



CODESCROLL CONTROLLER TESTER MANUAL

Software for safe world

www.suresofttech.com



User Manual

3.7 — Last update: Jun 09, 2022

Suresofttech

Table of Contents

1.	Before Starting	4
2.	Overview	5
3.	Install	7
4.	Uninstall	. 11
5.	Run	. 12
6.	Set License	. 14
7.	Set a Toolchain (Analyzer)	. 16
	7.1. Auto-registration of Toolchain	. 17
	7.2. Add a Toolchain	. 18
	7.3. Edit a Toolchain	. 21
	7.4. Duplicate a Toolchain	. 22
	7.5. Remove a Toolchain	. 23
	7.6. Export a Toolchain	. 24
	7.7. Import a Toolchain	. 26
8	Create a Project	27
0.	8.1. C/C++ Project with Source Files	
	8.2. C/C++ Project from Embedded	
	8.3. C/C++ Project from Visual Studio Project	
	8.4. C/C++ Project form Existing CodeScroll Project	
	8.5. Create a C/C++ Project with CPI File	
	8.6. Create a C/C++ Project with Build Information	
9.	Create a Test	. 37
40). Test Editor	40
10	10.1. Test Info	
	10.2. Test Case	
	10.2. Test Case	
	10.4. Configuration	
	10.5. Test Macro	
11	. Test Run	. 49
12	2. Inspect Debug Information	. 51
13	B. Generate Function Call Flow Contents	. 53
14	- Properties	56
	14.1. Project Properties	
	14.2. Module Properties	
	14.3. Source File Properties	
	•	

15. Preferences	
15.1. Analysis	71
15.2. Exclusion	77
15.3. Performance	
15.4. Source File Types	79
15.5. Language	80
15.6. Test	82
15.7. Toochain	88
15.8. Editor	89
16. Test Perspective	
16.1. Dashboard	
16.2. Test Navigator View	
16.2.1. Merge Project Coverages	
16.3. Unit Test View	
16.4. Integration Test View	
16.5. Coverage View	
16.6. MC/DC View	128
16.7. Stub View	130
16.8. Class Factory View	
16.8.1. Using class codes	
16.8.2. Using mock objects	
16.9. Control Flow Graph View	
16.10. Call Graph View	
16.11. Error View	
16.12. Debug Information View	
16.13. Fault Injection View	
16.14. Input/output Data Graph View	
16.15. Console View	
16.16. Requirement View	
17. Analysis Perspective	178
17.1. Metrics View	
17.2. Metrics Chart View	
17.3. Metrics Top Chart View	
17.4. Metrics Bar Chart View	
17.5. Metrics Diagnosis Chart View	
17.6. Unused Function View	
17.7. Source-Header Relation View	
17.8. Global Variable Relation View	
17.9. Function Call Hierarchy View	
17.10. Source Code Editor section	
18. File Menu	
18.1. Create a Module	
18.2. Switch Workspace	
18.3. Import	198

	18.3.1. Requirement – Import from CSV file	200
	18.3.2. Requirement – Import from Polarion	202
	18.3.3. Requirement – Import from V-SPICE	
	18.3.4. General – File System	208
	18.3.5. General – Import project	211
	18.3.6. Coverage – Import Coverage	214
	18.3.7. Test – Import test	216
	18.3.8. Test – Import test using test code file	220
	18.3.9. Preferences - Virtual Memory Address	222
	18.3.10. Preferences - ToolChain	224
	18.4. Export	226
	18.4.1. Metrics - Metrics	228
	18.4.2. Requirement – Export test results connected to requirements to Polarion	230
	18.4.3. Requirement – Export test results to V-SPICE	232
	18.4.4. Requirement - Requirement test coverage report	234
	18.4.5. General - Export project	236
	18.4.6. Coverage – Export Coverage	238
	18.4.7. Test - Export test	240
	18.4.8. Test - Generate test report	243
	18.4.9. Preferences - Virtual Memory Address	246
	18.4.10. Preferences - ToolChain	248
10	Edit Menu	2/0
15.		245
20	Search Menu	250
		200
	Project Menu	
21.	Project Menu	252
21.		252
21. 22.	Project Menu	252 254
21. 22. 23.	Project Menu Window Menu Help Menu	252 254 255
21. 22. 23. 24.	Project Menu Window Menu Help Menu Troubleshooting	252 254 255 260
21. 22. 23. 24.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in	252 254 255 260 262
21. 22. 23. 24.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in 25.1. Target Environment Settings	252 254 255 260 262 263
21. 22. 23. 24.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in 25.1. Target Environment Settings 25.2. C/C++ Target Test Project.	252 254 255 260 263 268
21. 22. 23. 24.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in 25.1. Target Environment Settings 25.2. C/C++ Target Test Project. 25.3. Target Test Run Settings.	252 254 255 260 263 263 268 273
21. 22. 23. 24.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in 25.1. Target Environment Settings 25.2. C/C++ Target Test Project. 25.3. Target Test Run Settings. 25.4. Target Log Collector.	252 254 255 260 263 268 273 274
21. 22. 23. 24.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in 25.1. Target Environment Settings 25.2. C/C++ Target Test Project. 25.3. Target Test Run Settings.	252 254 255 260 263 268 273 274
21. 22. 23. 24. 25.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in 25.1. Target Environment Settings 25.2. C/C++ Target Test Project. 25.3. Target Test Run Settings. 25.4. Target Log Collector.	252 254 255 260 263 263 268 273 274 276
21. 22. 23. 24. 25.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in 25.1. Target Environment Settings 25.2. C/C++ Target Test Project 25.3. Target Test Run Settings 25.4. Target Log Collector 25.5. Preferences	252 254 255 260 263 263 268 273 274 276 278
21. 22. 23. 24. 25.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in 25.1. Target Environment Settings 25.2. C/C++ Target Test Project. 25.3. Target Test Run Settings. 25.4. Target Log Collector. 25.5. Preferences CONTROLLER TESTER RTV(Remote Target Verifier)	252 254 255 260 263 268 273 274 276 278 279
21. 22. 23. 24. 25.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in 25.1. Target Environment Settings 25.2. C/C++ Target Test Project 25.3. Target Test Run Settings 25.4. Target Log Collector 25.5. Preferences CONTROLLER TESTER RTV(Remote Target Verifier) 26.1. RTV Server Settings	252 254 255 260 263 263 263 273 274 276 278 279 283
21. 22. 23. 24. 25.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in 25.1. Target Environment Settings 25.2. C/C++ Target Test Project. 25.3. Target Test Run Settings. 25.4. Target Log Collector. 25.5. Preferences. CONTROLLER TESTER RTV(Remote Target Verifier) 26.1. RTV Server Settings. 26.2. RTV Toolchain Settings.	252 254 255 260 263 268 273 274 276 278 279 283 286
21. 22. 23. 24. 25.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in 25.1. Target Environment Settings 25.2. C/C++ Target Test Project 25.3. Target Test Run Settings 25.4. Target Log Collector 25.5. Preferences CONTROLLER TESTER RTV(Remote Target Verifier) 26.1. RTV Server Settings 26.2. RTV Toolchain Settings 26.3. Create a RTV Project	252 254 255 260 263 263 263 273 274 276 277 278 279 283 286 291
21. 22. 23. 24. 25.	Project Menu Window Menu Help Menu Troubleshooting CONTROLLER TESTER Target Plug-in 25.1. Target Environment Settings 25.2. C/C++ Target Test Project 25.3. Target Test Run Settings 25.4. Target Log Collector 25.5. Preferences CONTROLLER TESTER RTV(Remote Target Verifier) 26.1. RTV Server Settings 26.2. RTV Toolchain Settings 26.3. Create a RTV Project 26.4. Refresh/Add RTV Source Files	252 254 255 260 263 268 273 274 276 277 279 283 286 291 295

27.	. CONTROLLER TESTER RTV(Remote Target Verifier) Target Plug-in	300
	27.1. RTV Server Settings	301
	27.2. RTV Toolchain Settings	
	27.3. RTV Target Environment Settings	303
	27.4. Create a RTV Target Project	306
	27.5. Refresh/Add RTV Target Source Files	312
	27.6. RTV Target Project Information	313
	27.7. Run RTV Target Test	315
28.	. DISCOVERY Plug-in	316
	28.1. DISCOVERY Install	317
	28.2. DISCOVERY Execution	319
29.	. CONTROLLER TESTER CLI(Command Line Interface)	321
	29.1. CLI Install	322
	29.2. Project Import/Export	324
	29.3. CLI Settings	
	29.4. CLI Execution	332
30.	. EULA(End-User License Agreement)	334

1. Before Starting

Purpose of Document

This document provides the information on how to use the CodeScroll Controller Tester Product.

Some images contained in this document may vary slightly depending on the version, but the image corresponding to the feature described is the same as the version specified in the documentation.

2. Overview

Introduction

Controller Tester is a test automation solution for unit and integration tests of software developed and executed in various environments.

Toolchain supported in Controller Tester

Controller Tester supports about 100 toolchain. For more information, please contact the Technical support contact at the bottom of this document <u>Troubleshooting</u> page.

Features

Controller Tester is a source code-based software test automation tool for unit and integration tests and the major features are as follows.

Test design

- 1. Automatically create tests and test data
- 2. Intuitive test design interface
- 3. Ability to import test data in various format
- 4. Ability to design of integration test for complex nested structures

Achieving coverage goals

- 1. Supports statement, branch, MC/DC and function call coverage
- 2. Provides guides to achieve the goal of MC/DC
- 3. Checks the coverage results by linking the control flow graph with the source code
- 4. The coverage goal can be achieved quickly by testing only uncovered section after measuring the system coverage (QualityScroll COVER product).

Others

- 1. Supports the simulation and the actual target environment tests
- 2. Links with DevOps(issue management + continuous integration + configuration management) tools
- 3. Shares project settings with CodeScroll product family: By carrying out the project settings only once, coding rule inspection + runtime error detection + unit/integration test can be possible.
- 4. Reports in various formats
- 5. Interaction with requirements management tools to support traceability of requirements to tests and results

Standard certification

- 1. IEC 61508-3 (Electrical/Electronic)
- 2. ISO 26262-8 (Automotive)
- 3. IEC 60880 (Nuclear)

- 4. IEC 62279 / EN 50128 (Railway)
- 5. DO-178C / DO-330 (Aviation)
- 6. IEC 62304 (Medical device)

Overall screen of Controller Tester

It provides high DOF (degrees of freedom) windows made by Eclipse RCP. User can set them in [Window] menu and each view.

	itroller Tester RTV(Remote Target Verifier) + Target Plugin Project Target Run Window Help					- 🗆 X
zlib				Target 👱 💽 Glo	bal Search (Ctrl+3) bh 📲 Call Graph 🔓 Functi	्रों 😰 🖃 🗟 ion Call Hierarchy 🔭 Error
C trees.c ≥3 885 885 887 888 899 894 895 895 895 895 896 897 895 896 897 898 899 906[TF] 902 903 904 905 S 905 S	<pre>vorage vorage * Flush the bits in the bit buffer to pending output (leaves at // void 2LIB_INTERNAL_tr_flush_bits(s) deflate_state *s; { bi_flush(s); } /* * This takes 10 bits, of which 7 may remain in the bit buffer. */ void 2LIB_INTERNAL_tr_flush_bits(s) deflate_state *s; { send_bits(s, STATIC_TREESccl, 3); send_code(s, END_BLOCK, static_ltree); #ifdef DEBUG *->compressed_len += 10L; /* 3 for block type, 7 for EOB */ PodiFfce Code Editor * Determine thSec:Lioonne current block: dynamic tre * trees or store, and output the encoded block to the zip file. */ void 2LIB_INTERNAL_tr_flush_block(s, buf, stored_len, last) deflate_state *s; deflate_state *s; */ void 2LIB_INTERNAL_tr_flush_block(s, buf, stored_len, last) deflate_state *s; */ void 2LIB_INTERNAL_tr_flush_block(s, buf, stored_len, last) deflate_state *s; */ */ * deflate_state *s; */ * one if this is the last block for a file */ */ * Build the Huffman trees unless a stored block is forced *, ** ** trees void */ ** ** ** ** ** ** ** ** ** ** ** ** *</pre>	Coverage — _ w Unit Test Run ► type file, function, Name > ○ _tr_flus > ○ _tr_flus > ○ _tr_flus > ○ _tr_tstor > ○ _tr_tstor - ○ _tr_tstor > ○ _tr_tstor - ○ _t	Integration Test Integration	(809 (809 (809 (809) (80) (80	(0 / 286) 1095 (1 / 0 / 7) 8 (1 / 0 / 7) 8 (1 / 0 / 7) 8 (0 / 0 / 8) 8 (0 / 0 / 8) 8 (0 / 0 / 8) 8 (0 / 0 / 8) 8 (0 / 0 / 3) 3 (9 / 0 / 4) 13 (5 / 0 / 0) 5 (5 / 0 / 0) 5 (5 / 0 / 0) 5 (11 / 0 / 0) 11 (5 / 0 / 6) 11 o Edit	In Call Hierarchy Error Statement Coverage 39.1% (1751/4467) Coverage 71.4% (15/21) 100.0% (1/1) 100.0% (1/1) 100.0% (1/1) 100.0% (1/1) 100.0% (1/1) 100.0% (1/1) 100.0% (1/2) 100.0% (1/
927 928 929	s->strm->data_type = detect_data_type(s); } <	Test Info Test Case	Test Code Configuration	108M of 248M	Ū :	

3. Install

Installation requirements

OS	Microsoft Windows 7/10 (64bit)
RAM	512MB or more
HDD	Free space about 1GB (GCC Compiler installed) (When carrying out tests actually, HDD usage may be increased due to the test results.)

Install Controller Tester for Windows

By using the install package, you can install Controller Tester as follows:

1. Execute controller-tester_3.x_xxx.exe file.

Name	Date modified	Туре	Size	
controller-tester_3.6_x64.exe	2021-11-09 오후 3:28	Application	1,010,902	

2. Click [Next] after the installation wizard runs and gathers installation information.

CodeScroll Controller Tester Setup	×
CODESCROLL CONTROLLER TESTER	
Welcome to the CodeScroll Controller Tester Setup Wizard	
The Setup Wizard will install CodeScroll Controller Tester on your computer. Click Next to continue or close the window to exit the Setup Wizard.	
Next >	

3. Accept the end-user license and click [Next].

deScroll Controller Tester Setup	
Read the Suresoft Technologies Incs.	. License Terms
To continue you must read and accept the terr want to accept the Suresoft Technologies Incs	
Software End User License Agreement	^
Important: This End User License Agreement ("EULA") Suresoft Technologies, Inc. producing software product id Software product includes computer software, related me electronic) manual. When installing, copying, downloadin product, user must accept the conditions in the EULA, othe license, user cannot install, copy, download, backup, and a	dentified by product cognition card/label. edia, related documents and "online" (or g, backing up, or accessing software erwise license cannot be issued. Without
Software Product License Software product is protected by intellectual property right international copyright regulations. Software product licens term "computer" used in this agreement means a single of described below.	se is registered but cannot be sold. The
✓ I <u>a</u> ccept the terms of this agreement,	
	< Back Next >

4. Set the path to install Controller Tester and click [Next].

CodeScroll Controller Tester Setup			_		
Choose a file location	ı				
This is the folder where C	odeScroll Controller Tester will	be installed			
	Duescroll Controller Tester will	be installeu.			
To install in this folder, clic click "Browse".	:k "Next". To install to a different	folder, enter	it belo	ow or	
C:₩Program Files₩Suresoft	₩CodeScroll Controller Tester 3,6t	* ~	Brows	e	
		< Back	Next	t>	

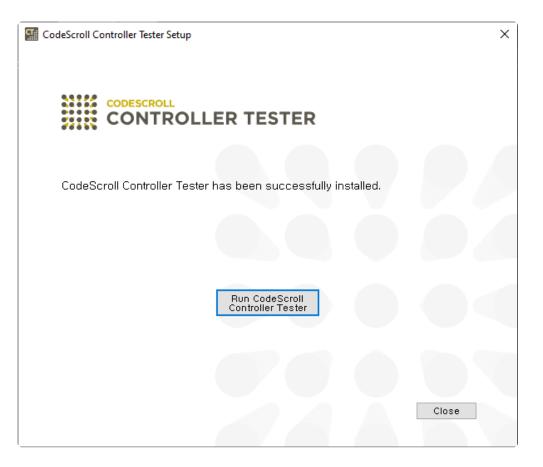
5. All required information for the installation has been collected. Click [Install].

