



Offenburg (Germany) / Tampere (Finland), 19 May 2015

Please be informed that a new  
**Testwell CTC++ version 7.3.3** has been released.

Testwell Oy  
Verifysoft Technology GmbH  
15 May 2015

CTC++ System Version 7.3.3

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

Version 7.3.3 (15 May 2015)  
-----

This revision 7.3.3 of CTC++ has been worked up only for Windows platform. The idea is that this would be a kind of pre-version of v7.4 for some selected customers. The actual v7.4 on all supported platforms is expected to come later.

This revision has the following version numbers in its components:

Preprocessor	7.3.3	(was: 7.3.1; seen by -h option)
Run-time libraries	7.3	(unchanged; seen by 'ident' command applied on the library in some environments)
Postprocessor	7.3.3	(was: 7.3; seen by -h option and in the listings)
Header file ctc.h	7.3	(unchanged; ; seen in the file)
Configuration file ctc.ini	7.3.3	(was: 7.3[.1]; seen in the file)
CTC++ to HTML Converter	5.0	(was: 4.1; seen by -h option)
CTC++ to Excel Converter	3.1	(was: 3.0; seen by -h option)
CTC++ XML Merger utility	2.3	(was: 2.2; seen by -h option)
ctc2dat receiver utility	3.2	(unchanged; 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	3.3	(was: 3.2; 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)
------------------------	-----	--------------------------------

Testwell CTC++ Version 7.3.3 - page 2

This CTC++ v7.3.3 version contains enhancements and bug fixes:

In the CTC++ preprocessor (ctc):

- New: Introduced configuration parameter `WARNING_LEVEL` with possible values `none` (like `-no-warnings` option), `warn` (normal warnings), `info` (some new warnings of tool limitations).
- New: introduced `#pragma CTC ANNOTATION annotation_text`. The `annotation_text` comes visible to CTC++ reports (`.txt`, `.xml`, `.html`).
- Bug fix: When `"CTC SKIP"; ... "CTC ENDSKIP";` is used to skip (not to instrument) code in a function, the starting `"CTC SKIP";` is no more counted to a statement.
- Enhancement: On primary host platforms fine-tuned the default `ctc.ini` files to work more properly with some rarely used compiler options.
- Enhancement: For newer Borland compiler versions, taught `ctc.exe` to work properly with so called BOM marker, which the compiler can internally generate to a C-preprocessed file.
- New: Introduced new `'ASSEMBLY_COMMENT_CHARS=chars'` configuration parameter. Helps in handling new kind of assembly dialects, e.g. where `';` does not start (assembly-)comment.
- Change: If there is `'const'` in a function, e.g. in context  
`const int xsize = X > Y ? 10 : 20;`  
the decision in the ternary is no more instrumented. If it were instrumented, possible later use of `xsize` in an array declaration would cause a compilation error.
- Bug fix: Declaration in condition starting with `'constexpr'` (C++11) was instrumented incorrectly resulting in a compilation error. For example,  
`if (constexpr T t = f()) ...`  
(also in `while` and `for`). Now it is instrumented like other declarations in a condition.
- Bug fix: Under some circumstances, deriving the datafile name from symbolfile name was erroneous. E.g., if the symbolfile name was `"D:\symbols\MON.sym"`, its datafile name came `"D:\datbols\MON.dat"`. Now it comes correctly `"D:\symbols\MON.dat"`.
- Bug fix: In some (build system generated) response files there could occur the following kind of file names `"dir1\dir2"\file.xxx`. Of these `ctc.exe` erroneously parsed `dir1\dir2\ file.xxx`. Now they are parsed to `dir1\dir2\file.xxx` or to `"dir1\dir2\file.xxx"`, if there is real reason to use `"..."`.

