



Release Note

3.4 — Last update: Dec 17, 2020

Suresofttech

Table of Contents

1. Improvements	1
2. Added Features	2
2.1. Add C/C++ RTV Target Project	3
2.2. Add Test Reconfiguration After Source Code Modification.....	4
2.3. Add Feature to Insert Default Values for Empty Input	7
3. Bug Fixed and Feature Modified.....	8
3.1. List of Fixed Bugs	9
3.2. List of Modified Features	10

1. Improvements

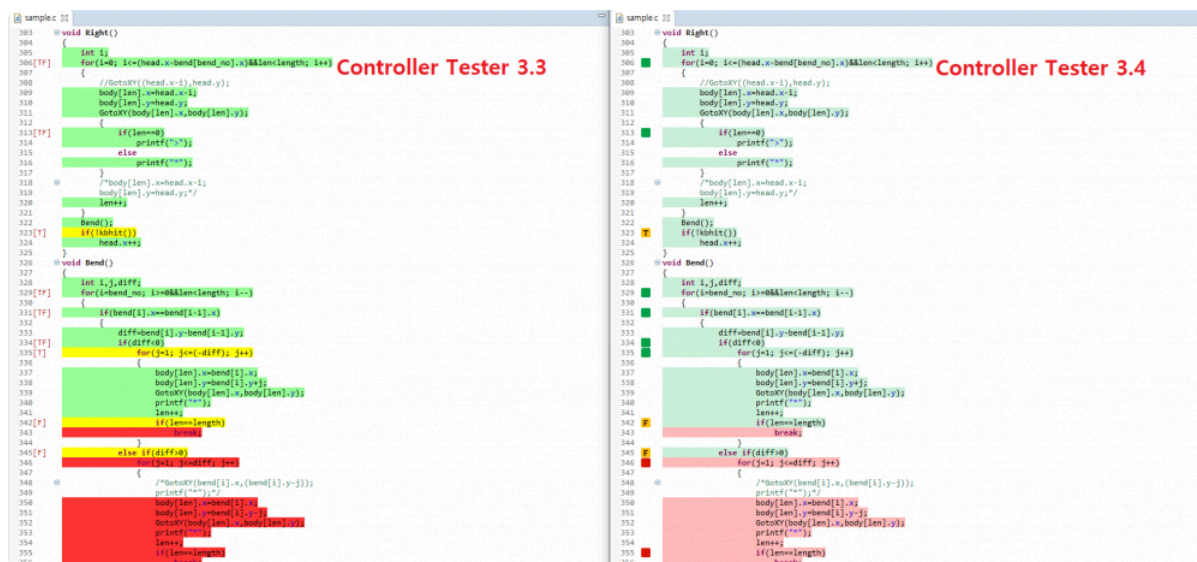
Improvement of C++ test

- Improve test generation engine.
- Improve class factory.

Improvement of search feature in test editor

- Available to search member variable of structure.
- Available to search although starting string doesn't match.

Improvement of coverage display method in source code editor.



Before 3.3	After 3.4	Description
[T]		When the branch is only true-covered
[F]		When the branch is only false-covered
[TF]		When the branch is fully covered
		When the branch is not covered
ex)[TF][T]		When a line contains multiple branches

Others

- Available to use CS_XX_INPUT(OUTPUT) macro in target tests.
- Available to build with command line when Code Composer Studio project is closing.
- CAVM code generation is optional.

2. Added Features

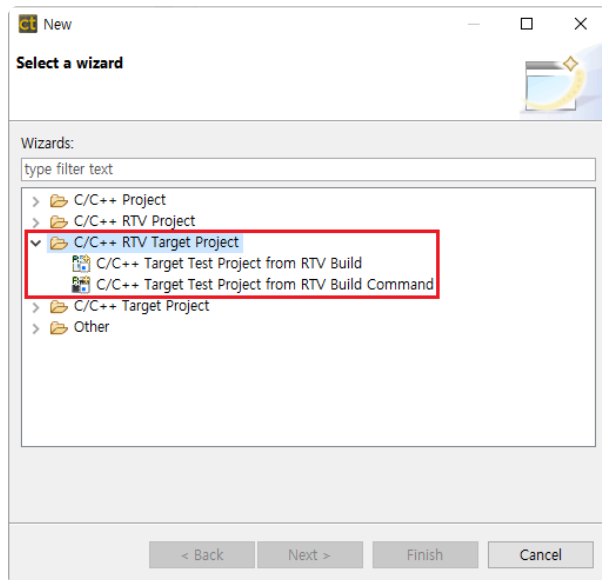
- [Add C/C++ RTV Target Project](#)
- [Add Test Reconfiguration After Source Code Modification](#)
- [Add Feature to Insert Default Values for Empty Input](#)

2.1. Add C/C++ RTV Target Project

Create an RTV project to test the source files that exist in the RTV environment.

