Tampere (Finland) / Offenburg (Germany), 15 February 2008

Please be informed that a new **CTC++ version 6.5.2** has been released.

The upgrade is available on all supported platforms (Windows, Linux, Solaris and HPUX). The new version is downloadable for support customers from the Testwell Customer Area of our web pages in the normal manner.

The previous version was CTC++ v6.5 from February 2008.

Primarily this v6.5.2 is a bug fix version on v6.5 on certain special cases where C++ is used "heavily". It may well be that in your code base there are no such incidents where these ctc bugs would show up.

See the VERSION.TXT excerpt below for full v6.5.2 details.

The previous version of CTC++ 6.5 was CTC++ v6.4 from July 2007. Mentioning of this v6.5 version the following:

- Introduced XML reporting
- Enhanced the 'ctcwrap' utility to support parallel ctc-builds
- various other enhancements and bug fixes

Version 6.5.2 (6 March 2008)

This revision 6.5.2 of CTC++ is a bug fix version over CTC++ 6.5.
In this revision effectively only the preprocessor component has changed. (On the Windows platform, a partially fixed ctc.exe v6.5.1 was made available, but here the changes are compared to v6.5[.0]).

Preprocessor	6.5.2 (was 6.5, seen with the -h option)

All the other CTC++ subcomponents are unchanged from the v6.5 level (see below the section "Version 6.5" for details).

The corrections and enhancements in this version are the following:

In the CTC++ preprocessor (ctc):

- Bug fixes: Corrected 3 C++ specific cases, which may be not so common in typical code. The exact situations are too complex to explain in detail. However, it was needed that multiple namespaces, class forward declarations, typedef definitions for classes, inheritance, templates, instantiations were involved.

  -- Under some special conditions, the following kind of code

```
...if (Namespace::TemplateClass<Params> object = value)...
```

  was parsed erroneously resulting in non-compilable instrumented code. Now fixed.

  -- Under some special conditions, the following kind of code

```
...typedef OldClass X;       <-- in some namespace A
...
...class X : public Y {...   <-- in another namespace B
```

  caused ctc to crash. Now fixed.

  -- It is possible that an innocent-looking function local variable hides a template class or function name in an outer scope. Then, if there was the following kind of use

```
...int _Count = 5;           <-- hides an outer template _Count
...
...if (_Count < 10)...       <-- ctc failed here
```

  ctc parsed the code erroneously, resulting in, for example, ctc to report a syntax error or non-compilable instrumented code. Now fixed.

- Enhancement: Started to support ">>" as two closing angle brackets of nested template argument lists. For example, the following case

```
...templatename1<templatename2<int>>...
```

  We consider that this is actually against C++, but some compilers, for example VC++ 8.0, have started to allow this. And so does also ctc now.

Full v6.5 details:

Version 6.5 (12 February 2008)

This revision 6.5 of CTC++ has the following version numbers in its components:

Preprocessor	6.5	(was 6.4, seen with the -h option)
Run-time libraries	6.5	(was 6.4, seen by the 'ident' command applied on the library in some environments)
Postprocessor	6.5	(was 6.4, seen with the -h option and in the listings)
Header file ctc.h	6.5	(was 6.4, seen in the ctc.h comments)
Configuration file ctc.ini	6.5	(was 6.4, seen in the ctc.ini header)
CTC++ to HTML Converter	2.3	(was 2.2, seen with the -h option)
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)

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

CTC++ IDE Integration	3.1	(was 3.0, seen by clicking the Tw-icon in the dialog program and selecting "About...". This integration is used at - Visual Studio .NET 2003 IDE - Visual Studio 2005 IDE - CodeWarrior IDE [Symbian/emulator] - Carbide.c++ IDE [Symbian/emulator])
Visual Studio 5/6 Integration	2.2	(was 2.1, version number seen by clicking the TW-icon in the CTC++ dialog boxes and selecting "About CTCui...")
CTC++ Wrapper for Windows	2.0	(was 1.0, seen by "ctcwrap -h")

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

CTC++ Wrapper for Unix	1.2	(was 1.1, seen by "ctcwrap -h")

The corrections and enhancements in this version are the following:

In the CTC++ preprocessor (ctc):

- New: Certain new information is now collected regarding the instrumented functions: file name and line number where declared, parameter profile, and "category" (public, protected, private, standalone). This information is shown in the new XML report (see ctcpost).

- Bug fix: When a destructor of a nested class is defined outside the class declaration, e.g.,

```
Outer::Inner::~Inner() {...}
```

  its name is now shown correctly as "Outer::Inner::~Inner" in coverage listings. Previously it was erroneously shown as "Inner".

- Bug fix: Certain kind of bit field declarations caused non-compilable instrumented code, for example,

```
struct S {...; TYPEDEF_NAME (field) : 4; ... };
```

  The following conditions were required to trigger the bug: the type specifier had to be a typedef name (not a basic type), there had to be extra parentheses around the member name, and it had to be followed by a colon and a bit-field width. Now fixed.

