Verifysoft TECHNOLOGY

Offenburg (Germany) / Tampere (Finland), 16 May 2017

Please be informed that a new
Testwell CTC++ version 8.2 has been released.

Testwell CTC++ v8.2 available

CTC++ v8.2 contains corrections of the bugs that have been found since
the previous version. There are also enhancements that are mainly
focused on various advanced use cases. As a new thing, the coverage
report can now be written also in JSON (JavaScript Object Notation)
format, logically similar to XML report, but easier and faster to
process in javascripts.

The CTC++ Host-Target add-on (HOTA) component is upgraded from
subversion v5.3 to v5.4.

The VERSION.TXT excerpt below explains the changes in more detail.

Support customers can download the new version from Testwell Customer
Pages in the normal manner. There are now versions on Windows, Linux
and HPUX. Other supported platforms come later from Verifysoft.

Olavi Poutanen
Testwell

--------------Excerpt of VERSION.TXT:
Testwell Oy   CTC++ System Version 8.2
                Verifysoft Technology GmbH
                        9 May 2017

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

Version 8.2 (9 May 2017)

This revision 8.2 of CTC++ has the following version numbers in its
components:

This revision has the following version numbers in its components:

        Preprocessor   8.2   (was: 8.1; seen by -h option)
         Run-time libraries   8.2   (was: 8.1; seen by 'ident'
                                    command applied on the library
                                      in some environments)
    Postprocessor   8.2   (was: 8.1; seen by -h option and in
                            the listings)

```
Header file ctc.h           8.2   (was: 8.1; seen in the file)
    Configuration file ctc.ini 8.2   (was: 8.1; 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   3.3   (was: 3.2; seen by -h option)
    ctc2dat receiver utility   3.6   (was: 3.5; 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   (unchanged: 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++ v8.2 version contains enhancements and bug fixes:

In the CTC++ preprocessor (ctc):

- Bug fix: When a struct variable was initialized in global scope and
  no explicit '=' was given (allowed in newer C++ standards), e.g.
  'S s[]{1, 2, i, j};', ctc confused this with a lambda function and
  generated non-compilable code. Now fixed.

- Bug fix: When a lonely inline asm statement appeared e.g. as
  then-part or as else-part of an if-statement, or similar, without
  being enclosed by {...}, ctc identified the control structure
  nesting erroneously or could generate non-compilable code. Now fixed.

- Bug fix: Now properly handling a line that contains only one char '\'
  (line continuation). Depending on the used compiler brand and this
  ctc bug, even non-compilable code could result in some rare cases.

- Enhancement: At Windows, if a long source file name+path is given
  in 8.3 (short filename) form, it is recorded in symbolfile in long
  form. Certain duplicate file name problems get avoided.

- Enhancement: Better parsing of #pragma lines, e.g. in the following
      switch(...) {
      #pragma XXX
      case 1: ...

Now the 'case 1' is properly identified as the first case label and instrumented appropriately. This was not a problem in C/C++ code, but in C# code there was dead code, and the instrumented code did not compile.

- Enhancement: At Windows, the setting COMMAND=XXX in the ctc.ini file gives now a match also to [path\to\]XXX-orig[.exe]. This makes the 'ctcwrap -hard -modeon ... [path\to\]XXX.exe' usage easier.

- New: Introduced the configuration setting CONST_INSTR={ON|OFF} in ctc.ini. It has effect on statements like 'const int i = expr;'. If the value is ON, the possible ternary-? and &&,|| (when -i m instrumentation) are instrumented in the 'expr'. Such instrumentation is not possible, if 'i' is later used so, e.g. as array size, that it needs to have the 'const' property. Default value is OFF ('expr' not instrumented here).

- Clarification: CTC++ v8.0.1 version.txt had a note that a lambda function like 'f([=]()->int{return 1;}());' no more generates non-compilable instrumented code. It is correct. But if the lambda function appears in context 'var = f([=]()->int{return 1;}());', it is not instrumented in '-i m' mode, but is instrumented in 'i -d' mode. This is a tool limitation.