Add [C/C++ RTV Target Project] feature

- C/C++ Target Test Project from RTV build
- C/C++ Target Test Project from RTV build command



Add settings for RTV target projects

Create a RTV target project through the following steps.

- RTV server setting
- RTV toolchain setting
- RTV target environment settings
- Create a RTV target project

Create and run RTV target tests

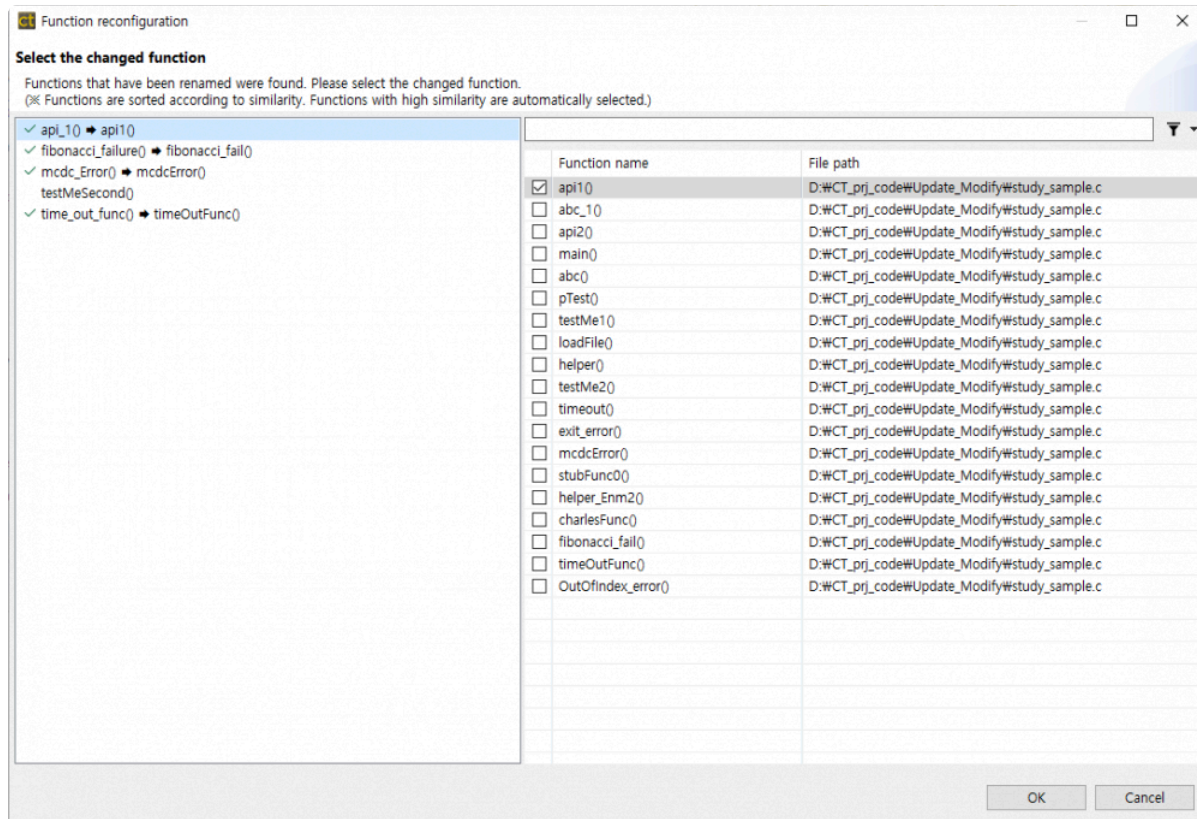
Create unit/integration tests and run host/target tests.

2.2. Add Test Reconfiguration After Source Code Modification

[Test reconfiguration] dialog is displayed when detecting modifications of function name, function return type, number of parameter, or parameter name using integrity check.

