



Tampere (Finland) / Offenburg (Germany), 27 February 2009

Please be informed that a new
CTC++ version 6.5.4 has been released.

This version contains some bug fixes and enhancements. Of the latter mentioning:

- Introduced 64-bit support as standard part of the CTC++ delivery on each supported platform (Windows, Linux, Solaris, HPUX).
- Introduced an easy to use arrangement to make periodic coverage data writing from the instrumented executable by an auxiliary side thread. Can be used e.g. when testing never-ending processes.
- Constructs like 'while (1) {...}' are no more instrumented. The impossible situation that the condition is false is no more alarmed, which lowered the TER%.

The new version is available on all supported platforms. It is downloadable for support customers from the Testwell Customer Area of our web pages in the normal manner.

Olavi Poutanen
Testwell

```
----- excerpt of CTC++ v6.5.4 version.txt:
Testwell Oy                               CTC++ System Version 6.5.4
26 February 2009
```

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

Version 6.5.4 (26 February 2009)

This revision 6.5.4 of CTC++ has the following version numbers in its components:

Preprocessor	6.5.4	(was 6.5.3 seen with the -h option)
Run-time libraries	6.5.4	(was 6.5.3, seen by the 'ident' command applied on the library in some environments)
Postprocessor	6.5.4	(was 6.5.3, seen with the -h option and in the listings)
Header file ctc.h	6.5.4	(was 6.5, seen in the ctc.h comments)
Configuration file ctc.ini	6.5.4	(was 6.5, seen in the ctc.ini header)
CTC++ to HTML Converter	2.5	(was 2.4, seen with the -h option)



Testwell CTC++ Version 6.5.4 - page 2

CTC++ to Excel Converter	1.1	(unchanged, seen with the -h option)
CTC++ Merger utility	1.0	(unchanged, seen with the -H option and in the merged listings)
ctc2dat receiver utility	2.0	(new, previously part of CTC++ Host-Target add-on)

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

CTC++ IDE Integration	3.2	(unchanged, seen by clicking the Tw-icon in the dialog program and selecting "About...". This integration is used at - Visual Studio .NET 2003/2005/2008 IDEs - CodeWarrior IDE [Symbian/emulator] - Carbide.c++ IDE [Symbian/emulator]) - Eclipse IDE
-----------------------	-----	--

Visual Studio 5/6 Integration	2.2	(unchanged, version number seen by clicking the TW-icon in the CTC++ dialog boxes and selecting "About CTCui...")
-------------------------------	-----	---

CTC++ Wrapper for Windows	2.4	(was 2.1, seen by "ctcwrap -h")
---------------------------	-----	---------------------------------

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

CTC++ Wrapper for Unix	1.3	(was 1.2, seen by "ctcwrap -h")
------------------------	-----	---------------------------------

The corrections and enhancements in this version are the following:

In the CTC++ preprocessor (ctc):

- New: Options -no-warnings and -no-comp have been added.
- Bug fix: Typedef'ed classes, structs and unions were not recognized and processed correctly, if the typedef keyword was followed by some qualifier or modifier. For example in the following:

```
typedef const class C { ... } CC;  
typedef __declspec(...) struct D { ... } DD;
```
- Bug fix: An escaped backslash was not properly preserved in the tool chain (ctcwrap.bat, ctcagent.ex_, ctc.exe), when there was e.g. the following '... "X:\include dir\\" ...' on the command line.
- Bug fix: If a template instantiation, as a string, was longer than 1024 characters, it was truncated, and the instrumented code did not compile. Example: `T<sizeof("very...long...string")> t;.`
- Bug fix (Windows only): Compiler/linker commands and source files were not identified at all or, in some cases, processed correctly, if given like 'C:cl' (vs. 'C:\dir\cl') or 'F:file.cpp' (vs. 'F:\dir\file.cpp').



Testwell CTC++ Version 6.5.4 - page 3

- Enhancement (Windows, MICROSOFT dialect, C++/CLI code only): Keywords 'abstract' and 'sealed' are now properly handled in class and struct definitions. For example:

```
ref class C abstract { ... };
```

Keywords 'override', 'sealed' and 'new' are now properly handled in function declarations and definitions. For example:

```
virtual int f() override { ... }
```

- Enhancement: In the `-i m` instrumentation, there is a limit on the evaluation alternatives that `ctc` can still instrument in the `-i m` way. The limit has grown from 270 to 500. Beyond that, the `-i d` instrumentation is used.
- Change: If the condition expression in `if`, `for`, `while`, `do-while` or `ternary-?:` is a literal, instrumentation is not done. Now for example `'while (1) {...}'` gets no true/false counters and in the TER calculation there is no "penalty" that the condition has not been evaluated in both ways.
- Enhancement: Some compilers (at least VC++ 8.0/9.0) allow the identifier 'default' to be used as a user's variable name. Now the same is allowed by CTC++: identifier 'default' is considered the keyword 'default' only if it is followed by `':'`.
- Problem fix (Windows only): The length of the compiler or linker command can now be longer than 2047 (Win2000) or 8191 (WinXP) characters. In practice, this was a problem only with such compilers and linkers that do not support the use of response files (e.g., the Symbian GCCE build environment). Normally, CTC++ uses response files with long command lines.
- Enhancement: If, after the instrumentation phase, it turns out that not a single function got instrumented, certain `ctc`-internal code is not generated. As this code would be unused, compiler warnings might follow.
- Enhancement: If the compiler, linker or other command emitted by the CTC++ preprocessor fails, the return code (other than 0 or -1) is now shown in the corresponding CTC++ error message. For example

```
CTC++ error 8: Cannot execute C or C++ compiler or compilation \  
failed: 0xC0000005
```
- New: Introduced an arrangement how coverage data writing to a datafile can be activated periodically from a "side-thread". This is especially useful in never-ending process cases. See the CTC++ User's Guide.

In the CTC++ run-time library:

- New: On all supported host platforms (Windows, Linux, Solaris, HP-UX), introduced 64-bit support as standard part of the delivery package.



Testwell CTC++ Version 6.5.4 - page 4

In the CTC++ postprocessor (ctcpost):

- Bug fix: ctcpost is made more robust in the extreme cases where the same source file was instrumented twice during one second (the granularity of CTC++ timestamp) but changed in the between. Counter data vector sizes could have changed. Previously, when processing such coverage data instances, ctcpost could give unpredictable results or crash.
- Change: 'ctcpost -h' or 'ctcpost' now exits with status 0 (was: 1). This is in accordance with 'ctc -h', and may be relevant in makefile use.

In CTC++ to HTML converter (ctc2html):

- Enhancement: Changed the titles of the HTML pages to more descriptive.
- Enhancement: Added "Top" links to the detailed Execution Profile HTML pages to support easier navigation.

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

- Bug fix: In parallel use (ctcwrap command issued from separate command prompts), ctcwrap is made more robust against certain race conditions when reserving a %temp%\ctc<n> directory for its use.
- Enhancement: Added support for the new Symbian Build System (SBS). Its use is similar to the ABLD based builds. For example:

```
ctcwrap -i d -v sbs -c winscw_udeb build
```

- Change: 'ctcwrap -h' or 'ctcwrap' now exits with status 0 (was: 1). This is in accordance with 'ctc -h', and may be relevant in makefile use.
- Enhancement: It is now checked that ctcwrap is not called recursively.

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

- Enhancement: Added certain "sanity checks" that the CTC++ installation is healthy.
- Bug fix: An escaped dollar sign, e.g. \`\$ABC`, is now properly handled.

General:

- CTC++ User's Guide upgraded to v6.5.4 level.