Tampere (Finland) / Offenburg (Germany), 17 March 2010

Please be informed that a new **CTC++ version 6.5.6** has been released.

This version contains bug fixes and enhancements. Some of the enhancements are developed to enable handling of certain special situations that have come up in Symbian or QT code use. See details from the v6.5.6 version text below.

The new version is available on all supported host platforms.

Version 6.5.6 (16 March 2010)

This revision 6.5.6 of CTC++ has the following version numbers in its components:

Preprocessor	6.5.6	(was 6.5.5; seen by -h option)
Run-time libraries	6.5.6	(was 6.5.5; seen by 'ident' command applied on the library in some environments)
Postprocessor	6.5.6	(was 6.5.5; seen by -h option and in the listings)
Header file ctc.h	6.5.6	(was 6.5.5; seen in the ctc.h comments)
Configuration file ctc.ini	6.5.6	(was 6.5.5; seen in the ctc.ini header)
CTC++ to HTML Converter	2.6	(was 2.5; seen by -h option)
CTC++ to Excel Converter	1.2	(was 1.1; seen by -h option)
CTC++ Merger utility	1.1	(was 1.0; seen by -H option and in the merged listings)
ctc2dat receiver utility	2.0	(unchanged; seen by -h option)

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...")
Visual Studio 5/6 Integration	2.2	(unchanged; seen by clicking the TW-icon in the dialog program and selecting "About CTCui...")
CTC++ Wrapper for Windows	2.7	(was 2.5; seen by -h option)

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

```
    CTC++ Wrapper for Unix     1.3   (unchanged; seen by -h option)
```

The corrections and enhancements in this version are the following:

In the CTC++ preprocessor (ctc):

- Enhancement: Introduced four new configuration parameters for better handling of certain demanding use cases:

  -- RUN_BEFORE_ALL: user-defined script can fine-tune instrumentation options and do other actions before the actual "ctc-processing"
  -- OPT_ADD_PREPROC: e.g. for handling some problematic precompiled header file cases (adding -I options as needed)
  -- DATAFILE: explicitly determining the datafile path and name, and not deriving it from the symbolfile path and name
  -- OPT_NO_LINK: removing options from "ctc-link"

- Bug fix: The CTC++ preprocessor now recognizes "__pragma(...)", a Microsoft extension in VC++ 7.x and later. Previously, it could sometimes cause problems.

- Bug fix: Better handling of "__extension__ (...)" cases. Previously, it could happen that something was incorrectly instrumented. Now, as with GCC extensions in general, nothing is instrumented inside (...).

- Bug fix: Various GCC __attribute__ and __attribute extensions are now better handled, e.g. in the following cases:
  ```
     void abort(void) __attribute__((__noreturn__)) { }
     class __attribute__((aligned(16))) C { };
     typedef __attribute__((align(4))) class D { } DD;
     class E { } __attribute__((__packed__));
  ```
  Previously, for example, some function could remain uninstrumented.

- Bug fix. A declaration in condition could in certain cases be incorrectly instrumented. For example (occurred in QT code),
  ```
     template<typename T> class list;
     typedef long LONG;
     if (LONG list = f()) { }
  ```

- Bug fix: In some (apparently rare) cases, it could happen that a source file remained entirely uninstrumented. An example of the problem is that at the command line the source file name was given with an extra/unneeded "/./", but the compiler C-preprocessor had taken this away from the #line directive in the C-preprocessed file.

In the CTC++ run-time library:

- Change: Some technical internal changes, no effect to normal users.

In the CTC++ postprocessor (ctcpost):

- Enhancement: Introduced new option -nf for unchoosing source files from output listings. Cf. option -f having reverse functionality.

- Enhancement: Now also symbolfiles can be combined (added) with the -a option. Previously, this was supported only for datafiles.

- Enhancement: It is now allowed to use -f (and the new -nf) option with the -a option to constrain the resultant symbolfile or datafile.

- Change: Now explicit -p is required to get an Execution Profile Listing. Previously, it could appear "suprisingly" with some option combinations.

- Change: In various reports, the report header information is adjusted a bit, esp. the Chosen/Unchosen source files (-f/-nf) information.

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

- Enhancement: Added options -rvct22, -rvct31, -rvct40 and -no-conf-check to ctcwrap. (Symbian usage issues only)

In CTC++ to HTML converter (ctc2html):

- Enhancement: Now the directories of the used symbolfiles are also used as places where the source files are searched from (as if those directories were given last using the -s options)

- Bug fix: Corrected a case where an Execution Profile Listing contained no instrumented files (generated e.g. using ctcpost -f/-nf options).

In CTC++ Merge utility (ctcmerge):

- Bug fix: Now allowing Execution Profile Listing merging from its common #included portions also in a situation when the input listing is generated using ctcpost -ff (reduce to function coverage) option. Also a bug fix, if the code was instrumented for timing.

General:

- CTC++ User's Guide upgraded to v6.5.6 level.

Version 6.5.5

Information available from http://www.verifysoft.com/ctcpp655.pdf

Verifysoft Technology GmbH – Technologiepark – In der Spöck 10 – D-77656 Offenburg
Geschäftsführer : Klaus Lambertz - Handelsregister Freiburg HRB 472242
Bank : Sparkasse Offenburg/Ortenau BLZ 664 500 50, Konto : 568 719
IBAN : DE30 6645 0050 0000 5687 19 SWIFT-BIC : SOLADES1OFG
Tel +49 781 6392-027 Fax +49 781 6392-029 Email: info@verifysoft.com Internet: http://www.verifysoft.com