Function reconfiguration

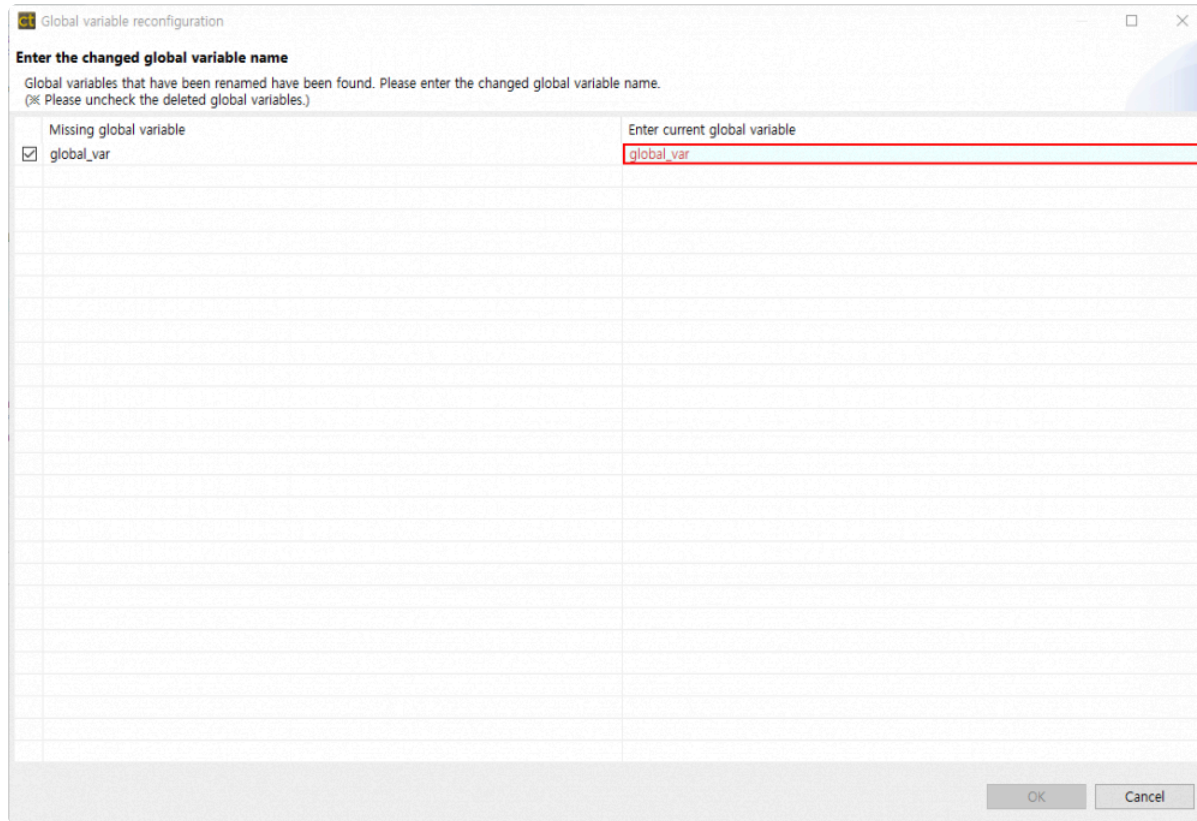
When modifying names of test or stub functions, you can use [Function reconfiguration] feature.



Functions are sorted by similarity of the name and function with high similarity is connected automatically.