- New/Enhancement: When C code, functions with old-style (K&R1) parameter declarations are now instrumented. For example:

```
int foo()
  int i;
  {...function body;}
```

Previously this type of K&R1 code was forbidden, not instrumented, and sometimes could cause nondeterministic behavior.
- Bug fix: Some extreme case "... " safeguarding in the ctc-generated temporary file names (e.g. if they needed to be created to a directory with space in the path).
- Bug fix: In Unixes, some extreme case corrections in handling adjacent '\ ' and ' ' and '\ ' characters on the command line for preserving the intended value in ctc-constructed intermediate commands. E.g. in

```
-DMACRO='\ \ \ \'xxx\ \ \ \' , -DMACRO='\ xyz\ \ \ '
```

In the CTC++ postprocessor (ctcpost):

- New: For helping the ctc2html utility to generate HTML report with line coverage information, some new information is added to textual execution profile report. They are (and have meaning):
  - each if/for/while/do/switch/try statement block ending-} (either when there is or is not explicit '}' in user's code) show as "}"+" (code after } was entered) or "}"-" (code after } was not entered) line.
  - labels show now as "L:" (L was "flown to", and possibly also "gotoed") or "L:+" (L was not "flown to" but it was "gotoed") line.
  - case labels show now as "case n:" (this case was explicitly "jumped to") or "case n:+" (this case was not explicitly "jumped to" but it was anyway "flown to").
  - "default:" or "default:+" lines (same logic as in "case n:[+]").Same information is produced to XML form report. These additional ± markings are not added, if statement coverage is not available on the function.
- New: Annotation concept introduced for commenting. In the user's code there can be line "#pragma CTC ANNOTATION text", where the "text" shows in the generated report (and later in the HTML report).
- Enhancement: Made the utility more robust against some crashes (they could occur in some pathological cases, if only parts of a function were instrumented and real control structure was not known).
- Change: In statement coverage flow analysis, if previous probe was hit, and statement list following it was started, and next probe is not hit (e.g. due to exception occurred or control remained in some called code), all statements upto next probe are counted to hit statements. Previously they were excluded from hit statements.

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

- Bug fix: "ctcwrap -hard ..." now work also if the specified compiler is given with path that contains spaces.
- Enhancement: In "ctcwrap -hard ..." use some underlying absagent\*. \* components are made more tolerable/recoverable in situation when at the build context CTCHOME is not set or if it is not in PATH.
- Change: Some new sanity checks on errors on command line arguments and clarified some error messages.

In CTC++ to HTML converter (ctc2html):

- New: In the HTML form report introduced 'line coverage'. It shows as color-coded lines what code lines were/were not executed.
- Change: Greatly rearchitected and improved the look-and-feel of the HTML report, notably the source code mapped Execution Profile HTML page. It has structural coverage counters as before, but the color-coding is redesigned and there is new line coverage color-coding on the source code lines.
- New: Introduced (to advanced users) capability to modify the used style sheet for fine-tuning used colors and other graphical details in the generated HTML pages.
- New: Some new --enable-XXX=... options added.
- New: --enable-XXX=0/1 options can alternatively be given as --disable-XXX/--enable-XXX on command-line.
- Change: Syntax-highlighted presentation is now default in displaying the source code (with option to disable it).

In the CTC++ XML Merger utility:

- Change: Adapted to the new ctcpost generated .xml reports.
- Change: When inside a function there is an #include, which brings code (that was also instrumented), the #included file is no more pulled out to a separate MONITORED HEADER FILE element in the report. Instead the #included code is reported as part of its includer function.

In the CTC++ to Excel Converter (ctc2excel):

- Bug fix: corrected a bug in lines having FILE OVERALL column. All columns after this were shifted to the left, and FILE word was lost.



Testwell CTC++ Version 7.3.3 - page 5

General:

- CTC++ User's Guide upgraded to v7.3.3 level (ctcug.pdf).

Version 7.3.1 (3 October 2014)  
-----

For this version, please have a look to  
<http://www.verifysoft.com/ctcpp731.pdf>