- Change: ctc again records the function parameters profile into symbolfile. This recording was dropped off in v8.1. Additionally, it is now recorded also the possible 'const', 'volatile', '&', '&&', which can appear after '(...)', and can be the differentiating factor in function overload resolution.

In the CTC++ run-time library:

- Change: At Windows, the error messages are no more given via a popup message (needing user's action to proceed). Instead, they are just written to stderr (and the instrumented program aborts, as before)

In the CTC++ postprocessor (ctcpost):

- Bug fix: When generating an XML output file, the file-specifc <lines> tag had a wrong value under certain conditions.

- Bug fix: If there was a GCC extension in the following context: '...else if (...__extension__({statements;})...)...', the statements (the ';'s) of the 'else if' decision were left out from statement coverage TER% calculus. Now they are considered.

- Bug fix: Made the utility more robust (no more crash), i.e. tolerating
  some input which should not occur but has occurred anyway. (__try{} block with no __except(){} or __finally VC++{} to follow, or in xmlmerge connection #linedirectives caused wrong probe order).

- New: Introduced option '-j json-file'. It is similar to '-x xml-file'
  but the output file data format is JSON (JavaScript Object Notation).

- Change: The function parameters profile is taken back to the XML
  output file (was dropped off in v8.1)

- Change: To the generated Untested Listing there no more comes the
  possible line coverage auxiliary markings '+' or '-' even if the full
  Execution Profile Listing would have them. '}+', '}-', 'case n:+',
  'default:+' and 'Label L:+' are meant here.

In the Visual Studio IDE Integration:

- Enhancement: Functionality unchanged, but the integration is checked
  to work also with Visual Studio 2015 and 2017.

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

- Enhancement: The HTML form report can now be viewed also by character
  based viewers, like Lynx. Issue was in displaying the TER%
histograms.

- Enhancement: At Windows, when in the Execution Profile Listing a file
  is specified containing UTF-8 characters in its name, such a file can
  now be found, opened and html'ized to the HTML report.

- Change: To the generated Untested Code HTML page there no more comes
  the possible line coverage auxiliary markings '+' or'-' even if the
  input profile.txt would have them. '}+', '}-', 'case n:+',
'default:+'
  and 'Label L:+' are meant here.

In the CTC++ XML Merger utility:

- Bug fix: No more crash if by #line directives ctcxmlmerge is given a
  "wrong" understanding of the order of the probes in a function

- Bug fix: "case n:", "default:", "Label:" were not handled correctly
  in all cases when writing the merged-profile.txt. The error showed in
  wrong line coverage background color in the HTML form report.

- Bug fix: When statement coverage TER% was recalculated (if it was
  possible to do in the first place), in the merged-profile.txt the
  calculus sometimes was wrong at "break", "continue" or "goto Label"
  probes.

In the ctc2dat receiver utility:

- Change: Now uses the new v8.2 compliant CTC++ host run-time library
  (assumes v8.2 level ctc.ini, error reporting change in Windows)

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

- New: Upgraded the HOTA package from v5.3-->v5.4 (separate delivery
  package with its own version.txt etc.). Notably, now ensuring that the
  HOTA target runtime files and the instrumented code has same 32-bit
  perception of the counter sizes even if the used compiler has already
  type 'unsigned long' as 64-bit.

General:

- Change: At Windows, when the license is Flexlm-based floating or
  machine-locked, and when the license cannot be granted, now the
  Flexlm License Finder popup never appears (regardless of whether the
  the FLEXLM_BATCH environment variable is set or not). Instead, normal
  textual error message is written and the tool aborts.

- Compatibility:
  -- ctc2html accepts only a profile.txt input that has been generated
     by CTC++ v8.1 or later.
  -- ctcxmlmerge accepts only XML input that has been generated by
     CTC++ v8.2 or later.

Version 8.1 (29 December 2016)
For this version, please have a look to http://www.verifysoft.com/ctcpp81.pdf