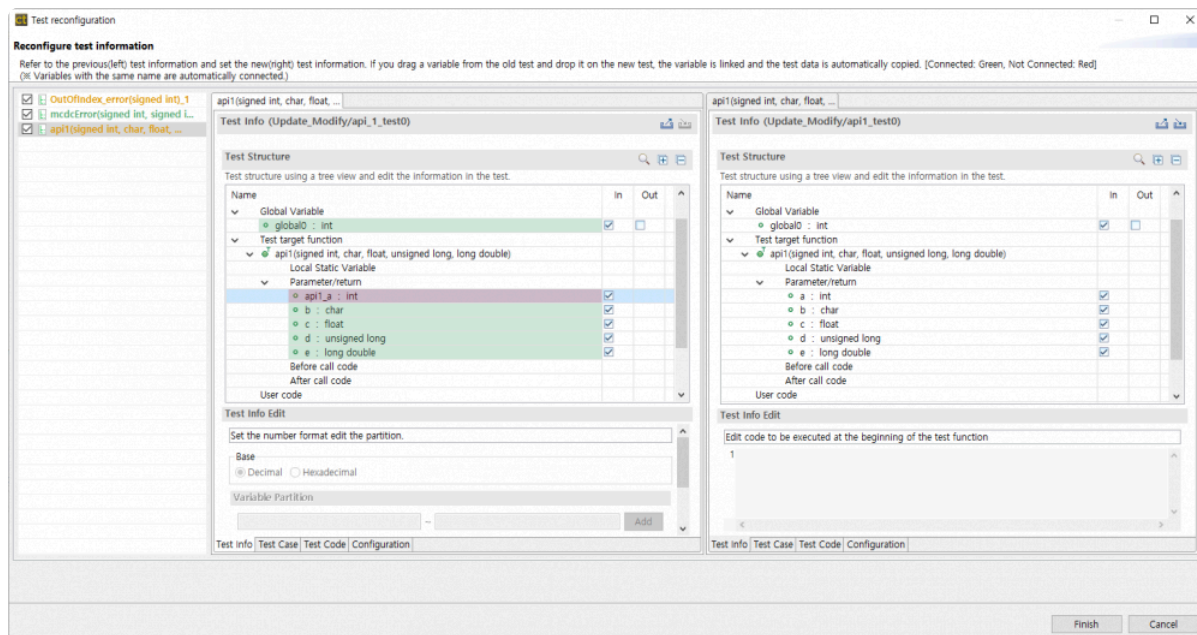
Global variable reconfiguration

When modifying names of global variables used in tests, you can use [Global variable reconfiguration] feature.



Test reconfiguration

When modifying names of global variables used in tests, you can use [Global variable reconfiguration] feature.



You can design a new test based on original test using [Test reconfiguration] feature. When drag a variable of function before modifying and drop to a variable of function after modifying, variables are connected and test data are copied automatically.



You can use [Test reconfiguration] feature manually when right-click the test.

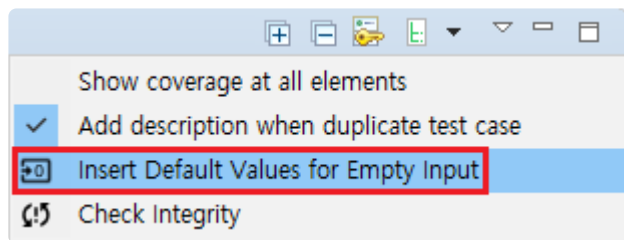
Integrity check

If you modify source code and cancel reconfiguration, you can use manually reconfiguration with [Integrity check] feature in pull-down menu of unit test view.

2.3. Add Feature to Insert Default Values for Empty Input

When input fields of test cases are empty, you can enter default values selecting [Insert Default Values for Empty Input] feature in pull-down menu of test view. Following table is the default values.

Variable type	Default value
string	<i>(empty string)</i>
without string	0



3. Bug Fixed and Feature Modified

- [List of Fixed Bugs](#)
- [List of Modified Features](#)

3.1. List of Fixed Bugs

- Fix an issue occurred when create tests and an error when print stack trace informations.
- Fix a bug that doesn't create automatically *link.mk* file when create a RTV project with CLI.
- Fix an issue of recognizing symbols of host and target stub as duplicate symbols when performing [Export Test Data].
- Fix issues of case-sensitivity, description, and breakpoint occurred in target environment settings of Code Composer Studio.
- Fix a bug where the [Next] button doesn't work when creating an RTV project.
- Fixed an issue where test generation was not possible when the same function was in different TUs.
- Fix an issue that build stub is disabled when running tests.
- Fix a preprocessing logic error for test data of stub symbol.
- Fix an error when converting code related to *asm pragma*.
- Fix an issue that Excel files of each test report is abnormally generated when the test case diagnosis message is too large.
- Fix an issue that files made by [Export Project] in Controller Tester are broken and [Import Project] doesn't work if description of test data contains double quotation marks.
- Fix an issue that CS_OUTPUT macro is generated abnormally for structure pointer members.
- Fix an issue that test isn't run when *entry point* is not *main* using Code Composer Studio script.
- Fix an error that occur when executing [Inspect Debug Info].
- Fix a macro issue to automatically generate test data.

3.2. List of Modified Features

Modify coverage display standard

Modify coverage display standard from branch coverage to statement coverage.

Controller Tester 3.4	Controller Tester 3.3
<pre> 326 void Bend() 327 { 328 int i,j,diff; 329 for(i=bend_no; i>=0&&len<length; i--) 330 { 331 if(bend[i].x==bend[i-1].x) 332 { 333 diff=bend[i].y-bend[i-1].y; 334 if(diff<0) 335 for(j=1; j<=(-diff); j++) 336 { 337 body[len].x=bend[i].x; 338 body[len].y=bend[i].y+j; 339 GotoXY(body[len].x,body[len].y); 340 printf("%*"); 341 len++; 342 if(len==length) 343 break; </pre>	<pre> 326 void Bend() 327 { 328 int i,j,diff; 329 for(i=bend_no; i>=0&&len<length; i--) 330 { 331 if(bend[i].x==bend[i-1].x) 332 { 333 diff=bend[i].y-bend[i-1].y; 334 if(diff<0) 335 for(j=1; j<=(-diff); j++) 336 { 337 body[len].x=bend[i].x; 338 body[len].y=bend[i].y+j; 339 GotoXY(body[len].x,body[len].y); 340 printf("%*"); 341 len++; 342 if(len==length) 343 break; </pre>