIP\_FUNCTION\_NAME conf. parameter
- some bug fixes, also little changes, in statement coverage handling
- new C++ 11 string and character literals
- small HTML report changes, now better "understandable"

The new version is available on all supported host platforms. See below the v7.0.2 for more detailed info.

Version 7.0.2 (29 February 2012)

-----  
This revision 7.0.2 of CTC++ has the following version numbers in its components:

Preprocessor	7.0.2	(was 7.0.1; seen by -h option)
Run-time libraries	7.0.2	(was 7.0; seen by 'ident' command applied on the library in some environments)
Postprocessor	7.0.2	(was 7.0.1; seen by -h option and in the listings)
Header file ctc.h	7.0	(unchanged; seen in the file)
Configuration file ctc.ini	7.0.2	(was 7.0; seen in the file)
CTC++ to HTML Converter	3.1	(was 3.0; seen by -h option)
CTC++ to Excel Converter	2.0	(unchanged; seen by -h option)
CTC++ Merger utility	2.0	(unchanged; seen by -H option and in the merged listings)
ctc2dat receiver utility	3.0	(unchanged; seen by -h option)

and the following version numbers in its Windows platform specific components:

Visual Studio 2003-2010 IDE Integration	4.0	(unchanged; seen by clicking the Tw-icon in the dialog program and selecting "About...")
CTC++ Wrapper for Windows	3.0	(unchanged; seen by -h option)

and the following version numbers in its Unix platform (Linux, Solaris, HPUX) specific components:

CTC++ Wrapper for Unix	1.3	(unchanged; seen by -h option)
------------------------	-----	--------------------------------



Testwell CTC++ Version 7.0.2 - page 2

This CTC++ v7.0.2 version contains small enhancements and bug-fixes:

In the CTC++ preprocessor (ctc):

- New: Introduced NO\_INCLUDE configuration parameter. Now with triple EXCLUDE, NO\_EXCLUDE, NO\_INCLUDE the following use convention comes possible for selecting the files to be instrumented: First "EXCLUDE=\*", then "NO\_EXCLUDE=some dirs and files", and finally "NO\_INCLUDE=%INCLUDES%,and these dirs and files". Current use convention, where NO\_INCLUDE is empty, works as before.
- New: Now in SKIP\_FUNCTION\_NAME argument list wildcards can be used.
- New: In the optional/additional instrumentation phases in scripts RUN\_BEFORE\_ALL, RUN\_AFTER\_CPP, RUN\_AFTER\_INSTR and RUN\_AFTER\_COMP the used instrumentation options can be read from environment variable CTCOPTS. [Came in v6.5.6, but was forgotten to mention then]
- New: Now correctly parsing new C++11 standard character literals that are prefixed with u, U, e.g. U'x'. (prefix L was handled already)
- New/changed: Now correctly parsing new C++11 standard string literals that are prefixed with u8, u, U, R, uR, u8R, UR, LR, e.g. U"abc". (plain L and S prefixed string literals were handled already).
- Change: When a multi-line string literal, e.g. "abc\ndefg" needs to be stored in the symbolfile (MON.sym), it is there as a one-liner like "abc\ndefg" (previously "abcdefg").
- Change: If #pragma CTC SKIP/ENDSKIP have been used inside a function, a warning of it is given as before, but now the the function is labeled as "no statement coverage available".
- Change: Control-C disabling is shortened to the time when the symbolfile (MON.sym) is actually being accessed. Previously Control-C termination was disabled effectively all the time, which prevented aborting of a long run (when many files to instrument).
- Change: It is now a warning if the issued "double compilation" command fails. Instrumentation is tried anyway. Previously this terminated the ctc run.
- Bug fix: Fixed a bug if in the connection of RUN\_BEFORE\_ALL some later phase failed. Occurred only at Linux, ctc crashed.
- Bug fix: Now \_\_pragma(...) is recognized properly in function profile after parameter list. Previously the function remained uninstrumented.
- Bug fix: Fixed a bug when constructs like 'if (v<int>\* = ...) {...}' were instrumented erroneously (non-compilable code), if -no-templates option was used.

In the CTC++ postprocessor (ctcpost):

- Bug fix: Corrected statement coverage reporting in a couple of cases, for example sometimes statements after 'try{...}catch(...){...}...' could remain uncounted even though they were executed.
- Change: If at instrumentation time a function was labeled as "no statement coverage available", it is reported as 0/0 (as if 0 statement executed of total 0 statements). Because all control paths are not known, statement coverage would be grossly biased anyway.

In CTC++ to HTML converter (ctc2html):

- New: Made the summary level HTML pages more clear by adding to the structural TER% and histogram columns a header text "MC/DC", "multicondition", "decision", etc. telling what coverage type the numbers and the histogram stand for.
- New: Support for UTF-8 encoding added. This means better HTML code generation when Japanese, Chinese, Cyrillic, euro sign, "A a O o with dots", etc. category of characters.

General:

- CTC++ User's Guide upgraded to v7.0.2 level (ctcug.pdf).

For further questions, please call +49 781 127 8118-0

Version 7.0 (29 August 2011)

=====

For this version, please have a look to  
<http://www.verifysoft.com/ctcpp70.pdf>