

Run-time libraries	8.2.2	(was: 8.2; seen by 'ident' command applied on the library in some environments)
Postprocessor	8.2.2	(was: 8.2; seen by -h option and in the listings)
Header file ctc.h	8.2.1	(was: 8.2; seen in the file)
Configuration file ctc.ini	8.2.2	(was: 8.2; seen in the file)
CTC++ Coverage Comparator	1.0	(new; seen by -h option)
CTC++ to HTML Converter	5.5	(was: 5.4; seen by -h option)
CTC++ to Excel Converter	3.3	(unchanged: seen by -h option)
CTC++ XML Merger utility	3.4	(was: 3.3; seen by -h option)
ctc2dat receiver utility	3.7	(was: 3.6; seen by -h option)

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

Visual Studio IDE Integration	4.3	(was: 4.2; seen by clicking the Tw-icon in the dialog program and selecting "About...")
-------------------------------	-----	---

CTC++ Wrapper for Windows	3.5	(unchanged: seen by -h option)
---------------------------	-----	--------------------------------

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

CTC++ Wrapper for Unix 1.4 (unchanged; seen by -h option)

In the CTC++ preprocessor (ctc):

- Bug fix: When an explicit cast was inside a multicondition, then it wasn't instrumented under all circumstances, e.g. this
if ((int) (A && B))
Now it gets instrumented appropriately.
- Enhancement: Added a generic warning message "Cannot instrument ..."
for some tool limitations. Currently only used if unable to figure out that e.g. (Cast) is a cast or a function name Cast.
- Enhancement: The basename of the file that is instrumented is now encoded into the temporary file name, e.g.
CTC.1234.2.c => CTC.1234.2.calc.c
When those file names were recycled in some use scenarios linking failed (-k option had to be used). New style is also more documentary.
- Workaround: (Windows) In ctc.ini MAX_CMDLINE_LENGTH is set to 8000. Although technical limit is 8191, ctc needs some command line space for OPT_ADD_COMPILE content.

- Enhancement: In verbose mode (-v option) instrumentation, all warnings are now displayed without checking configuration parameter WARNING_LEVEL value.
- Enhancement: A const assignment expression, like 'const int i = expr;', no longer issues a warning, unless the expr has '&&' or '||' operator, and the instrumentation is for multicondition.
- Bug fix: ctc no longer puts -C options into environment variable CTCOPTS. There are use scenarios where the environment variable capacity is not enough, e.g. when generated -C EXCLUDE/NO_EXCLUDE file selections are used. Now these options are internally handled by ctc's response file (@ctcopts.rsp).
- Bug fix: Fixed time-stamp handling in native 64-bit builds.
- Enhancement: On Unix it is now accepted '\' as path separator. E.g. if '\' appears in a #line directive.
- Enhancement: In parallel use the time slice where ctc waits for exclusive access to symbolfile (MON.sym.lock file) has been lowered from 100ms to 25ms. Better throughput is expected.
- Bug fix: Warnings at info level are now displayed in the way as the User's Guide tells, as "CTC++ warning". Previously they were displayed as "CTC++ info".
- Enhancement: Supports now OpenMP. Previously the whole file could not be instrumented, if openmp switch was used. Now such files can be instrumented and ctc detects the for(...) loops that are subject to OpenMP behavior. The loop control part is not be instrumented, instead a probe is put to the begin of the loop body.
- Bug fix: Ternary operators inside angle brackets of member function template c
all, e.g.
a.template foo<iM==iN?3:4>();
are not longer instrumented.
- Enhancement: Various Cygwin build enhancements.

In the CTC++ run-time library:

- Bug fix: Fixed time-stamp handling in native 64-bit builds.
- Bug fix: 64-bit timing adjustments in ctc.h for 64-bit builds
- Minor: For Cygwin builds, we pass internal 8.3 names to the ctc2dat utility

- Bug fix: Fix a resource leak

In the CTC++ postprocessor (ctcpost):

- Bug fix: Fixed time-stamp handling in native 64-bit builds
- Bug fix: Fixed a crash. Happened when Java's labeled break pointed to an 'if' with a missing corresponding 'else'.
- Bug fix: specific fix for MSB big endian machine (int vs. long issue).

In the CTC++ XML Merger utility:

- Bug fix: Fixed a too much memory consumption crash, when many probes on a single physical line.
- Bug fix: Fixed issue, when overlenghted entries in headers, e.g. many MON.sym files mentioned. Previously, if that entry was bigger then 32000 it reported that input is not an CTC++ xml file, now line can be unlimited.

In the CTC++ Coverage Comparator utility (ctcdiff):

- New: This is first released version (v1.0) of the component. Its detailed behavior may change in future depending on user's feedback.

In CTC++ to HTML converter (ctc2html):

- Enhancement: Opens now HTML in browser on all (graphical) platforms.

In the Visual Studio IDE Integration:

- Improvement: modify_msbuild_path.bat internal search path extended, finds now Microsoft.CSharp.*target files from newer MSBuild installations.

In the ctc2dat receiver utility:

- Bug fix: Fixed time-stamp handling for 64-bit builds

Version 8.2 (9 May 2017)

For this version, please have a look to
<http://www.verifysoft.com/ctcpp82.pdf>