I CodeScroll Controller Tester Setup	>
Begin installation of CodeScroll Controller Tester	
The Setup Wizard is ready to begin the CodeScroll Controller Tester installation	
Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Close the window to exit the wizard.	
< Back]

6. Install Controller Tester.

Co	odeScroll Controller Tester Setup	×
	Installation Progress	
	Installing CodeScroll Controller Tester	
	Please wait while the Setup Wizard installs CodeScroll Controller Tester. This may take several minutes.	
	Extracting files from archive	

7. Click [Run CodeScroll Controller Tester] to run Controller Tester immediately after installation is complete, and click [Finish] to end the installation.



4. Uninstall

Uninstall Controller Tester

How to uninstall Controller Tester is as follows.

1. Uninstall using [Control Panel] > [Uninstall a program]



"Uninstall Controller Tester" removes the files necessary to operate the Controller Tester. The workspaces set by users are not removed.

The procedures for [Uninstall a program]

1. Click [Control Panel] > [Uninstall a program].

Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Organize 🔻 Uninstall	
Name	Publisher
CodeScroll Controller Te	Suresoft Technologies Incs.
Offinistan	

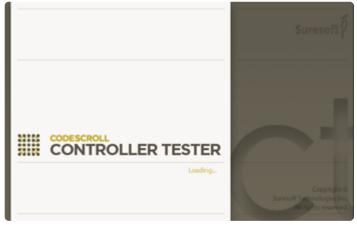
2. Click [Yes] to uninstall CodeScroll Controller Tester



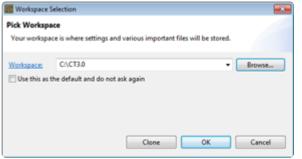
3. All the necessary information for uninstalling has been collected. When the Controller Tester is successfully uninstalled, the window closes.

5. Run

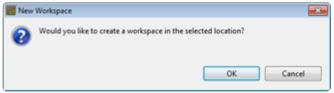
Select the shortcut icon or [Start] -> [All programs] -> [CodeScroll Controller Tester 3.x] -> [CodeScroll Controller Tester 3.x].



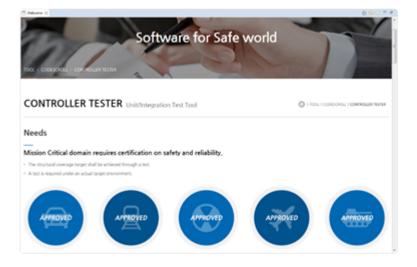
Select the workspace (working space) following the splash screen of the Controller Tester. You can select either the already existed directory or enter a new directory as the workspace path. You can also use the currently selected workspace as the default so that you do not need to reselect the workspace the next time you run again. When the path has been set, click the [OK] button.



If the selected directory does not exist, confirm the user whether to create the directory.



Create a directory in the workspace you have set and the welcome page of the tool and [Create a Project] wizard is displayed. The welcome page is displayed only when creating a workspace newly, and is not displayed when selecting the workspace created previously.



Close the welcome page and the product screen is displayed.

6. Set License

To use Controller Tester, you must register the license first. The procedures for registering the license are as follows.

1. Select [Help] > [License] in Menu.



2. Select [Product] > [Controller Tester] and click [Edit].

License		-*-
Product Controller Tester	Invalid license.	
	Edit	
	ОК	

3. Enter the provided license in [Editing License Information] and click [OK] button to register the license.

Edit license inform	nation 💿 💌
 License Information Floating Please enter floating 	n g license server information.
Operating system:	[Window 👻]
IP/hostname:	
Port:	
	Authentication connection
Nodelock	
Please enter nodelo (License file name :	ock license file path. ct.lic)
File path:	Browse
	OK Cancel

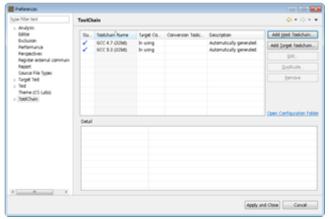
- Floating: Register it through the license server. Enter the server information (operating system, IP, port) and click [Authentication connection] button.
- Nodelock: Register the license file provided.

7. Set a Toolchain (Analyzer)

To perform the test using Controller Tester, the toolchain setting is required.

To create a project, you must have a toolchain(compiler information) of the sources to test.

The toolchain can be set in [Window] -> [Preferences] -> [Toolchain].



And it can also be set through [Toolchain setting] in Create a source file C/C++ project wizard or in Create a C/C++ Project from Embedded wizard.

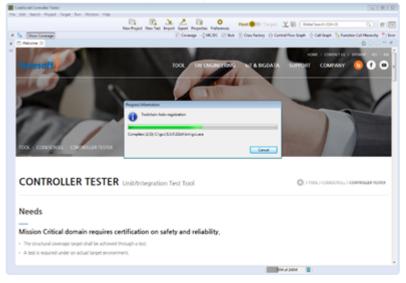
🔣 New Proje	ect	- • • •
New Projec	t	
Create new	project	
Project nam		
Location:	C:\CT3.0	
Select Tool		
Default	Toolchain Name	Description
	GCC 4.7 (32bit)	Automatically generated.
	GCC 5.3 (32bit)	Automatically generated.
		Toolchain Setting
		Toorstrain searchy
	< Back	Next > Finish Cancel

The functions related to the toolchain setting provided by Controller Tester are as follows.

- Auto-registration of Toolchain
- Add a Toolchain
- Edit a Toolchain
- Duplicate a Toolchain
- <u>Remove a Toolchain</u>
- Export a Toolchain
- Import a Toolchain

7.1. Auto-registration of Toolchain

When executing the Controller Tester, it extracts the information of Visual Studio compiler installed in user PC and the information GCC compiler installed along with the Controller Tester and registers the toolchain automatically (the registered toolchain is not registered again.).



The auto-registered toolchain can be checked in [Window] -> [Preferences] -> [Toolchain] window, and it displays the message "Automatically generated." in the description.

Preferences						
type filter text	TeelO	halin				⇔ • ⇔ • •
Analysis Editor Bechanion Performance Perpectives Register extenuit commance Report Source File Types > Test Theree (C5 Leftel) > TestChain	Sta	Testchań Name GCC 43 (32kk) GCC 53 (32kk)	Target Co In using In using	Conversion Took	Description Automatically generated. Automatically generated	Cpen.Certiguation Faile
					Apply at	d Close Cancel

7.2. Add a Toolchain

1. In [Window] -> [Preferences] -> [Toolchain] window, click the [Add Toolchain] button.

	tatus	Toolchain Name	Target Compi	Conversion Toolchain	Description	Add Toolchain
sion Jage						Add RTV Toolcha
mance						Edit
ectives e File Types						Duplicate
e (CS Labs) hain						Remove
De	etail					Open Configuration

2. In [Add a toolchain] window, enter the toolchain information.

Edit Toolchain			×
Toolchain(Method Manage)			
Manage toolchain information.			
Toolchain Information			
Name: GCC 5.3 (32bit)			
Description:			
Env Script: C:#Users#x#AppData#Roaming#CodeScroll#1.1#parserCor	nfig₩GCC 5.	. <u>B</u> rowse	
Configuration: ▶ GNU Compilers ▶ gcc ▶ 5.3 ▶ others			
C C++		1	
Compiler: C:#gcc#5.3.0#32bit#bin#gcc.exe		<u>B</u> rowse	
Linker: C:#gcc#5.3.0#32bit#bin#gcc.exe		<u>B</u> rowse	
Archiver: C:\#gcc\#5.3.0\#32bit\#bin\#ar.exe		Browse	
System Header:		_	
c:#gcc#5.3.0#32bit#lib#gcc#mingw32#5.3.0#include c:#gcc#5.3.0#32bit#include		<u>A</u> dd	
c:\gcc\s.3.0\s2bit\lib\gcc\mingw32\s3.0\linclude-fixed		<u>E</u> dit	
		<u>R</u> emov	e
		<u>U</u> p	
		Down	
Libraries:			
		<u>A</u> dd	
		<u>E</u> dit	
		<u>R</u> emov	e
		2	_
Automatically extract toolchain information from entered compiler.	Extract T	Toolchain I	nfo
< <u>Back</u> <u>Next</u> > <u>Fini</u>	sn	Cancel	

- When you click [Extract toolchain Info]² button after [Compiler]¹, the toolchain information is extracted automatically from the compiler entered.
- In [Compiler], enter the compiler path.

- If it is based on Visual Studio: C:\program Files (x86)\Microsoft Visual Studio 10.0\VS\bin\cl.exe
- If it is based on GCC:
- C:\MinGW\bin\gcc.exe
- CodeWarrior 5.x or higher for Freeescale HC (s) 12:
 C:\Program Files (x86)\Freescale\CWS12v5.1\Prog\chc12.exe
- 3. In [Name], enter the name of toolchain to be created.
- 4. In [Description], enter the description of toolchain to be created.
- 5. The entry information of [Env Script], [Configuration] and [C/C++] is set automatically by extracting the information from the compiler or can be entered directly by user.
- 6. After entering the toolchain information, click the [Next] or [Finish] button.

Add a toolchain	— 🗆 X
dvanced Compile setting	
Set the parser configuration. (Depending on the toolchain, some options apply differently)	
C C++	
Keyword Directive Define Predefined Type etc.	
Keywords:	
	Add
	Edit
	Remove
ASM Keywords:	
	Add
	Edit
	Remove
	Remore
Example:	
	^
< Back Next > Finis	sh Cancel

7. In [Advanced Compile setting] window displayed when clicking the [Next] button, the detailed items related to the configurations selected in the [Configuration] are shown. Each item can be changed by the user. The application of the changed items can be checked in [Example].

For the detailed setting procedure for the other toolchain detail, please contact the Technical support contact at the bottom of this document <u>Troubleshooting</u> page.

7.3. Edit a Toolchain

1. Select the toolchain to be edited and click the [Edit] button.

Edit Toolcha	in						×
olchain(Met	hod Manage)						
Aanage toolch	ain information.						
Toolchain Info	rmation						
Name:	GCC 5.3 (32bit)						
Description:							
Env Script:	C:#Users#x#Ap	opData\Roaming	g#CodeScroll#1.1#	+parserConfi	g#GCC 5.:	Brow	se
Configuration:	GNU Compi	lers ► gcc ►	5.3 > others				
C C++							
	¥gcc₩5.3.0₩32bi	t₩bin₩gcc.exe				Brow	se
Linker: C:	+gcc₩5.3.0₩32bi	t#bin#gcc.exe				Brow	se
Archiver: C:	- ¥gcc₩5.3.0₩32bi	t#bin#ar.exe				Brow	se
System Heade	-					-	
	₩32bit₩lib₩gcc¥	4mingw32#5.3.0)#include		_	Add	
	₩32bit₩include ₩32bit₩lib₩gcc¥	¥mingw32₩5.3.0)Winclude-fixed			Edit	
						Remo	ove
						Up	
						<u>D</u> ov	'n
Libraries:							
						Add	
						<u>E</u> dit	
						Remo	ove
utomatically e	ktract toolchain ii	nformation from	entered compiler.		Extract To	olchair	Info
		< <u>B</u> ack		Finish			
			Next >			Cance	

- 2. Edit the information of toolchain.
- 3. Click the [Finish] button.

7.4. Duplicate a Toolchain

- Select the toolchain to be copied and click the [Duplicate] button. (It can copy only a toolchain which a compiler is not entered.)
- 2. Edit the toolchain information that you want to change.
- 3. Click the [Finish] button.

7.5. Remove a Toolchain

Select the toolchain to remove and click the [Remove] button.

If there is a project using the toolchain you want to remove, the project will not work properly.

7.6. Export a Toolchain

The toolchain information can be exported via 'Export' function.

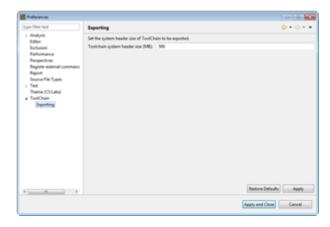
1. Click [File] > [Export] in menu and select [Toolchain] in [Preferences].

	New		
	Close Close All	Chrl+W Chrl+Shith+W	
	Save Save All Rename	Levet Colors aport vized.	
2 12	Refresh Switch Workspace Import Export	Select an expert wicard. Type filter text	
	Properties Exit	s bo Coverage > Do Coverage > Do Metrics > Do Metrics - Do Partementes - ToroCovera	
		> 🐎 Test	
		< Back Not > Finish Cancel	

 Select the toolchain to be exported and click [Browse] button to specify the path to be exported. If you set the system header file in the toolchain settings, a checkbox will be displayed in the [Exporting system header] column, where you can choose whether to export or not.

vailable	Toolchain List:			
Default	Toolchain Name	Description	Exporting system header directory	Exporting system header
2	GCC 4.7 (32bit)	Automatically		
	GCC 5.3 (32bit)	Automatically	clgccl53.032bit/binl_18blgcclming_	
ielect Al	Deselect All			
	system header size: 0 / 1	500 (ME)		
elected t	he export location:			Browse

 Click [Window] > [Preferences] > [Toolchain] > [Exporting] in menu to set the system header size to be exported. If the size of the system header is larger than the set value, the system header cannot be exported.



7.7. Import a Toolchain

The exported toolchain can be imported via the [Import] function.

1. Select [File] > [Import] in menu and click [Toolchain] in [Preferences].

File Edit Search	Project Run Window	Ergest.
New		Select Import TootOuin configuration informations.
Close	Ctrl+W	Select as import using
Close All	Ctrl+Shift+W	(speller ted
Save	Ctrl+S	2 De Conseque 2 De General 4 De Federances
Save All	Ctrl+Shift+S	E, ToolChain
Rename	F2	
Refresh	F5	
Switch Works	pace	
Import		
i Export		
Properties	Alt+Enter	
Exit		clast Red a

2. Select the file to import and check the toolchain to import when the toolchain of the file is displayed.

	toolchain and destination	location. Ivagrant Desktogi toolchairi toolchai	a lub	Brouse.
dect the	toolchain to import			
Default	Tookhain Name	Description	Status in import	
1	GCC 4.7 (32bH)	Automatically generated.	Need to configure	
0	GCC 5.3 (IZH4)	Automatically generated.	Need to configure	
	8. Deselect All	ther micard in finished.		