- Bug fix: If decision coverage instrumentation (the default) was used together with safe counter incrementing (i.e., the macro CTC_SAFE was defined), the following kind of condition

```
if (ClassTypeObject) {...
```

  could in many cases cause non-compilable instrumented code (unless some suitable conversion function, e.g., "operator int()", was available). This was a problem with C++ (not C) code only. Now fixed.

- Change: In coverage listings, names of class template member functions and function template specializations are now shown with the <...> included. For example,

```
long X<long>::foo(){...}          // now "X<long>::foo", was "X::foo"
template <> int bar<int>(){...}   // now "bar<int>",     was "bar"
```

- Change: For example, in the following case

```
void foobar
   (int i) {...
```

  the function's start line is now the line in which the function name 'foobar' is located. Previously, the start line was that of the opening parenthesis of the parameter list. (Now, in the HTML report, the function start line is more accurately mapped to the source file.)

- New: A system error message - generated by a call to strerror(errno) - is now shown at the end of all file error messages. For example,

```
CTC++ error 9: Cannot create file X:\MON.sym: Permission denied
```

- Change: Verbose messages (enabled by the -V and/or -v option) are now written to stderr. Previously they were written to stdout, which could cause problems in some special build scenarios.

In the CTC++ run-time library:

- Change: An explicit FLEXlm license check-in call is now done at the end of the instrumented program execution. The only case we know this to have some user-noticeable effect is the use of the Symbian EPOC emulator on Windows. In that context, the license linger count-down (time when the license is returned to the free license pool) will now start when the run of the instrumented program has ended within the EPOC emulator. Previously, it did not start until the whole EPOC emulator session had ended.

In the CTC++ postprocessor (ctcpost):

- New: Added option "-x outputfile" for generating an XML format report.

- Bug fix: ctcpost did not work correctly in the following situation
  -- there were two or more input .sym files
  -- which contained descriptions of the same instrumented source file, and the file descriptions were identical except for the timestamps (concluding that the same file was in question, but instrumented independently at different times)
  -- there were two or more input .dat files containing coverage data for this source file and the timestamps for this file matched in the .sym and .dat files (presumably separate test runs, using different instrumented versions of this file)
  The aggregation (summing) of the two or more coverage results failed. CTCPost internal error 11 was displayed and the run was aborted. Now fixed.

- Bug fix: ctcpost crashed, when "-a outputdatfile" option was used and the creation of the target datafile failed. Now fixed.

- New: A system error message - generated by a call to strerror(errno) - is now shown at the end of all file error messages. For example,

  CTCPost error 8: Cannot create file X:\profile.txt: Permission denied

- Change: The "CTCPost notice messages", if there are any, are now displayed always. Previously they were displayed only if the -v option was specified.

- Change: Verbose messages (enabled by the -V option) and notice messages are now written to stderr. Previously they were written to stdout.

In CTC++ to HTML converter (ctc2html):

- New: Added option "-o output-dir" (default directory is ./CTCHTML)

- Enhancement: The "-s source-dir" option behavior is extended so that source file searching is performed also based on relative directory paths.

- New: In the Directory Summary view of an HTML report, there is now JavaScript-based on-line sorting by various columns.

- New: Added option "-no-javascript". It prevents generation of Javascripts into the HTML report (they can be unwanted for some browsers or browser settings).

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

- Change/bug fix: Now the same user can have multiple parallel and independent ctcwrap commands in execution. Previously, the technical solution to use the file ~/ctcopts.rsp to remember the ctc options prevented parallel use.

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

- Enhancement: Now there can be parallel and independent ctcwrap commands in execution at the same time (each command running in its own command shell).

- Bug fix: For example, in the "ctcwrap ... abld ..." case, if the ctcwrapped command (here "abld") emitted an argumentless compilation command, e.g. "armcc", there was such an error that the compilation command was not executed at all. Now fixed.

In the IDE integrations on Windows:

- Visual Studio 5/6 IDE integration: Adjusted to CTC++ v6.5 level, notably the XML reporting. See %CTCHOME%\Devstud\version.txt.

- Visual Studio .NET 2003 and 2005 IDE integration: Adjusted to CTC++ v6.5 level, notably the XML reporting. Also some other "implementation technical changes". See %CTCHOME%\Vs_integ\version.txt.

- New: Checked that Visual Studio .NET 2003 / 2005 IDE integration works also with Visual Studio 2008.

In CTC++ for Symbian EPOC Emulator add-on:

- See %CTCHOME%\Sym_cw\version.txt.

General:

- CTC++ User's Guide upgraded to v6.5 level.

Version 6.4 (5 July 2007)
---
for further information: http://www.verifysoft.com/ctcpp64.pdf