



Offenburg (Germany) / Tampere (Finland), 13 January 2017

Please be informed that a new
Testwell CTC++ version 8.1 has been released.

Testwell Oy
Verifysoft Technology GmbH
29 December 2016

CTC++ System Version 8.1

This file describes the changes in successive versions of CTC++.
The latest version is described first.

Version 8.1 (29 December 2016)

This revision 8.1 of CTC++ has the following version numbers in its components:

This revision has the following version numbers in its components:

Preprocessor	8.1	(was: 8.0.1; seen by -h option)
Run-time libraries	8.1	(was: 8.0.1; seen by 'ident' command applied on the library in some environments)
Postprocessor	8.1	(was: 8.0.1; seen by -h option and in the listings)
Header file ctc.h	8.1	(was: 8.0.1; seen in the file)
Configuration file ctc.ini	8.1	(was: 8.0.1; seen in the file)
CTC++ to HTML Converter	5.3	(was: 5.2; seen by -h option)
CTC++ to Excel Converter	3.3	(unchanged: seen by -h option)
CTC++ XML Merger utility	3.2	(was: 3.1; seen by -h option)
ctc2dat receiver utility	3.5	(was: 3.4; seen by -h option)

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

Visual Studio IDE Integration	4.2	(unchanged; seen by clicking the Tw-icon in the dialog program and selecting "About...")
CTC++ Wrapper for Windows	3.5	(was: 3.4: 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)
------------------------	-----	--------------------------------

This CTC++ v8.1 version contains enhancements and bug fixes:

In the CTC++ preprocessor (ctc):

- Bug fix: Source line numbers were not counted correctly when certain kind of multiline templated input, e.g. 'Templname<smtg> \n obj;'. The coverage report (.txt and .html) had wrong line number mapping. This bug occurred only in CTC++ v8.0.1.
- Bug fix: In certain kind of nested template case, something like X< ... Y<...>>, the inner template class name Y was not recognized properly. Either the instrumented code did not compile or file end remained uninstrumented. This bug occurred only in CTC++ v8.0.1.
- Enhancement: When switching to an included file having instrumented code, which ctc sees in the form of a #line directive, now remembering also the line number where the switch started. The ctc2html phase later can now do more properly the coverage report HTML'ization.
- Change: If a code snippet is #included to a function, which is instrumented, also the code snippet is instrumented, regardless of the EXCLUDE/NO_EXCLUDE/NO_INCLUDE ctc.ini settings.
- Change: The ctc.ini SOURCE_IDENTIFICATION setting default value is changed from "as_given" to "absolute". Now in later ctc2html phase the source files get found better without needing to give -s options.
- New/Change: Introduced setting SKIP_PARAMETER_LIST=identifierlist to ctc.ini. If such identifier is met followed by '(...)', nothing is instrumented inside the parentheses, notably if there would be ternary-?:, which must not be instrumented. Previous hard-coded support on identifier '__compile_time_assert' is removed. E.g.,
SKIP_PARAMETER_LIST=__compile_time_assert, __builtin_object_size
- Enhancement: At Windows, in the ctc.ini setting COMMAND, the file extension '.exe' is no more necessary (but is still allowed). Now e.g.
COMMAND = cl
matches to 'cl', 'cl.exe', 'path\to\cl', 'path\to\cl.exe'.
- Change: No more recording into symbolfile the function parameter profile, function category, defined and declared (in file and at line) information. Later, in ctcpost phase in XML form report these ones are no more reported.
- Change: In the ctc.ini EMBED_FUNCTION_NAME setting it is no more accepted plain '*' wildcard argument. Triggering the coverage data write-out at every function end would not be wise anyway.
- New: Introduced environment variable CTC_COND_OPER_NO_INSTR. When it is set, ternary-?: decisions are not instrumented. Previously, and still, same effect was got by macro -DCTC_COND_OPER_NO_INSTR, but with it the counter locations are anyway allocated (always 0 hits), which lower the TER%.
- Bug fix: Better handling of gcc-specific 'asm volatile(...);', 'asm goto(...);' and 'asm volatile goto(...);'. Previously, a ';' inside the parentheses could confuse ctc's parsing of the code.

In the CTC++ run-time library:

- Enhancement: More robust behavior, for avoiding certain problems, in a situation when two threads of an instrumented program trigger coverage data write-out at a same time. Now the later thread's write-out request is just silently ignored. The ongoing write-out takes care also of the later thread's need.

In the CTC++ postprocessor (ctcpost):

- Enhancement: The coverage report can now be restricted also by header files (new handling of -f and -nf options).
- Change: The format of line directive (in .txt and .xml form coverage reports) now has also the line number from what line at the first file the file switch was.
- Change: The coverage report (.txt and .xml form) bottom line summary no more has the "Measurement points : n" line.
- Change: The .xml form coverage report no more contains the following tags per each function: <parameters>, <category>, <declared_in_file>, <declared_at_line>, <defined_in_file>, <defined_at_line>.
- Bug fix: The XML form coverage report had wrong value in <lines> item in <file_summary> sections. The ctcxmlmerge step later now reports the line count correctly in its per files and overall summary.
- Bug fix: If a C++ try-function catch(...){...} blocks had #includes (brought instrumented code), the report writing failed. Now fixed.

In the Visual Studio IDE Integration:

- Enhancement: The "basic integration engine" (vsCTC.exe) is unchanged, but for installation there is improved modify_msbuild_path.bat for better C# Framework64 support.

In the CTC++ Wrapper for Windows (ctcwrap):

- Enhancement: Technical changes how ctcagent.ex_ component passes information of ctc options, compile command and its arguments to ctc-instrumentation. The new arrangement avoids problems with very long commands.
- Enhancement: More robust ctcwrap.bat behavior when handling command like 'ctcwrap ... "path to\vcbuild" ... "arg with spaces" ...'.

Testwell CTC++ Version 8.1 - page 3

In the CTC++ to HTML converter (ctc2html):

- Change: The utility accepts only v8.1 level (or later) input.
- Enhancement: The html'ized report is now more "accurate" in source file switches, thanks to the new line directive format in input.
- Change: Moved options -no-sources and -no-javascript to --enable-XXX=1/0 category of options.
- Bug fix: Now propely generating HTML of a source file having line number > 99999 (real big file or forged by #line directive)

In the CTC++ XML Merger utility:

- Change: The utility accepts only v8.1 level (or later) input.
- Bug fix: Now properly handling and accepting a response file command line argument, whose last line did not end to a newline.

In the ctc2dat receiver utility:

- Enhancement: The utility gave "Internal error" message and gave up processing, if the input was "really bad" (seriously corrupted?). Now also in such cases it continues and tries to find a re-synck from next <START:... section, or until file end.

In the Host-Target add-on (HOTA):

- New: Upgraded the HOTA package from v5.2-->v5.3 (separate delivery package with its own version.txt etc.). Notably also the writing of MON.txt by parallel threads is handled more robustly, as in host run-time, which was a likely problem in some recent customer cases.

General:

- Change: The license check-out retrial (was used for some reasons before) count is lowered from 3 to 1. At Windows this means that Flexlm License Finder pop-up comes only 2 times (when FLEXLM_BATCH is not set), and needs to be canceled, in possible network problems.
- Compatibility: Best is to reinstrument the code base and generate the reports anew.
- CTC++ User's Guide upgraded to v8.1 level (ctcug.pdf).



Testwell CTC++ Version 8.1 - page 4

Version 8.0.1 (28 June 2016)
For this version, please have a look to
<http://www.verifysoft.com/ctcpp801.pdf>