The name of the toolchain to import is duplicated with the existing toolchain name, the status of 'import' becomes "Need to configure". The name of the toolchain to import can be modified in [Select how to resolve] dialog shown when clicking the [Next] button.

8. Create a Project

Checklist before creating a project

Since Controller Tester is a tool that detects errors by actually executing source codes, the source code under test must be executable. In other words, the source code under test must be buildable for normal testing.

Before creating and testing a project, you need to set up a 'toolchain' that takes into account the development environment of the target software. Please refer to <u>Set a toolchain</u> for details.

Controller Tester automatically generates a stub if the target object to link does not exist during the build process. However, the automatic test stub generation feature may have limitations in performing tests, so it is recommended to prepare an appropriate test strategy in advance.

Also, the name of the project cannot be more than 100 characters.

Create a project

To create a Controller Tester project, execute [Create a Project wizard].

Select [File] -> [New] or click the shortcut icon.

File	Edit Search Project	Target Window	He	۹þ	
	New	>	C	C/C++ Project with Source Files	
	Close	Ctrl+W		C/C++ Project from Embedded(CodeWarrior, Green Hills, NEC)	
	Close All	Ctrl+Shift+W	Ø	C/C++ Project from Visual Studio Project	
			đ	C/C++ Project from Existing CodeScroll Project	
	Save	Ctrl+S		Create a C/C++ project with CPI File	
	Save All	Ctrl+Shift+S	1	Create a C/C++ project with Build Information	
	Rename	F2	-	Other	Ctrl+N
	Refresh	F5	_	o anona	Carrie

The following types of project creation wizards are provided.

• <u>C/C++ Project with Source Files</u>

Creates a project by selecting the source file directly.

- <u>C/C++ Project from Embedded</u> Creates a project by using an embedded project file.
- <u>C/C++ Project from Visual Studio Project</u>
 Creates a project by using Visual Studio dsw, sln, vcxproj and vcproj files.
- <u>C/C++ Project form Existing CodeScroll Project</u>
 Creates a project by using the CodeScroll project created previously.
- <u>Creates a C/C++ Project with CPI File</u>
 Creates a project by using project creation information located in a CPI file.
- <u>Creates a C/C++ Project with Build Information</u>
 Creates a project by using a makefile and build script.

8.1. C/C++ Project with Source Files

Create a project with C/C ++ source code files. The user needs to enter the necessary information to build the source code.

1. Enter the project name in [Project name].

Project nam	e	
Location:	C:\CT3.0	
Select Tool		
Default	Toolchain Name	Description
	GCC 4.7 (32bit)	Automatically generated.
	GCC 5.3 (32bit)	Automatically generated.
		Toolchain Sett

- 2. In [Select Toolchain], select the toolchain to use in the project. If there is no created toolchain, create a toolchain through [Toolchain Setting]. Please refer to <u>Set a Toolchain</u> for details.
- 3. Click the [Next] button to move to the next screen. If you click the [Finish] button, an empty project containing no source file is created.

New Project New Project Choose source files Top Directory: P:\#Example	ple_Project₩c₩m	nam₩zlib-1.	2.8	_		×
Directory Image: Settings Image: Seting Image:	ource files	File type E<	filter text adler32.c compress.c crc32.c gzclose.c gzlib.c gzread.c gzwrite.c infback.c inflate.c inflate.c		Brow Rem	
	< <u>B</u> ack	<u>N</u> ext >	<u> </u>		Cance	1

- You can select the source file directly, or through a text file with a file path.
- 4. Select source files directly.
 - a. Click [Browse ...] to specify the [Top Directory] to be displayed in the directory screen below.
 - Except in special cases, make the directory one level higher than the directory containing the source files you want to select as the top-level directory.
 - b. On the [Directory] screen shown on the left, select the directory containing the source files to be used for project creation.
 - c. The files in the selected directory appear on the [File] screen shown on the right. Check the files to add.
- 5. Select by using a text file with the file paths.
 - a. On the [Text file written list of source files] screen shown below, click [Browse ...] to select a text file with a list of source files(absolute file path).
 - b. Clicking the [Remove] button deselects the selected text file and the source file selected through the file.
- 6. After all settings are completed, click the [Finish] button to create the project.

8.2. C/C++ Project from Embedded

Create a project with an embedded project file(.xml, .gpj, .prw, .prj). Except in special cases, the information required for the build is automatically entered, so there is nothing to specify by the user.

New Project			×					
New Project								
Create new project Hills MULTI project	from CodeWarrior project file(* file(*.gpj)	.xml), Green						
Project name:								
Location:	P:\Product_Workspace\ct_3_	1_02						
Select Toolchain								
Default Toold	hain Name	Description						
CPP_	TI_TMS320_6000_SoC_Unit							
GCC	4.7 (32bit)							
GCC	5.3 (32bit)							
Micro	osoft Visual Studio 2010 (32							
Micro	osoft Visual Studio 2010 (32							
Micro	osoft Visual Studio 2015 (32							
Micro	osoft Visual Studio 2015 (32							
gcc5								
iar iar								
			т	oolchain	Settina			
Import Setting			-					
Embedded Project I	File:			<u>B</u> rov	/se			
	< Back Next > Finish Cancel							

- 1. Enter a project name in [Project name].
- 2. Choose a toolchain that fits the embedded project you select.
- 3. In [Import Setting] -> [Embedded Project File], specify the embedded project file of the software under test.
- 4. After completing all settings, click [Finish] to create the project.

8.3. C/C++ Project from Visual Studio Project

Create a project with a Microsoft Visual Studio project file(.dsw, .sln, .vcxproj, .vcproj). Except in special cases, the information required for the build is automatically entered, so there is nothing to specify by the user.

New Project		_		×
New Project				
Create C/C++ Proj .vcxproj, .vcproj)	ect from Visual Studio Project. (.dsw, .sln,			
Project name:	sample			
Location:	P:#Product_Workspace#ct_3_1_02			
Import Setting				
Visual Studio Proje	ect: P:\Example_Project\c\nvip\cpfe\cpfe.sln		Brov	vse)
● Advanced				
	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish		Cance	į

- 1. Enter the project name in [Project name].
- 2. Specify the Visual Studio project file to import in [Visual Studio Project].
- 3. If there are values that must be set before calling the compiler of the toolchain to be registered, add a script file to set the values to [Advanced]-> [Environment Script File].
- 4. Click [Next] to move to the next screen. When you click Done, module settings are randomly selected for the imported Visual Studio project.

New Projec	t		– 🗆 X			
New Project						
Create C/C++ *.vcxproj, *.vc		dio Project. (*.dsw, *.sln,				
Select Active N	Iodules:					
Active	Module misc\mk_errinfo	Configuration (Click to Debug Win32	Edit)			
	src₩disp₩disp src₩fe	Debug Win32 Debug Win32				
	accient All		Cat All Configurations			
Select All Deselect All Set All Configurations						
	< <u>B</u> ack	<u>N</u> ext > <u>F</u> in	ish Cancel			

- 5. Select the module to be activated from the modules included in the Visual Studio project.
- 6. Select the configuration of each module.
 - [Set All Configurations ...] allows you to change the configuration for all modules in a batch.
- 7. After all settings are completed, click the [Finish] button to create the project.

8.4. C/C++ Project form Existing CodeScroll Project

Import an existing CodeScroll project and create a new project with that information. You can easily create a new project with the same configuration(source file, exclude/include analysis, compile flags) as the existing project.

New Project				_		×
Import Projects If the project of of the project.	the same name e	exist, the '_copy' i	s added after tl	ne nar	me	
Select roo <u>t</u> directory Projects:		/orkspace₩ct_3_1		× (B <u>r</u> owse	
 ☑ [zlib_build] ☑ [zlib_build] ☑ [zlib_build] ☑ [zlib_build_] ☑ [zlib_build_] ☑ [zlib_build_] ☑ cpfe (P:₩Pr ☑ stub_sample ☑ target (P:₩F 	example (P:\Proc minigzip (P:\Proc project]NotLinked project]example (project]minigzip oduct_Workspace e (P:\Product_Workspace	duct_Workspace# duct_Workspace# d (P:#Product_Work (P:#Product_Work (P:#Product_Work (P:#Product_Work e#ct_3_1_02#cpfe orkspace#ct_3_1_ ce#ct_3_1_02#cla #cct_3_1_02#cla	ct_3_1_02₩[zlii -ct_3_1_02₩[zlii -crkspace₩ct_3 -cspace₩ct_3_1_ kspace₩ct_3_1_ -p) 02₩stub_samp rget)	2 k 1 c	Deselect R <u>e</u> fresi	All
<			>			
	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish		Cance	1

- 1. Use the [Browse ...] button to select the CodeScroll workspace or CodeScroll project path.
- 2. Projects existing in the selected path are displayed in the [Projects] list.
 - When the CodeScroll workspace is selected, all the projects under it are displayed.
- 3. Click [Finish] to create the project.

8.5. Create a C/C++ Project with CPI File

Create a project from a CPI file. The CPI file is an information file for project creation in the command line interface.

The CPI file must be created by the user using the template file in the {installation path}\plugins\ com.codescroll.gp.cli_x.x.x.x\cpi folder.

New Projec	ct			_		×
New Project						
	+ project from the CP	l file (* .cpi) inforn	nation.			
Location:	P:₩Product_Works	pace\ct_3_1_02				
CPI file						
File:					Browse(<u>B</u>)
	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish		Cance	1

- 1. Select a CPI file with the [Browse...] button.
- 2. Click the [Finish] button to create the project.

You can also create a project by dragging and dropping a CPI file into the test navigator.

8.6. Create a C/C++ Project with Build Information

Create a C / C ++ project from build information. The project is created as many as the number of modules extracted from the build information.

1. Enter a project name in [Project name]. The final project name is determined by combining the entered project name with the module name extracted from the build information.

 \times

 Ex) [Project name]Extracted module name Rew Project New Project Create C/C++ Project from Build Command. (e.g. makefile or build script) zlib_build_project Project name: P:₩Product Workspace₩ct 3 1 02 Location: Select Toolchain Default Toolchain Name Description CPP TI TMS320 6000 SoC Unit GCC 4.7 (32bit) GCC 5.3 (32bit) \checkmark Microsoft Visual Studio 2010 (32bit) Microsoft Visual Studio 2010 (32bit_64bit) Microsoft Visual Studio 2015 (32bit) Microsoft Visual Studio 2015 (32bit_64bit) qcc5 iar Toolchain Setting Import Setting Build Command: make clean all P:\Example_Project\c\mam\zlib-1.2.3 Build Directory: Browse...

2. Choose the same toolchain as the compiler you use for the build.

< <u>B</u>ack

3. In [Import Setting]-> [Build Command], enter the command required for the build.

Note: The final project name is determined by combining the entered project name with the module name extracted from the build information. (e.g. [project name] extracted module name)

• Ex) Make all / make clean

+ Advanced

- 4. Specify the path where the makefile is located in [Import Setting]-> [Build Directory].
- 5. If there are values that must be set before calling the compiler of the toolchain to be registered, add a script file to set the values to [Advanced]-> [Environment Script File].

<u>F</u>inish

Cancel

- 6. After completing all settings, click [Finish] to create the project.
- 7. When the project creation is completed after building, the result screen appears.

Create a C/C++ project with the build informati	on
The C/C++ project(s) has been generated with the (If a project with the same name already exists, ski	
Result:	
Project name	State
[zlib build project]NotLinked	success
[zlib_build_project]example	success
[zlib_build_project]minigzip	success
	OK
	ÖK

- Success: Project creation complete.
- Error: If the creation failed due to an error during project creation.
- Exclude: If the same project name exists in the workspace.

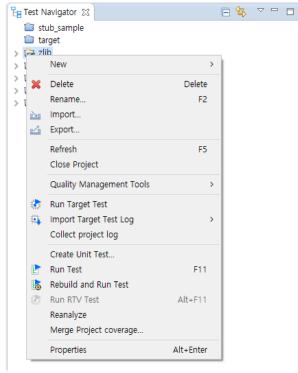
9. Create a Test

Generate test data and test code to test the functions of the source code. You can create individual tests for each function.

Create unit tests

Generate test data and test code for each function under test.

1. Select the project or module to create unit tests, right-click and click the [Create Unit Test] menu.



2. After the analysis is finished, in [Select function to be tested], select the function to perform the unit test.

New Test	— 🗆 X	Rew New	Test		- 0	×
Select Function to be tested		Select F	unction	to be tested		
Select Function to be tested.	Ep.	Select F	unction	to be tested.	E	H
Project: RTV_project		Droject	RTV_pr	alast		
Project. Krv_project		Project.	KIV_pi	Ject		
	Selection type: Source file 🗸			2	Selection type: Function	n ~
type filter text	type filter text		-	Name	Path	^
adler32.c	adler32(unsigned long, const unsigned char		-			
compress.c	adler32_combine(unsigned long, unsigned lo	1		 adler32(unsigned long, const unsigned char *, un 		_
□ crc32.c	adler32 combine64(unsigned long, unsigned	2		 adler32_combine(unsigned long, unsigned long, 		
deflate.c	adler32 combine (unsigned long, unsigned l	3		adler32_combine64(unsigned long, unsigned lon		
azclose.c	adler32 z(unsigned long, const unsigned cha	4		adler32_combine_(unsigned long, unsigned long,		
gzlib.c		5		 adler32_z(unsigned long, const unsigned char *, 		
gzread.c		6		compress(unsigned char *, unsigned long *, cons	📄 compress.c	
gzwrite.c		7		compress2(unsigned char *, unsigned long *, con	. 📄 compress.c	
infback.c		8		compressBound(unsigned long)	compress.c	
inffast.c		9		crc32(unsigned long, const unsigned char *, unsi	crc32.c	
inflate.c		10		 crc32_big(unsigned long, const unsigned char *, 	crc32.c	
inftrees.c		11		crc32_combine(unsigned long, unsigned long, si	crc32.c	
Trees.c		12		crc32_combine64(unsigned long, unsigned long,	crc32.c	
uncompr.c		13		crc32_combine_(unsigned long, unsigned long, si	. 📄 crc32.c	
zutil.c		14		 crc32_little(unsigned long, const unsigned char *, 	. 📄 crc32.c	
	< >	15		crc32_z(unsigned long, const unsigned char *, u	crc32.c	~
	0 items selected				0 items	selected
Run option		Run o	ntion			
Run after Test Creation				st Creation		
Host RTV		() () () () () () () () () () () () () (lost 🔘	RIV		
	Finish Cancel				Finish Car	ncel
	Cancer				Linish	icer

- a. You can filter the function under test using [type filter text].
- b. You can use the [Selection type] combo box to select the function to be tested based on the source file or function.
- c. You can easily select and deselect all functions under test with the right-click menu.
- d. When [Run after Test Creation] is selected, the test is executed after the test data is generated.
- 3. Click [Finish] to create the unit test.

Create and run tests

The [Create and Run] function allows you to create and run unit tests at once without using the [Create Unit Test] function.

1. Select the function to be tested in the test navigator and right-click.

E Test Navigator		⊟ 🔄	~ 8	
stub_sample			^	
🖉 target 🖌 🥁 zlib				
🗸 🎯 Default Module				
🗸 🗾 adler32.c				
adler22	Create Unit Test			
adle				
> 📙 compre	Create and Run			
> 🗾 crc32.c				
> 🗾 deflate.c				
> 🗾 gzio.c				
> 🗾 infback.c				
🔊 🔲 inffast c				

2. If you select the [Create and Run] menu, it creates a test for the target function and then runs the test.

Create an integration test

You can use the toolbar menu to create an integrated test.

Unit Test Integration Test	日 分 長 👫	🛃 🗄 🔹 🗢 🗖
	¥	Statement Coverage
Run 🕨	(33 / 0 / 0) 33	43.0% (1719/3995)
ype file, test, state, issue		
Name	Result	Coverage
<pre>> VINTEGRATION_0 > lest adler32_combine(unsigned long, unsigned long, signed case 1 case 2 case 2 case 3 case 4 case 5 case 6 case 7 case 8 case 9</pre>	(33 / 0 / 0) 33 (11 / 0 / 0) 11	100.0% (18/18)
Case 10 Case 11		

The integration test name is automatically assigned and can be renamed using the [Rename] context menu.

Unit Test	Inte	gration Test				🕒 分 🕹 🖣	🕯 😓 🗄 👻 🗁 🗖
							 Statement Coverage
Run 🕨	J					(33 / 0 / 0) 33	43.0% (1719/3995)
pe file, test, state,	issue						
Name						Result	Coverage
 INTEGRAT test adl case : 		Create Test Create sub-i	Rename		1	 (33 / 0 / 0) 33	100.0% (18/18)
Case Case Case Case	E:	Copy Paste Duplicate Duplicate m	New name:	INTEGRATION_0	ОК	Cancel	
Case Case Case	×	Paste as sub Delete Rename		Delete F2			
case : > test adl > test adl		Select Test Deselect Test			ig, signe	(11 / 0 / 0) 11 (11 / 0 / 0) 11	

10. Test Editor

The test editor can be opened by double-clicking the test or test case in each test view, located at the bottom of the unit/integrated test view.

Unit Test	Integration Test							E 😽 I	• ~ -	•
								✓ State	ment Cove	rage
Run 🕨						(19 / 0 / 5)	24	2.9	
type file, function, t	test, state, issue									-
Name							Result	C	overage	^
✓ ✓ adler32(unsigned long, const ur	signe	d char	*)			(8 / 0 / 5) 13		6 (100/103)	
	adler32(unsigned long,				ar *, unsi	g	(8 / 0 / 5) 13	51.07	0 (100/103)	
	2_combine(unsigned lor						11 / 0 / 0) 11	100.0)% (18/18)	
🗸 🗹 🛛 test	> adler32_combine(uns	igned	long,	unsigne	d long, s	i	(11 / 0 / 0) 11			
cas	e > 1									
cas	e > 2									
Contraction of the second seco	ie > 3									
cas	e > 4									
cas	ie > 5									
	e > 6									
	ie > 7									
	e > 8									
	e > 9									
2	ie > 10									•
🗄 adler32(unsigne	ed long, const u 🔀								-	
Test Info (zlib/a	dler32 test0)									3 ès
	- /									
Test Structure			0		Test Info	o Edit				
				. 🕀 🖻						
	ng a tree view and edit the in			e test.	Change	the notati	on edit the partition			
Name		In	Out	^	Set Nu	umeral Syst	em			
Test glob					10 10	Numeral S	/stem ○16 Numer	al System		
User code	•									
Global Va					Variab	ole Partitio	n			
	et function				Min		~ Max		Add	
	32(unsigned long, const unsig				IVIII		~ Iviax		Aud	
	ocal Static Variable				Singl	e Value			Add	
	arameter/return					ion List				
~ 0	adler : uLong		_			UT LISU				
	 adler : unsigned long buf : const Bytef * 	$\mathbf{\nabla}$			"0"				Delete	
	· · · · ·				"1"				Default	
	 ○ [0] : Bytef ✓ ○ [0] : Byte 				"2~1				Derault	
	 [0] : byte [0] : unsigned c 				"16"					
		•	-	~	"17-	~5551"				~
Test Info Test Case	Test Code Configuration									
					142M of 1	1032M				

You can move the editor and open it in a separate window.

10.1. Test Info

The Test Information tab is a feature that presents the test code as a tree of test structures, making it easy for users to modify the test code.

est Info (test/adler32_combine_test0)				4
fest Structure	C	Q 🗉 🖻	Test Info Edit	
est structure using a tree view and edit the information in the test.			Set the number format edit the partition.	
Name Test global code User code Global Variable	In	Out	Base	
 Test target function 			Min ~ Max	Add
Local Static Variable Parameter/return o adler1 : uLong			Single Value Partition List	Add
 Adler1 : unsigned long adler2 : uLong adler2 : unsigned long 			"0" "1~4294967295"	Defaul

- User code can be inserted into the test code.
- You can control global variables related to the function under test.
- You can control the input and output of the function parameters and return values.
- You can enter the code to be executed just before calling the function under test.
- * markers appear at modified items. When you save modifications, the markers disappear.

Repeat function calls

You can have the function under test be called multiple times.

adler32_combine(unsigned long, 🕅					
Test Info (zlib/adler32_combine_test0)				24 込	
The information has been changed after execution	ng the				
Test Structure		C	. 🕀 E	Test Info Edit	
Test structure using a tree view and edit the inf	ormati	on in th	ne test.	Test target function	
Name Test global code User code User code Global Variable Test target function of adlef32_combine(unsigned long, u Local Static Variable Parameter/return of adler1 : uLong	In	Out		iteration count 5	
 adler1 : unsigned long 				adler32_combinegned long 🕱	- 8
✓ o adler2 : uLong					
 adler2 : unsigned long 				32 33	^
 len2 : long o^R returnVar : uLong 	\checkmark			<pre>34 for(cs_index=0;cs_index<5;cs_index++){</pre>	
est Info Test Case Test Code Configuration				/* utong [5] adler32_combine(unsigned long, unsigned long, signed long) */ 7 returnVar[cs_index] * adler32_combine(adler1,adler2,len2); 39 } 40 } 41	
				<pre>42 //Output*/ 43 C5_UIUI_OUTPU(returnVar[0], "returnVar[0]"); 44 45 //*(END TEST>*/) 46 //*(S_DISCONNECT_CODE_START>*/ 47 //Do not remove disconnect_stub_configuration()*/ 48 disconnect_sdier32_combine_test0_stub_configuration(); 49 /*(S_DISCONNECT_CODE_END>*/ 49 } 51 52 eendif 53 </pre>	Ť

Export/Import variable partition

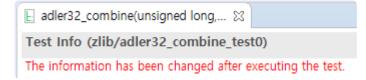
You can export/import variable partition information by selecting a variable on the Test Information tab.

: Info (test/adler32_combine_test0)				4
t Structure	(Q 🕀 🖻	Test Info Edit	
t structure using a tree view and edit the information in the test.			Set the number format edit the partition.	
ame	In	Out	Base	
Test global code			Decimal Hexadecimal	
User code				
Global Variable			Variable Partition	
Test target function				1
✓ ● ^T adler32_combine(unsigned long, unsigned long, signed lo			Min ~ Max	Add
Local Static Variable			Single Value	Add
 Parameter/return 			5	Auu
✓ • adler1 : uLong			Partition List	
 adler1 : unsigned long 	\checkmark		"0"	Delete
✓ ● adler2 : uLong			"1~4294967295"	
 adler2 : unsigned long 	\checkmark			Default
 len2 : long 	\checkmark			
✓ o ^R returnVar : uLong				
e ^R returnVar : unsigned long		\checkmark		
Before call code				
After call code				
User code				
Stub				
Host				
Target				

Menu	Description
🗳 Export Variable Partition	Export the partition information of the selected variable.
🟜 Import Variable Partition	Import the partition information of the selected variable.

Change detection

If you edit and save the information of a test that has already been executed, "The information has been changed after executing the test." is displayed at the top of the test editor.



10.2. Test Case

Test Case Tab provide the ability to edit inputs for a function under test and the expected value for the execution result.

Parameter Local Static Variable	Туре	Input	Expected Value	Host Output	Target Output
 Parameter/return 					
 adler1 					
 adler1 	unsigned long	65520			
 o adler2 					
 adler2 	unsigned long	131041			
 len2 	long	2147483647			
✓ o ^R returnVar					
o ^R returnVar	unsigned long			4279500767	
Before call code					
After call code					
User code					
 Stub 					

- In the Test Structure tree, you can enter input values for the variables specified as input.
 - If there is no input value, the value of the variable is 'undefined value', and the tool initializes the variable with no value entered as 0 (Blank if a string) for the convenience of the user.
- You can switch the representation of the Test Structure as a table or tree with the [Show as Tree], [Show as Table] toolbar menu.
- With the [Apply the test case in a lump] context menu, you can apply the input and expected values of the test case to other test cases in the same test at once.
- If you select the 'Out' of a variable in the Test Structure tree, You can check the value of the variable after the test is performed.
- Editable cells are displayed in a pale sky color.
- If the expected value is different from the host/target output, it is displayed in blue.
- If the host and target output values are different, it is displayed in yellow.
- You can use ~ (range) and logical operators! (Not), & (and), and | (or) in expected values.
- The following shortcuts are available.
 - Undo:Ctrl+Z
 - $\,\circ\,\,$ Move row: arrow keys ($_{\uparrow},~_{\downarrow})$
 - Editable cell focus: Enter
 - Editable cell focus out: Esc
 - Move column (editable cell): Tab(right), Shift + Tab(left)
 - Move test cases: Ctrl + \leftarrow / \rightarrow

Change detection

If you edit and save the information of a test case that has already been executed, "The information has been changed after executing the test." is displayed at the top of the test editor.

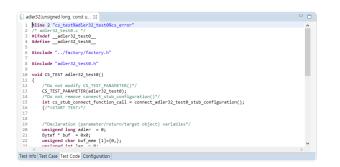
E adler32_combine(unsigned long,... ⊠

Test Case (zlib/adler32_combine_test0) #1

The information has been changed after executing the test.

10.3. Test Code

You can check the test code by clicking the Test Code tab in the test editor.



- If the user edits and saves the test structure tree, the changes are automatically reflected in the test code.
- If the return type of the function under test is a primitive type, a macro that outputs the return value is automatically generated.
- If you want to modify the test code yourself, you must switch the test to unmanaged code.
- The test execution function must have a return type and an input parameter of a void type.

10.4. Configuration

On the configuration tab, you can make various settings for unit testing.

E adler32_combine(unsigned long, ☆	-	
Configuration(zlib/adler32_combine_test0)		
Test Name		^
adler32_combine(unsigned long, unsigned long, signed long)_0		
Description		
	^	
	~	
Set related file		
P:#Example_Project#c#dip#zlib-1.2.3#adler32.c	Search	
Run Test		
Test Case Timeout(Second)		
% If the time-out value is empty, run the time-out value of the project.		
Maintain test case context		
Change Unm	nanaged Code	~
Test Info Test Case Test Code Configuration		

- If you modify and save the test name, the test name that appears in the unit test view changes.
- You can enter a description for the test.
- With the related file settings, you can select the file to which the function under test belongs.
- You can set the timeout for the test.
- You can choose whether to keep the context for the test.
- If you want to write test code yourself, you need to do [Change Unmanaged Code].

10.5. Test Macro

Controller Tester provides several macros to help users write test code more easily. You can check and write the macro provided using the shortcut(Ctrl + Spacebar) in the Test Editor.

ASSERT macro

Examine the conditional expression and print the success/failure in the Test Case Tab.

Macro	Parameter	Example
CS_ASSERT(_b)	_b: conditional expression	<pre>CS_ASSERT(val!=1);</pre>
CS_ASSERT_MSG(_B, _msg)	_B: conditional expression _mgs: message to print when the conditional expression is false	<pre>CS_ASSERT_MSG(val==1, "va l is not 1!");</pre>

Output macro

Print the values of specific variables in the Test Case Tab.

Масто	Parameter	Example
CS_INT_OUTPUT(_v, _s) CS_UINT_OUTPUT(_v, _s) CS_FLT_OUTPUT(_v, _s) CS_STR_OUTPUT(_v, _s)	v: a variable to print the value _s: a name to show in the Test Case Tab	<pre>CS_INT_OUTPUT(int_var, "int_var_name"); CS_UINT_OUTPUT(unsigned_int_var, "unsig ned_int_var_name"); CS_FLT_OUTPUT(float_var, "float_var_nam e"); CS_STR_OUTPUT(string_var, "string_var_n ame");</pre>

Input macro

Pass test data to the function under test.

Macro	Parameter	Example
CS_INT_INPUT(_t, _s) CS_UINT_INPUT(_t, _s) CS_FLT_INPUT(_t, _s) CS_STR_INPUT(_t, _s)	_t: a type of the variable _s: a name to show in the Test Case Tab	<pre>CS_INT_INPUT(int, "int_var_name"); CS_UINT_INPUT(unsigned int, "unsigned_i nt_var_name"); CS_FLT_INPUT(float, "float_var_name"); CS_STR_INPUT(char*, "string_var_name");</pre>

Address-related macros

If there is a part that directly assigns or fetches the value in the embedded address in the converted source code, it may not operate normally when executed on the local computer. In this case, you can use an address-related macro for the virtual address.

Масто	Description
CS_VIRTUAL_ADDR(_b,_e)	Creates the space from the address (_b) to the address (_e)
CS_ADDR_ASSIGN(_t,_a,_v) CS_ADDR_SET(_t,_a,_v)	Assigns the value (_v) of type (_t) to the address (_a)
CS_ADDR_GET(_t,_a)	Fetches the value of type (_t) from the address (_a)
CS_VIRTUAL_ADDR_CLEAR()	Frees the memory space created

Address-related macro example

```
// Create a virtual memory area of size 100 from 0xFFE40000U.
CS_VIRTUAL_ADDR(0xFFE40000U, 0xFFE40000U+100);
// Assign value 10 of int type to 0xFFE40000U
CS_ADDR_ASSIGN(int ,0xFFE40000U, 10);
CS_ADDR_SET(int ,0xFFE40000U, 10);
// Return the value from 0xFFE40000U to the variable a.
int a = CS_ADDR_GET(int ,0xFFE40000U);
// Return an address of 0xFFE40000U into variable ptr of pointer type.
int* ptr = CS_ADDR_PTR(0xFFE40000U);
// Frees the memory space created.
CS_VIRTUAL_ADDR_CLEAR();
```

For more information about address-related macros, see the <u>Virtual Address Usage</u> <u>Guide</u> page in the User Guides.

Other macros

Macro	Parameter	Description	Example
CS_LOG(_msg)	_msg: log message	Outputs user log	CS_LOG("User Log");
CS_TESTCASENO()		Returns the number of the test currently running	<pre>int testCaseNum = CS_TES TCASENO();</pre>

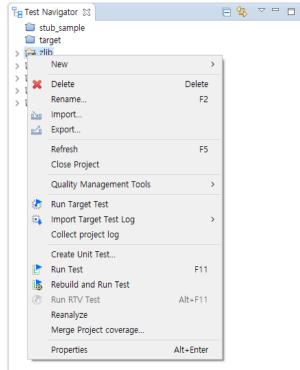
11. Test Run

Testing can be performed in a variety of ways.

Run all tests

Run the selected tests in the [Unit Test] view and [Integration Test] view.

1. Select the project to run the test, right-click ans slect [Run test]



2. A progress monitor of the test opens. When the test is completed, you can check the results through each test view.

Run unit tests

Run the selected unit tests in the [Unit Test] view.

1. Click the [Run] button on the dashboard in the [Unit Test] view.

✓ Statement Coverage
43.0%
(1719/3995)

2. A progress monitor of the test opens. When the test is completed, you can check the results through each test view.

To perform only a specific test case, select the test case you want to run and right-click the mouse and click the [Run Test Case] menu.

Run integration tests

Run the selected tests in the [Integration Test] view.

1. Click the [Run] button on the dashboard in the [Integration Test] view.

Unit Test Integra	ation Test	🗆 全 🕹 📫	👺 E 🔻 🖓 🗖 🗖
Dur N		¥	Statement Coverage
Run 🕨		(33 / 0 / 0) 33	43.0%
			(1719/3995)
type file, test, state, issue			

2. A progress monitor of the test opens. When the test is completed, you can check the results through each test view.

12. Inspect Debug Information

If an error occurs while running the test case, you can use the [Inspect Debug Info] function to determine the cause of the error.

[Inspect Debug Info] function can be checked in the unit test view with the context menu of the test case.

	Сору	Ctrl+C
	Paste	Ctrl+V
	Duplicate	Ctrl+D
	Duplicate multiple times	
×	Delete	Delete
	Add Test Case	
	Host Output Value -> Expected Value	
	Target Output Value -> Expected Value	
Ŀ	Run Test Case	
٢	Run Target Test Case	
	Add Stub	
	Open Host Error Loaction	
ð.	Inspect Debug Info	

Test cases that have not been executed are automatically executed before [Check Debug Information] is performed.

When [Inspect Debug Info] is performed, debug information is displayed in the [Inspect Debug Info].

👬 Debug Information 🔀			(x)= 🗖
Test Case (test/func_test0) #1			
Test Case (test/func_test0) #1 Stack trace:	List of variable/expression: Variable/Expression p (test1.c: 40)	Value 12319812	Location before line

The test case (Debug info included) is displayed in the test case where the [Inspect Debug Info] was performed.



13. Generate Function Call Flow Contents

Using [Generate Function Call Flow Contents by Test case] feature, you can print the called functions sequentially when you run the test case.

1. At the Unit Test View, right-click the test case and select [Generate Function Call Flow Contents by Test case].

-	-	
	Copy Paste	Ctrl+C Ctrl+V
	Duplicate	Ctrl+D
	Duplicate multiple times	
×	Delete	Delete
	Rename	F2
	Add Test Case	
	Host Output Value -> Expected Value	
	Target Output Value -> Expected Value	
Ŀ	Run Test Case	
٢	Run Target Test Case	
122	Add Stub	
ð,	Inspect Debug Info	
	Generate function call flow contents by test case	

2. When [Generate Function Call Flow Contents by Test case] is executed, the [Execute Test Success] window pops up.

G Exe	cute test success		×
1	Function call information file is generated.		
		Open Folder	Confirm

3. Click the [Open Folder] button in [Execute Test Success] window. The directory containing the function call flow content file is opened.

📙 🛃 🚽 functionCallFlow	-			_	×
File Home Share View					~ ?
← → × ↑ 📙 « .csdata > log > fur	ctionCallFlow	∨ Ü ∕	Search functionCallFlow		
Name	Date modified	Туре	Size		
[Table]compress_test0_1.csv [Tree]compress_test0_1.csv	5/16/2022 11:48 AM 5/16/2022 11:48 AM	CSV File CSV File	3 KB 2 KB		

2	items

<u>:</u>::: ■

The output file name is in the format of [type]test name_test case number.csv.

- The file includes name and parameters of the functions. There are two types in displaying function call flow.
 - Table
 - The calling relationship is displayed in two columns.
 - The caller functions are in the first column, and the callee functions are in the second column.
 - Tree
 - The function call information is displayed in multiple columns depending on the call depth.
 - The function call information is displayed on different lines if the call is located at different locations.
- If a function is consecutively called several times, calls are printed in an abbreviated way.
 - The number of calls is indicated by appending [called # times] after the function signature.
 - e.g., If the a function is consecutively called 5 times, [called 5 times] is appended to the first function signature.

Example

```
// example code
void callee() { /*doing something*/ }
void callee2() { /*doing something*/ }
void func1() { callee(); callee2(); }
void func2() { callee(); callee(); callee(); }
void caller() { func1(); func2(); }
```

• Table

caller-function	callee-function
caller()	func1()
func1()	callee()
func1()	callee2()
caller()	func2()
func2()	callee() [called 3 times]

Tree

caller()	func1()	callee()
		callee2()
	func2()	callee() [called 3 times]

Constructors, destructors, operators and functions without definition (e.g., system functions) are excluded from the output. Functions called from ternary operators or from try and catch statements may not output properly.

14. Properties

Controller Tester provides a page that enables to set the properties of the project, module, and source files.

The property page types are:

- Project Properties
- Module Properties
- Source File Properties

14.1. Project Properties

You can edit the properties of the created project.

Select [Project]-> [Properties] from the menu or right-click the project in the Test Navigator view and click the [Properties] menu.

Inclusion and Exclusion Info

You can set what to exclude or include when analyzing the project.

To analyze only selected directories or files, add the targets to be analyzed to [Included] and check [Exclude All].

Properties for zlib				×
type filter text	Inclusion and Exclusion	4	• =	• •
Inclusion and Exclusion Info > Target test > Test	Included		Add <u>F</u> ile d <u>D</u> irect <u>E</u> dit <u>R</u> emov	ory
		Ad	Add <u>E</u> ile d <u>D</u> irect <u>E</u> dit <u>R</u> emov	ory
	Exclude All	_		
	Note: If there is one resource in both places, the resource is inclusion item.			
	Apply and Close		Cance	

- Add targets to include in the analysis.
 - 1. Click the [Add File] or [Add Directory] button.
 - 2. Select files or a directory to analyze.
 - 3. Click the [Open] or [OK] button.
- Add targets to exclude from the analysis.
 - 1. Click the [Add File] or [Add Directory] button.
 - 2. Select files or a directory to exclude from the analysis.
 - 3. Click the [Open] or [OK] button.

Info

Shows brief information(path, type, location) about the project.

Properties for zlib		— 🗆 X
type filter text	Info	<-> < <-> < <-> <-> <-> <-> <-> <-> <->
Inclusion and Exclusion Info > Target test > Test	Project Information Path: /Zlib Type: Project <u>Location</u> : P:/Product_Workspace/ct_3_1_02/Zlib	
		Apply and Close Cancel

Virtual Memory Address

You can select a memory setting managed by the Virtual Memory Address preference.

Properties for zlib		— 🗆 X
ype filter text	Virtual Memory Address	← - ⇒ -
Inclusion and Exclusion Info Target test	Configuration name: Detail information	×
 Test External Command Generate test cases autor Mock Run Test Virtual Memory Address 	Begin	End
		~
		Apply and Close Cancel

Mock Object

For C++ project, you can select an option to use mock objects. [Use] is selected by default for toolchains(GCC 6.0 or later, Visual Studio 2015 or later) that can use mock objects.

Properties for poco			×
type filter text Inclusion and Exclusion Info Test Exclusion of coverage External Command Generate test cases automatically Mock Run Test Virtual Address	Mock Use Mock © Use O Disable If you select Disable, the expected value of the mock object is not reflected Reanalysis is required when changing the mock object usage option. Restore Defaults		
	Apply and Close	Cancel	

External Command

You can enter external commands to run before or after the test run.

In [Pre-command for test run], you can enter a command or batch file to execute an external command before running the test.

In [Post-command for test run], you can enter a command or batch file to execute an external command after running the test.

Properties for zlib		_			×
type filter text	External Command		⇔ -	⇒ •	
 Inclusion and Exclusion Info Target test Test Exclusion of coverage External Command Generate test cases aut Run Test Virtual Address 	Pre-command for test run P:#Example_Project#c#dip#zlib-1.2.3#rebuild.bat Post-command for test run P:#Example_Project#c#dip#zlib-1.2.3#rebuild.bat				
	Apply and	l Close	Ca	ncel	

Exclusion of coverage

You can set coverage exclusions on a per-file or per-function basis.

You can see that the coverage of the excluded function is excluded from the coverage results obtained by running the test.

pe filter text	Eve	clusion of coverage		(-) - (-)	⊑) ▼.
Inclusion and Exclu	usion			V-	~
Info		e list of resources excl	-		
Target test		ne list of files excluded	i coverage.		
 Test Exclusion of co 	S	Source File		A	dd
External Comm				<u>R</u> e	move
> Generate test c	ases aut				
Run Test Virtual Address	. -				
Virtual Address	, I				
	_				
	_				
		ne list of functions exc			
		e list of functions exc Function	luded coverage. Source File		<u>.</u> dd
					-
					-
					-
					-
					-
					-
					-
					-
					-
					-
					-
					-
					-
					udd

Run Test

Set the generated tests to run in a user-specified directory, not in a workspace. In this case, you do not need to copy the library or header files used by the target under test to the workspace or change the link settings.

Click the [Start Test in User Code] checkbox. Select the home directory to run the test and click the [Apply] button. Suresofttech

Properties for zlib		— 🗆 X
type filter text	Run Test	← ▼ □ → ▼
Inclusion and Exclusion Info > Target test Exclusion of coverage External Command > Generate test cases au Run Test Virtual Address	Start Test in User Code Test Launch Directory : Test Case Timeout(Second) 3 Maintain test case context Redirect standard output or standard error	Browse Browse Restore Defaults
		Apply and Close Cancel

[Test Case Timeout] Set the time to be used as the timeout judgment when executing the test case. If the test execution time exceeds the test case timeout time, the test for that test case ends and the result is reported as a timeout.

[Maintain test case context] is a feature that performs all test cases with one launcher. The previous test execution result (static, global variable) is maintained during the test, and each test case result value does not appear independently. This makes the relationship between the test case result values clear.

[Redirect standard output or standard error] is a feature to record standard output and standard error in the log file for each test case.

Generate test cases automatically

The search-based generation is not affected.

You can set to keep previous test cases when automatically generating test cases.

R Properties for zlib		– 🗆 X
type filter text	Generate test cases automatically	
Inclusion and Exclusion Info > Target test Exclusion of coverage External Command > Generate test cases automatically Run Test Virtual Address	Generate test cases automatically Generate test cases when auto generating test cases X Search-based generation is not affected.	
< >>		Restore Defaults Apply
		Apply and Close Cancel

Combination

You can set up how to extract test data if the variable partition is range.

Properties for zlib	— 🗆 X
type filter text	Combination $\diamond \star \bullet \star \bullet$
Inclusion and Exclusion Info > Target test Test Exclusion of coverage External Command • Generate test cases aut Combination Function Generator Search-based (CS L Run Test Virtual Address	Create Test Data from Partition ✓ Max ✓ Min □ Random Data Combination ● Flat Combination ● Pairwise Combination Number of Limit: 200 ○ Random Combination Number of Test case: 5 Array elements ● Generate array of the first element ○ Generate array of all the elements Generate global variable □ Create global variable in the relationship between function calls.
	Apply and Close Cancel

Select the desired test data extraction method by checking the checkbox. The meaning of each value is as follows.

Method	Description
Max value	Selects the maximum value of the partition section.
Min value	Selects the minimum value of the partition section.
Random value	Selects a random value among partition sections.

You can select multiple test methods of data extraction.

When all are unchecked, the default is automatically set to the maximum and minimum values. [Data combination] supports Flat, Pairwise and Random. The meaning of the combination is shown in the following table.

Combination	Description
Flat	Combines simply based on variables having the largest number of test data.
Pairwise	Combines so that each selected parameter data is paired at least once with the parameter data other than itself. Pairwise is performed as many as the maximum number(default 200) of symbols used for the combination, and when the number of symbols is exceeded, Flat is performed.
Random	Combines the test data as many as the number of test cases that the user defines any value between the minimum value and the maximum value for the variable partition of input parameters(default 5).

[Array elements] is a feature to select how to generate test data for each array element.

Method

Description

Generate an array of the first element	Creates only the first element regardless of the array size.
Generate an array of all the elements	Creates elements as much as the array size.

When creating a test, global variable creation is an option to create all global variables used by all functions called from the function under test.

Search-based (CS Labs)

CS Labs allows you to preview new features that will be released in CodeScroll tools later. Based on feedback for the new features, CS Labs features may or may not be applied as formal features.

You can set the maximum search when automatically generating test cases based on search. You can turn this feature on or off with [activating search-based test case generation] option. Reanalysis is required after turning the feature on.

Ct Properties for Update_Modify		- D X
type filter text	Search-based (CS Labs)	← ▼ ⇒ ▼
Inclusion and Exclusion Info Test External Command Generate test cases aut Combination	Activate search-based test case generation General settings for maximum search: Number of searches(for each branch case): Search time (seconds):	
Function Generator Search-based (CS L Run Test Virtual Address		Restore Defaults Apply
		Apply and Close Cancel

Option	Description
Number of searches(for each branch)	Searches for the number of times entered for each branch combination.
Search time(seconds)	Searches for the time entered.

Function Generator

When using the function generator to automatically generate test cases, specify common setting values that each function has in common.

Suresofttech

Properties for zlib							×
type filter text	Function Gener	ator			¢	• =	• •
Inclusion and Exclusion Info > Target test < Test Exclusion of coverage External Command < Generate test cases automatically Combination Function Generator Search-based (CS Labs) Run Test Virtual Address	Sample interval: Sample count: Start value:	0		Restore Def	aults	Дрр	y
				Apply and Clo	se	Cancel	

Option	Description
Sample interval	Interval of samples when sampling from function
Sample count	Number of samples when sampling from function (Number of test cases)
Start value	Default value at which the function starts, and generates a value based on the value

Import settings

Project created by Visual Studio project

pe filter text	Import			< → →	
Import	import			~ · ~ ·	
Inclusion and Exclusion	Visual Studi	o Project: P:\Example_Pr	oject₩c₩nvip₩cpfe₩cpfe.sln		
Info > Target test > Test	Select Active	Select Active Modules:			
	Active	Module	Configuration (Click to Edit)		
i cot		misc\mk_errinfo	Debug/Win32		
		src#disp#disp	Debug/Win32		
		src₩fe	Debug Win32		
	Select All	Deselect All		Set <u>All</u> Configurations	
			Restor	re Defaults Apply	
			Restor	e Deugues Ebbil	

You can see the path of the Visual Studio project used for creating the project and the modules active in the selected project.

You can change whether each module is active or not and the configuration.

You can change the configuration of all modules collectively through the [Set All Configurations] menu.

1. If you click the [Set All Configurations] button, a dialog box where you can select one of the configurations common to all modules opens.

×						
Common Configurations:						
ancel						

2. Select the configuration to be applied to all modules in common and click the [OK] button.

14.2. Module Properties

Build

You can set toolchain related information (toolchain, linker) and link flags for the module. Changing the toolchain also changes the toolchain set in all source files under the module.

Properties for Default Modul	e	— 🗆 X				
type filter text Build Compile flag	Build	<> ▼ <> ▼ ▼				
	Select a toolchain					
	Default Toolchain Name [^] CPP_TI_TMS320_6000_SoC_Unit GCC 4.7 (32bit) GC 5.3 (32bit) Microsoft Visual Studio 2010 (3 Microsoft Visual Studio 2015 (3 Gcc 5.3 Image: Studio 2015 (3 Image: Studio 2015 (3	Description				
	Linker: gcc.exe V					
		~				
	Undo <u>R</u> edo <u>F</u> ind/Repla	ice				
		Restore Defaults Apply				
		Apply and Close Cancel				

Compile flag

You can set compile flag information for that module.

Properties for Default Module							×	
type filter text	Compile flag	g				⇔ - ⇒		
Build Compile flag	Note: The compile flag information is saved in all source files set to use a parent compile flag.							
					Overwrite a	ll sub sourc	e files	
	Command	Include	Define Macro					
							^	
					1		~	
	<u>U</u> ndo		<u>R</u> edo	<u>F</u> ind/Replace				
					🚵 Import	🛃 Ехр	ort	
				A	pply and Close	Cance	I	

- Enter compile flags in [Command] tab. [Include] tab and [Define Macro] tab help to enter compile flags easily.
- [Overwrite all sub source files] overwrites the compile flag entered to all source files under the module.
- [Import] imports an external compile flag. [Export] saves the current compile flag externally.
 - Import
 - 1. Click the [Import] button.
 - 2. Select the file (.cf file) which has compile flags from the file open dialog.
 - Export
 - 1. Click the [Export] button.
 - 2. In the save dialog, enter the name and location of the file to enter the compile flag and save.

14.3. Source File Properties

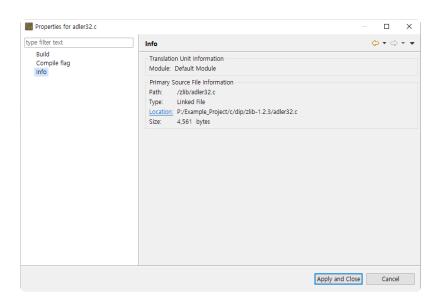
Build

You can set toolchain related information (toolchain, compiler) for the source file.

filter text	Build			<> ▼	⇒ •				
Build Compile flag	Select a to	Select a toolchain							
Info	Default	Default Toolchain Name Descrip							
		CPP_TI_TMS320_6000_SoC							
		GCC 4.7 (32bit)							
		GCC 5.3 (32bit)							
		Microsoft Visual Studio 20							
		Microsoft Visual Studio 20							
		Microsoft Visual Studio 20							
		Microsoft Visual Studio 20							
		gcc5							
		iar							
	Compiler	Compiler: gcc.exe 🗸							
		Restore Defaults Apply							
			-						

Info

Shows the information of the source file and the module containing the source file.



Compile flag

You can set compile flags for source files.

Properties for adler32.c						×
type filter text	Compile flag				⇔ - ⇒	
Build Compile flag Info			DP	arent compile	flag inher	itance
	Command Include	Define Macro				
						^
						_
	Undo	<u>R</u> edo	Eind/Replace			-
	<u>s</u> .100	1000			1.0	_
			Ľ	🔄 Import	🗳 Exp	οπ
			Apply	and Close	Cance	ł

- If you check the [Parent compile flag inheritance] checkbox, the compile flag of the module containing the source file is applied. Conversely, you can change the current compile flag information by unchecking the checkbox.
- [Import] imports an external compile flag. [Export] saves the current compile flag externally.
 - Import
 - 1. Click the [Import] button.
 - 2. Select the file (.cf file) which has compile flags from the file open dialog.
 - Export
 - 1. Click the [Export] button.
 - 2. In the save dialog, enter the name and location of the file to enter the compile flag and save.

15. Preferences

In the preference menu, you can check or change the current settings of the tool.

The preferences provided by Controller Tester are as follows.

- Analysis
- Exclusion
- Performance
- Source File Types
- Language
- <u>Test</u>
- <u>Toochain</u>
- Editor

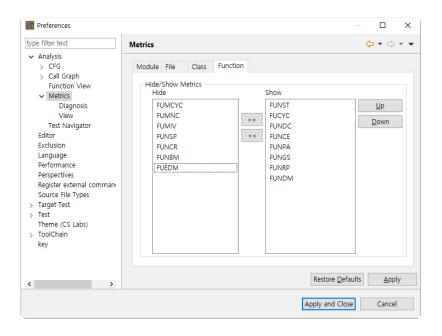
15.1. Analysis

You can check or change the settings related to the analysis.

Metric

For any view showing metric data, you can set which metrics are visible or hidden and the order in which they are displayed.

The name of the selected metric is displayed as a tooltip message.



After selecting a metric, you can move to the 'Hide' list or the 'Show' list with the [>>] or [<<] button. After selecting a metric in the 'Show' list, you can change the order in which the metrics are displayed with the [Up] or [Down] button.

Metrics View

You can change the settings of the views showing metric data.

Preferences	– D X
 Preferences type filter text Analysis CFG Call Graph Function View Metrics Diagnosis View Test Navigator Editor Exclusion Language Performance Perspectives Register external commani Source File Types Target Test Test ToolChain key 	View $> \checkmark > \checkmark$ Metrics View Rows per page: 100 Metrics Top Chart View
	Sum for the rest of the items to display Number of items to display:
	Number of items to display: 500 Metrics view and Metrics chart view interwork Filtering the items of the metric view based on the value selected in the metrics chart view Number of items to display: 5 The metrics chart view display a range of filtering in metrics view
	Metrics Diagnosis Chart View ☑ Show the sum of items that cannot be diagnosed
< >	Restore Defaults Apply Apply and Close Cancel

Diagnosis

You can check or set diagnostic information for metrics.

Set the message and diagnostic image (shown in the metrics view) to be displayed when the selected metric value is included in each range.

Preferences			_		×
type filter text	Diagnosis			<-> →	• •
 Analysis CFG Call Graph Function View Metrics Diagnosis View Test Navigator Editor Exclusion Language Performance Perspectives Register external commanic Source File Types Target Test Test Test Theme (CS Labs) ToolChain key 	Module File Class Diagnosis of: FUCYC ~ Message Simple(without mu More complex(mo Complex(high risk) Greatly complex(ve	[10.00, 20.00) [20.00, 50.00)	Image Q Q C C C C C C C C C C C C C	<u>E</u> dit	
< >>			Restore <u>D</u> efault	s <u>A</u> pp	ly
		Ap	ply and Close	Cancel	

To edit the diagnostic information, click the [Edit] button.

A dialog box appears where you can edit the diagnostic information for the selected metric.

Edit Diagnosis of 'FUCYC'	×
Diagnosis Level: 4	
Set of Images: Weather4	
Level1	_
Message: Simple(without much rish)	
Range: 0.00 ~ 10.00 Image: 🧼	
Level2	
Message: More complex(moderate risk)	
Range: 10.00 ~ 20.00 Image: 😜	
Level3	
Message: Complex(high risk)	
Range: 20.00 ~ 50.00 Image: 🚓	
Level4	
Message: Greatly complex(very high risk)	
Range: 50.00 ~ max Image: 🐏	
OK Cancel	

- 1. Set the diagnostic stage $(2 \sim 5)$.
- 2. Sets the diagnostic image to show in the metrics view.
- 3. Press the [Change Sort Order] button to change the sort order of diagnostic images.
- 4. Enter the diagnostic message and diagnostic range for each step.
- 5. Select the [OK] button.

CFG

You can change the settings for the control flow graph displayed in the control flow graph view.

Preferences		— 🗆 X
type filter text	CFG	←>
 Analysis CFG Call Graph Function View Metrics Test Navigator Editor Exclusion Language Performance Perspectives Register external commany Source File Types Target Test Test Theme (CS Labs) ToolChain key 	Control Flow Graph Preferences: Show Function Call Show Library Functions Enable Auto-Layout	
< >>		Restore Defaults Apply
		Apply and Close Cancel

Node Color

You can change the color of nodes in the control flow graph.

Preferences		- 🗆 X
type filter text	Node Color	⟨¬ → ¬
type niter text Analysis CFG Node Color Call Graph Function View Metrics Test Navigator Editor Exclusion Language Performance Perspectives Register external commani Source File Types Target Test Test Theme (CS Labs) ToolChain key	Node Color CFG Node Color: Begin Node Color: End Node Color: Condition Node Color: Block Node Color: Goto Node Color: Return Node Color: Function Node Color: Library Function Node Color: Single Statement Node Color: Single Declaration Node Color: Group Node Color:	
		Restore Defaults Apply
< >		Apply and Close Cancel

Test Navigator

For each operation that copies or moves source files in the Test Navigator view, you can set options to ask whether or not to continue.

Preferences				×
type filter text	Test Navigator	¢	• =>	• •
 Analysis CFG Call Graph 	Copy source files to the other module by drag and drop. ○ Always ○ Never ● prompt			
Function View Metrics Test Navigator	Move source files to the other module by drag and drop. Always ONever OPrompt			
Editor Exclusion Language	Copy and paste a source file to a module O Always O Never			
Performance Perspectives Register external comman Source File Types > Target Test Therme (CS Labs) > ToolChain key	Note: Drag and drop a source file to the other module with Ctrl key pushed down in ord	er to cop	y a sou	rce file.
< >>	Restore Def	aults	Арр	ły
	Apply and Clo	se	Cance	

Function View

Preferences				\times
type filter text	Function View			
 Analysis CFG Call Graph Function View Metrics Test Navigator Editor Exclusion Language Performance Perspectives Register external commany Source File Types Target Test Test Theme (CS Labs) ToolChain key 	Limit visible functions per group to (0: Unlimited):	0		
< >>	Restore	<u>D</u> efaults	<u>A</u> ppl	у
	Apply and	Close	Cancel	

Call Graph

You can change the settings of the function call graph displayed in the call graph view.

Preferences			— 🗆 X	(
type filter text	Call Graph		← → ⇒ →	•
 Analysis CFG Call Graph Function View Metrics Test Navigator Editor Exclusion Language Performance Perspectives Register external commany Source File Types Target Test Test Theme (CS Labs) ToolChain key 	Call Graph Preferences: Show Library Functions Call Depth Setting O Does not distinguish the Depth setting: 2 © Distinguish the direction Incoming depth setting: Outgoing depth setting:	2	Restore Defaults	
			Apply and Close Cancel	

The [Show Library Functions] checkbox allows you to set whether or not to display library functions. If you set the value of 'Does not distinguish the direction' to '2' in 'Call Depth Setting', it shows the call information of one depth for both the caller and callee, centering on the selected function(Just the callers or callee of the selected function shown).

If you want to set the caller and callee separately, you can specify the value of each step by selecting the sub-option. You can change the color of nodes visible in the function call graph.

Preferences		— 🗆 X
type filter text	Node Color	↓ ↓ ↓
 Analysis > CFG Call Graph Node Color Function View > Metrics Test Navigator Editor Exclusion Language Performance Perspectives Register external commany Source File Types > Target Test Theme (CS Labs) > ToolChain key 	Call Graph Node Color: Function Node Color:	
< >		Restore Defaults Apply
		Apply and Close Cancel

15.2. Exclusion

You can set the directories and files to exclude from the analysis.

type filter text Editor > Analysis Editor Editor Editor settings: Ecusion Iext file encoding Language © Default (UF-8) Performance Ogther: Perspectives Capted (UF-8) Capted (UF-8) Ogther: UF-8 Automatic detection Indentation Tab gize: Test Indentation Tab gize: 4 Detect modifications Ogheck the effect whenever the source file is modified (including external modifications) Check the effect whenever the source file is modified (including external modifications)	Preferences	- 0	×
Editor Exclusion Language Performance Perspectives Register external comman: Source File Types > Target Test > Test Theme (CS Labs) > ToolChain key		Editor 🗢 👻	⇒ • •
	Editor Exclusion Language Performance Perspectives Register external commany Source File Types > Target Test > Test Theme (CS Labs) > ToolChain key	Iext file encoding Default (UTF-8) Qther: UTF-8 Automatic detection Indentation Tab size: Detect modifications Image: Check the effect whenever the source file is modified (including external modified)	
	\$	Apply and Close Ca	ncel

- Add the analysis exclusion target.
 - 1. Click [Add].
 - 2. Select the directories or files to be excluded.
 - 3. Click [OK] or [Open].
- Delete the analysis exclusion target.
 - 1. Select the directories or files to be deleted.
 - 2. Click [Remove].

15.3. Performance

It provides the settings that can improve the work performance suitably for the user environment.

Preferences				×
type filter text	Performance	¢	• =>	• •
 > Analysis Editor Exclusion Language Performance Perspectives Register external commany Source File Types Target Test > Test Theme (CS Labs) > ToolChain key 	Multi Core usage ✓ Perform Multi-core Analysis Number of threads: 8 ☐ Using frozen code Note: Use this option if there are no changes to the source code. Minimizing unnecessary tasks to improve performance.			
< >>	Restore <u>D</u> efault	s	<u>A</u> ppl	у
	Apply and Close		Cancel	

Multi-Core Usage

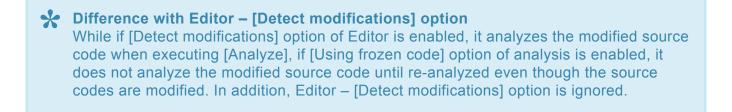
The option [Multi Core usage] improves performance by using multiple cores when executing the project analysis. (If PC has not enough CPU resource, it can affect the performance of the other tasks.)

 Perform Multi-core Analysis: Enter the number of the core to be used when analyzing the source code. (The range of the core number: 2 ~ the maximum number of cores in the execution environment)

Using frozen code

The option [Using frozen code] improves the performance by minimizing unnecessary tasks (detecting and re-analyzing source code modifications etc.) assuming that there is no change in the source code configuration.

Use this option only if the PC performance is very slow and the source code configuration is never changed during testing. (If the source code or analysis setting is changed while the option is enabled, it may cause unexpected problems.)



15.4. Source File Types

You can set the extension to be used as a source file for each project type.

Preferences				– 🗆 X
type filter text	Source File	Types		(⇒ ▼ ⊂) ▼ ▼
Analysis Editor Exclusion Language Performance	Source file ty Project Cate	pes can add or delete. gory:		
Perspectives Register external command Source File Types > Target Test	C/C++ File 1	Types:		
 > Test Theme (CS Labs) > ToolChain key 	File Type *.c *.C *.cc *.ccp *.cxx	e · · · · · · · · · · · · · · · · · · ·	Status Product Def Product Def Product Def Product Def	Add Remove
< >			Restore <u>D</u> efau	ilts <u>A</u> pply
			Apply and Close	Cancel

- Add a source file extension.
 - 1. Select the project type.
 - 2. Click [Add].
 - 3. Enter an extension, select the type and click [OK].

C/C+·	+File Type		×
C/C++ F	ile Type		
Enter a	file type.		
	-		
Pattern :	*.		
type :	C Source File		\sim
	OK	Cancel	

- Remove a source file extension.
 - 1. Select the project type.
 - 2. Select the extension to be removed.
 - 3. Click [Remove].

15.5. Language

Sets the language to be displayed in tools and reports.

You can select a language from the drop-down menu for [System] and [Report].

ct Preferences		— 🗆 X
type filter text	Language	↓ → → ▼
 > Analysis Editor Exclusion Language Performance Perspectives Source File Types > Target Test > Test Theme (CS Labs) > ToolChain key 	System: English (United States) ~ Report: English (United States) ~	re Defaults Apply
	Apply an	nd Close Cancel

If you click the [Apply] button after changing the [System], a dialog box asking you to restart Controller Tester will open.

C Preferences				\times
type filter text	Language	¢	• =>	• •
Analysis Editor Exclusion Language Performance Perspectives Source File Types	System: Korean (South Korea) v Report: English (United States) v			
> Tarc data Change Language			×	
Test The Too key You need to re Would you like	start Controller Tester to change the [System] language e to restart? Yes	No		
		fer the		
	Restore De	rauits	Appl	у
	Apply and Clo	se	Cancel	

* When you change the system language, the report language also changes automatically.

To set the report language differently from the system language, change the report language after restart.

15.6. Test

Provides settings related to editing test and execution.

[Except for non-executable tests when errors occur during test execution] is an option that allows you to execute only executable tests, excluding non-executable tests, when the whole test cannot be performed due to an error that occurs during test execution.

The [Using Macro Constant] option allows you to edit the test using macro constants defined in the source file (ex: using macro constants for input/expected values of array indexes and test cases).

V This setting only applies to converted projects. Preferences × type filter text **⇔** - ⇒ - -Test > Analysis Except for non-executable tests when errors occur during test execution Editor ○ Alwa<u>v</u>s ○ <u>N</u>ever ● <u>P</u>rompt Exclusion % This does not apply to the target test. Language Performance Note: If you can not proceed with the test due to an error that occurred during the test run, continue to run only the runnable tests, with the exception of non-executable tests. Perspectives Register external command Source File Types Using Macro Constant Target Test 5 Note: Changes to the 'Use Macro Constants' option are not reflected in existing projects. To be reflected in an existing project, you must perform a "reanalysis". > Test Theme (CS Labs) > ToolChain key Restore Defaults Apply < Apply and Close Cancel

Virtual Memory Address

It provides the memory settings for the embedded environment test. You can manage the memory address group to be used in the project properties.

Preferences			_	
type filter text	Virtual Memory A	ddress	<	;
type filter text > Analysis Editor Exclusion Language Performance Source File Types > Target Test Coverage Export test data External editor Perspective Type Partition Unit Test View Virtual Memory Address Theme (CS Labs)	Configuration n	Description	st:	Add Edit(E) Remove
> ToolChain	Begin		End	
			Apply and Close	Cancel

In the screen above, you can check the virtual memory address information set. Click [Add...] button to add the virtual memory address.

Virtual mer	mory address Crea	ite	×
Virtual memo	ory address Con	figuration	
		address value list is up to 50. alue format is hexadecimal.	
Basic Configu	ration		
Name:			
Description:			< >
Virtual memo	ry address Config	uration	
Begin Virtual	memory addre:	- End Virtual memory addre:	Add(<u>A</u>)
Begin		End	Delete(<u>R</u>)
		OK	Cancel

The virtual memory address areas can be specified up to 50 areas and it can be entered in hexadecimal.

Export test data

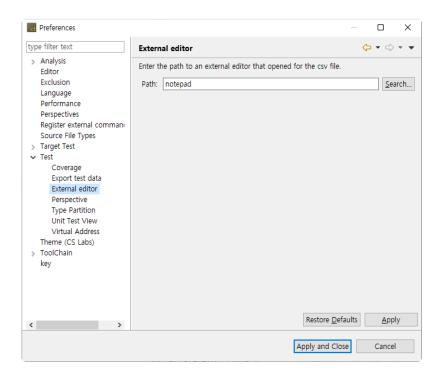
When exporting data, you can set the output direction of variables to either horizontal or vertical

Select the output direction of the export variable and click [Apply].

Preferences		- 🗆 X
type filter text	Export test data	←>
 > Analysis Editor Exclusion Language Performance Perspectives Register external commani Source File Types > Target Test > Test Coverage Export test data External editor Perspective Type Partition Unit Test View Virtual Address Theme (CS Labs) > ToolChain key 	Export data Configuration Horizontal Vertical	
< >>		Restore <u>D</u> efaults <u>A</u> pply
		Apply and Close Cancel

External Editor

You can set the external editor to open CSV file. Enter the path of the external editor and click [Apply].



Unit Test View

You can set the function node display in the Unit Test view included in the test perspective that UI/UX is improved.

Preferences		_		×
type filter text	Unit Test View	4	> ▼ <>	• •
 Analysis Editor Exclusion Language Performance Perspectives Register external commani Source File Types 	Function node show mode O All functions O Only functions where tests exist			
 Target Test Test Coverage Export test data External editor Perspective Type Partition Unit Test View Virtual Address 				
Theme (CS Labs) > ToolChain key				
< >>	Restore Defa	aults	<u>A</u> ppl	у
	Apply and Close	ie	Cancel	

Coverage

You can set the branch coverage measurement and whether to display the covered/uncovered.

Preferences	- D X
type filter text	Coverage 🗘 👻 🗸
Analysis Editor Exclusion Language Performance Perspectives Register external commani Source File Types Target Test Coverage	Branch coverage column info display
Export test data Export test data External editor Perspective Type Partition Unit Test View Virtual Address Theme (CS Labs) > ToolChain key	Number Operators(&&, , !=): 40 * Number of truth-table combinations: 20000 * Show MC/DC Show decision unit 100 * % more than satisfied combinations show
< >>	Note: The operator option change does not apply to the existing project. To apply, use 'Reanalysis' and 'Run Unit Test' function. Restore Defaults Apply
	Apply and Close Cancel

[Branch coverage Column Info display] radio button allows you to select the information of the marker to be displayed in the left column of the editor.

- T/F marker option is shown in the source code editor.
- Covered Decision: T/F indication of a covered decision
- Uncovered Decision: T/F indication of an uncovered decision

The checkbox [Branch coverage, MC/DC measurement operator] sets the branch coverage measuring

target. If it is re-analyzed and executed, the settings are applied.

- Conditional Operator(?:) expression: Measures the branch coverage for ternary operator expression.
- Boolean Operator(&,||,!,<,<=,>,>=,==,!=): Measures the branch coverage for boolean operator expression.
- If unchecked all, measures the branch coverage for 'if', 'for', 'while', 'do-while' and 'switch' statements only.
- When changing the 'Conditional Operator(?:)' setting, the COVER's [Toolchain] > [Standard] must be set identically to share coverage with COVER. For details, please refer to the <u>Import Coverage by Ternary Operator Settings</u> page in User Guides.

Type Partition

You can edit the basic type partition for each toolchain and restore it into the default value provided. In Toolchain combo box, select the toolchain to be modified, edit the partition and click [Apply].

Preferences		_	C	X
type filter text	Type Partition		⇔ -	⇒ • •
> Analysis Editor	Toolchain: Microsof	t Visual Studio 2015 (32bit_64bit)		~
Exclusion Language Performance Perspectives Register external command Source File Types > Target Test Coverage Export test data External editor Perspective Type Partition Unit Test View Virtual Address Theme (CS Labs) > ToolChain key	Type int64 wchar_t char double float int long long double long long short signedint64 signed long signed long signed long long signed short unsignedint64	Partition -9223372036854775808~-1,0,1~92233 0,1~65535 -128~-1,0,1~127 -1.7976931348623158e+308~-2.225073 -3.402823466e+38~-1.175494351e-38,0 -2147483648~-1,0,1~2147483647 -1.7976931348623158e+308~-2.225073 -9223372036854775808~-1,0,1~92233 -32768~-1,0,1~32767 -9223372036854775808~-1,0,1~92233 -128~+138548~-1,0,1~2147483647 -2147483648~-1,0,1~2147483647 -9223372036854775808~-1,0,1~92233 -32768~-1,0,1~217 -2147483648~-1,0,1~2147483647 -9223372036854775808~-1,0,1~92233 -32768~-1,0,1~32767 0,1~18446744073709551615		Add Edit(E) <u>R</u> emove
	unsigned char	0,1~255	~	
< >>		Restore Defaults		<u>A</u> pply
		Apply and Close	Ca	ancel

Perspective

After the test has created, you can set the opening option for the perspective associated. Select one among [Always open], [Never open] and [Prompt] and click [Apply].

Suresofttech

Preferences		- 🗆 X
type filter text	Perspective	← → ⇒ → →
 Analysis Editor Exclusion Language Performance Perspectives 	Open the associated perspective when creating a O Always open O Never open O Prompt	new unit test
Register external comman Source File Types > Target Test		
 larget lest Test Coverage Export test data External editor Perspective Type Partition Unit Test View Virtual Address Theme (CS Labs) ToolChain key 		
< >>	Restore	Defaults Apply
	Apply and	Close Cancel

15.7. Toochain

You can set the information for the toolchain to be used in the tool.

You must have a toolchain (compiler information) for the source to be tested in order to create or analyze the project.

The detail description of the toolchain settings can be found in the <u>'Set a Toolchain (Analyzer)'</u>.

Export Toolchain

It sets the system header size of the toolchain to be exported. The system headers larger than the size you set cannot be exported.

Preferences		— 🗆 X
type filter text	Exporting	← → ⇒ → →
 Analysis Editor Exclusion Language Performance Perspectives Register external commany Source File Types Target Test Test Theme (CS Labs) ToolChain Exporting key 	Set the system header size of ToolChain to be exported. Toolchain system header size (MB): 500	
< >	Restore [<u>D</u> efaults <u>A</u> pply
	Apply and C	Close Cancel

15.8. Editor

You can change the settings related to the editor.

Preferences	– 🗆 X
type filter text	Editor 🗘 🔹 🖒
 Analysis Editor Exclusion Language Performance Perspectives Register external commany Source File Types Target Test Test Theme (CS Labs) ToolChain key 	Editor settings: Iext file encoding © Default (UTF-8) O Qther: UTF-8 V Automatic detection Indentation Tab size: 4 Detect modifications Ø Check the effect whenever the source file is modified (including external modifications) Market the effect whenever the source file is modified (including external modifications)
< >	Apply and Close Cancel

Text file encoding

You can set the encoding to be used when opening a text file in the editor. If the [Automatic detection] option is on, it will automatically detect the encoding of the text file. (If it cannot be detected automatically, it will open with the encoding you set.)

Indentation

If you change the tab size, the tab size shown in the editor is changed too.

Detect modifications

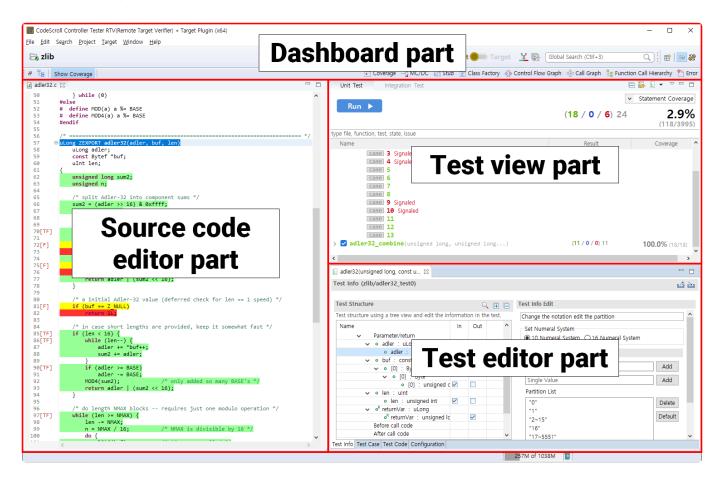
If the [Detect modifications] option is on, it checks the impact of the changes each time the source file is modified.

* If the disc speed if slow, deselect this checkbox to improve the performance.

16. Test Perspective

The Test Perspective provides UI for displaying only the necessary information so as to keep the flow of the testing task and focus on the goal.

It provides a high DOF screen with Eclipse RCP. You can set it in the [Window] menu and in each view.



The components that make up the test perspective are:

- Dashboard
- <u>Test Navigator View</u>
- <u>Unit Test View</u>
- Integration Test View
- <u>Coverage View</u>
- MC/DC View
- <u>Stub View</u>
- <u>Class Factory View</u>
- <u>Control Flow Graph View</u>
- <u>Call Graph View</u>
- Error View
- Debug Information View
- Fault Injection View
- Input/output Data Graph View
- <u>Console View</u>
- Requirement View

Views that are not included by default can be opened from [Window]> [Show View]> [Other...] on the top menu.

16.1. Dashboard

The dashboard section provides a summary of frequently used functions or tests that are currently in progress.



Project name

It shows the name of the project selected in the Test Navigator View.



Bookmark menu

You can use the [New Project], [New Test], [Import], [Export], [Properties] and [Preferences] features.

😂 test	New Project	New Test	🏄 Import	Export	Properties	Preferences	Host 🔵	Target	¥®	Global Search (Ctrl+3)	Q 🖻 🖃
🖶 🦙 🔓 🔜 Show Coverage										àraph 🛛 💀 Call Graph 😰	Fault Injection

Host/Target setting menu

You can change the test environment to the host or target.

😹 test	New Project	EP New Test	🊵 Import	Export	Properties	Preferences	Host 🔵 🗆 T	arget	<u>¥</u> 💀	Global Search (Ctrl+3)		Q 🖻 🖃
🖶 🔓 🔒 Show Coverage			[Cove	rage 🖂 Mo	C/DC 🔡 Stub	Class Factory	🚭 Con	trol Flow Gr	aph 📲 Call Graph 💼	Requirement	- Fault Injection

Coverage-related menu

You can use the Show the [Merger coverage between host and target] and the [Show full coverage (Include External Coverage)] features.

😂 test	New Project	EP New Test	놀 Import	Z Export	Properties Preferences	Host 🔵 🗆 🕯	Target Y 🔀 Global Search (Ctrl+3) Q	
🖶 🧞 Show Coverage				Cover	rage 🖂 MC/DC 🔃 Stub	🔋 👸 Class Factory	ry 🚭 Control Flow Graph 😳 Call Graph 😰 Requirement 📲	Fault Injection

Open Perspective menu

You can select and open the perspective located in Controller Tester

😂 test	New Project	EP New Test	2 Import	A Export	Properties	Preferences	Host 🔵	Target	<u>¥</u> 💀	Global Search (Ctrl+3)	Q) 🖬 🖃
🖶 🧤 🔜 Show Coverage				Cover	age 🖂 M	C/DC 🔡 Stub	👸 Class Fact	ory 👍 Co	ntrol Flow G	aph 📲 Call Graph 💼 R	equirement 📲 Fault Injection

Global Search

The Global Search allows you to search New, Views, Editor, Menus, Properties, Preferences, Project resource (source file, function), Test, Stub and Class Factory, etc. by keywords. Select the Global Search window or press the shortcut key (Ctrl + 3) to use the Global Search feature.

Properties P	🕸 Host 🔵 Target 👱 💽 Global Search (Ctrl+3) Q	B
Views	💷 Bookmarks (General) aul	lt Injection
	🖶 Call Graph (Analysis)	-
	🕮 Cheat Sheets (Help)	
	Class Factory (Test)	
	🔗 Classic Search (General)	
Perspectives	s 🦃 Analysis	
	to External Test	
	≕⁄ Test	
Menus	About Controller Tester	
	Exit - Exit Workbench	
	🖾 Export Export	
	Generate command line interface INI file	
	🔤 Import Import	
New	C/C++ Project from Embedded(CodeWarrior, Green Hills, NEC) - Create a new project	
	💕 C/C++ Project from Existing CodeScroll Project - Create a new project from an existir	
	C/C++ Project from RTV Build - C/C++ Project from RTV Build	
	🔚 C/C++ Project from RTV Build Command 🛛 - C/C++ Project from RTV Build Command	
	🗭 C/C++ Project from Visual Studio Project - Create C/C++ Project from Visual Studio	
Preferences	Analysis	
	CFG - Analysis	
	Call Graph - Analysis	
	Coverage - Test	
	Results per category are limited. Press 'Ctrl+3' to see all	

The project resource (source file, function, etc.), test, stub and class factory can be searched by selecting the analyzed project.

Perspectives	🦃 Analysis	
	test 🐻 External Test	
	⇒⁄ Test	
Menus	About Controller Tester	
	Close Project - Close Project	
	🔀 Delete - Delete	
	Exit - Exit Workbench	
	🖾 Export Export	
New	C/C++ Project from Embedded(CodeWarrior, Green Hills, NEC) -	Create a new project from CodeWarrior(*.xml), Green Hills MULTI(*.gpj) or N
	C/C++ Project from Existing CodeScroll Project - Create a new pro	project from an existing CodeScroll project file(*.csp).
	C/C++ Project from RTV Build - C/C++ Project from RTV Build	
	C/C++ Project from RTV Build Command - C/C++ Project from	n RTV Build Command
	C/C++ Project from Visual Studio Project - Create C/C++ Project	t from Visual Studio Project. (*.dsw, *.sln, *.vcxproj, *.vcproj)
Class Factory	Abstract - classFactory.cpp	
	O Abstract2 - classFactory.cpp	
	InfoLoader - classFactory.cpp	
	Processor - classFactory.cpp	
	G Timer - classFactory.cpp	
Project	Concrete::Concrete() - classFactory.cpp	
	Concrete::func2() - classFactory.cpp	
	InfoLoader::infoLoader(const char *) - classFactory.cpp	
	InfoLoader::getData() - classFactory.cpp	
	Processor::Processor(InfoLoader, Timer) - classFactory.cpp	
Stub	📫 Concrete_1 - classFactory.cpp	
Unit Test	E Concrete::Concrete()_0 - classFactory.cpp	
	E Concrete::func2()_0 - classFactory.cpp	
	E InfoLoader::InfoLoader(const char *)_0 - classFactory.cpp	
	🔄 InfoLoader::getData()_0 - classFactory.cpp	

Results per category are limited. Press 'Ctrl+3' to see all

Show Coverage

It allows you to set the covered area in the source code editor is marked or not.



Open/Close the test-related view menu

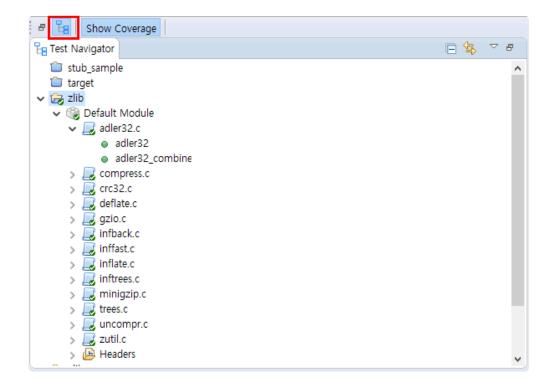
You can open and close the test-related views.

😂 test	New Project	Rew Test	1mport	A Export	Properties	Preferences	Host	Target	¥ 🚱 (Global Search (Ctrl+3)) (2 🖻 🖃
🖶 🧏 🔜 Show Coverage				Cover	rage 🖂 Mo	C/DC 🔡 Stub	📋 Class Fact	ory 👍 Co	ntrol Flow Gra	aph 📲 Call Graph 💈	🗄 Requirement 🍷	Fault Injection

16.2. Test Navigator View

The Test Navigator View shows the hierarchy structure of the projects and the test models under the projects included in the workspace.

The Test Navigator View does not occupy any space but remains the minimized state. If you select [Test Navigator] in the dashboard, the Test Navigator View is located above the Source Code Editor.



lcon

lcon	Description(*: created after analysis)
æ	Open project
Ê	Closed project
٢	Module
	Source file
۲	*Global variable
•	*Function
Þ	*Header folder
.h	*Header file
ĥ	*System header file

Toolbar menu

Menu	Description
5 Link with Selection	Highlights the items selected in the other view.
Collapse All	Hides all tree nodes.

Icon overlay

Icon overlay	Description
~	Analyzed
•	Changed after analysis

Copy the source file into the other module

To copy the source file into the other module, with Ctrl key pressed, drag and drop the source file to be copied into the target module.

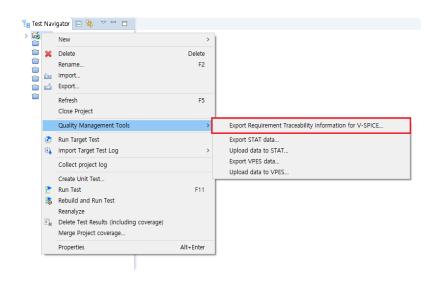
Alternatively, right-click the source file to be copied, select [Copy], right-click the target module and select [Paste]. (Shortcut keys: Ctrl + C and Ctrl + V)

Move the source file to the other module

To move the source file to the other module, drag and drop the source file to be moved into the target module.

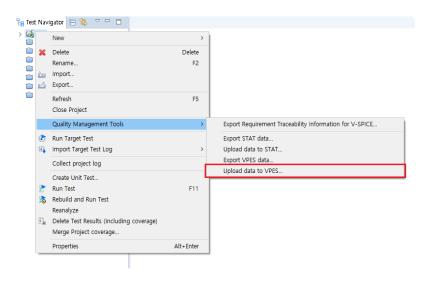
Export Requirement Traceability Information for V-SPICE

Connection information between requirements and tests can be exported to an xml file that conforms to the V-SPICE format.



Select a project and right-click on the menu to select [Quality Management Tools] > [Export Requirements Traceability Information for V-SPICE...] function.

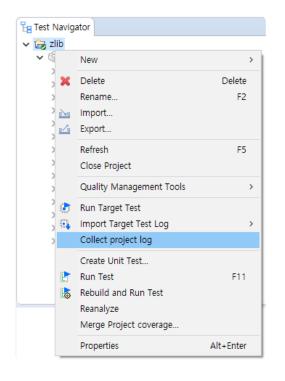
Upload data to VPES



When you select and right-click a project, [Upload data to VPES...] is indicated in the menu. For this feature, please contact <u>Technical support</u>.

Collect Project Log

This feature collects the logs left in use of Controller Tester on a project basis in batch and exports them. When you contact Technical support due to an error occurred, please send us the collected compressed file so that we can solve the problem more quickly.



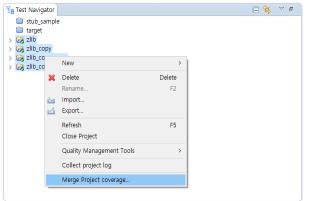
When you select and right-click a project, [Collect project log] item is indicated.

16.2.1. Merge Project Coverages

If a project size is too large to run the test at once, you can run the test by dividing it into multiple projects. [Merge Project Coverage] function runs the test by dividing one project into multiple projects and then shows the overall coverage results by merging the result of the test for these projects.

Merge Project Coverage

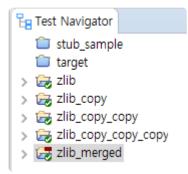
1. After selecting all the projects that you want to merge the coverage results, right-click it and select [Merge Project Coverage].



2. Select [Merge Project Coverage] menu to open for creating the project name. Enter the project name to be created in [Project Name].

Merge project coverage results	×
Merge project coverage results	
Add project's db files(*.csp) to merge. The merged project cannot create and run test.	
Project Name: zlib_merged	
Add project db files to merge:	
P:#Product_Workspace#ct_3_1_02#zlib#.csdata#zlib.csp	<u>A</u> dd
P:\Product_Workspace\Ct_3_1_02\Zilb_copy\Csdata\Zilb_copy.cs P:\Product_Workspace\Ct_3_1_02\Zilb_copy_copy\Csdata\Zilb_ci	Pomovo
P:#Product_Workspace#ct_3_1_02#zlib_copy_copy#.csdata#	
< >	
ОК	Cancel

- By clicking the [Add] button, you can perform a merge coverage function for projects not included in the workspace.
- 3. Click [OK] to check that a new project that the project results had been merged has created in the Test Navigator view. The merged project is displayed with a red mark on the upper right corner.



4. Select the project that the results had been merged and check the coverage view to see the overall coverages.

overage	information of 'zlib' project						Host coverage infor	mation of 'zlib_copy' p					
_	Target Function	Statement -	Branch	MC/DC	Function Cal	Function		Target Function	Stateme	nt Branc	h MC/DC	Function (Call Functio
	lign(struct internal_state *)			- 0.00% (0			-	truct internal_state *)			(0 = 0.00% (0		
	lush_block(struct internal_state * nit(struct internal state *)	0.00% (0	-	= 0.00% (0	0.00% (0.	N		lock(struct internal_stat uct internal_state *)		6 (= 43.75 % 📼 N/A	% (= 0.00% (0	= 70.00%	
				-0.00% (0		. – N		block(struct internal_st			(0 = 0.00% (0		
	ally(struct internal_state *, unsig					- N		ruct internal_state *, un signed long, const uns					- N
	r32(unsigned long, const unsign r32_combine(unsigned long, un					-Y		ombine(unsigned long,	-				- N
_	ush(struct internal_state *)			- 0.00% (0		- N		ruct internal_state *)			(0 = 0.00% (0		- N
	everse(unsigned int, signed int) indup(struct internal state *)			- 0.00% (0 - 0.00% (0		- N		unsigned int, signed in (struct internal_state *)			% (= 0.00% (0 % (= 0.00% (0		= Y
				- 0.00% (0		— N		ee(struct internal_state	*) = 100.00	% = 75.00	% (== 50.00% ((💻 100.00	% — Y
	d_tree(struct internal_state *, stru	- 0.00% (0		0.00% (0			12 build_tree(struct internal_state *, s	- 0.00%			- 0.009/	/0 - N
tal				0.95% (1			Total		1		(1 3.19% (4.		
	t testing mode, the coverage of	f the function	including Asm	i code can not			In target testi	ng mode, the coverag	ge of the funct	ion including	Asm code can n		
overage (coverage i	information of 'zlib_copy_copy' p	proiect					Coverage 😒	mation of 'zlib_copy_co	nov conv' proje	+		t	E 💽 🗖
	Target Function	Statement	Branch	MC/DC	Function Cal	Function ^		Target Function	Stateme		h MC/DC	Function	Call Function
3 check	la baardaa (abaarda amada amada amada am	-	-	-				a , const char ,	0.00%		•	•	• •
	k_header(struct gz_stream *) press(unsigned char *, unsigned			= 0.00% (0 N/A	0.00% (0		70 gzread(voi	d *, void *, unsigned in			(0 = 0.00% (0		
	press2(unsigned char *, unsigne						71 gzrewind(v 72 gzseek(voi	/oid *) d *, signed long, signed			(0 = 0.00% (0		
	pressBound(unsigned long) press_block(struct internal_state	0.00% (0				= N	2	ns(void *, signed int, sig					
	_block(struct internal_state *, ch						74 gztell(void			(0 N/A	□ N/A	-0.00%	(0 = N
	2(unsigned long, const unsigne							signed int, void *) id *, const void *, unsic		•			
D crc32	2_big(unsigned long, const unsi				N/A 100.00%	- Y Y		ct z_stream_s *, signed					
1 crc32	2 combine(unsigned long, unsi	100.00%	- TUU.UU %										
2 crc32	2_combine(unsigned long, unsi 2_little(unsigned long, const un	— 100.00%	— 100.00%	— 100.00%	N/A	- Y		(struct z_stream_s *, ur End/struct z stream_s *					
2 crc32 3 deflat	2_little(unsigned long, const un te(struct z_stream_s *, signed int)	= 100.00% = 0.00% (0	= 100.00% = 0.00% (0	= 100.00% = 0.00% (0	□ N/A ■ 0.00% (0	- Y N	79 inflateBack	:(struct z_stream_s *, ur :End(struct z_stream_s * :Init_(struct z_stream_s	r) = 40.009	5 (== 50.009	% (💻 0.00% (0	= 0.00%	(0 — Y
2 crc32 3 deflat 4 deflat	2_little(unsigned long, const un	 100.00% 0.00% (0 0.00% (0 		 100.00% 0.00% (0 0.00% (0 	N/A 0.00% (0 0.00% (0	Y N N	79 inflateBack 80 inflateBack 81 inflateCop	End(struct z_stream_s *	r) = 40.009 *, s = 13.639 ru = 0.00%	6 (= 50.009 6 (= 10.009 (0 = 0.00%	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0	0 = 0.00% = 0.00%	(0 - Y (0 - Y (0 - N
2 crc32 3 deflat 4 deflat tal	2_little(unsigned long, const un te(struct z_stream_s *, signed int)	 100.00% 0.00% (0 0.00% (0 3.05% (1 	100.00% 0.00% (0 0.00% (0 1.93% (4	 100.00% 0.00% (0 0.00% (0 1.67% (2 	 N/A 0.00% (0 0.00% (0 2.13% (8 	Y N N 6.14 ↓	79 inflateBack 80 inflateBack 81 inflateCop Total	End(struct z_stream_s * init_(struct z_stream_s	r) = 40.009 *, s = 13.639 rru = 0.00% 0.65% (6 (= 50.009 5 (= 10.009 (0 = 0.00% 2 0.14% (% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 deflat tal deflat	2_little(unsigned long, const un te(struct z_stream_s *, signed int) teBound(struct z_stream_s *, un teConv(struct z_stream_s *, un	 100.00% 0.00% (0 0.00% (0 3.05% (1 	100.00% 0.00% (0 0.00% (0 1.93% (4	 100.00% 0.00% (0 0.00% (0 1.67% (2 	 N/A 0.00% (0 0.00% (0 2.13% (8 	Y N N 6.14 ↓	79 inflateBack 80 inflateBack 81 inflateCop Total	End(struct z_stream_s + dnit_(struct z_stream_s y(struct z_stream_s *, st ng mode, the coverag	r) = 40.009 *, s = 13.639 rru = 0.00% 0.65% (6 (= 50.009 5 (= 10.009 (0 = 0.00% 2 0.14% (% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 deflat fal deflat n target	2_little(unsigned long, const un te(struct _stream_s * signed int) te8ound(struct z_stream_s *, un te6ound(struct z_stream_s *, tru testing mode, the coverage of	 100.00% 0.00% (0 0.00% (0 3.05% (1 the function is 	 100.00% 0.00% (0 0.00% (0 1.93% (4 including Asm 	 100.00% 0.00% (0 0.00% (0 1.67% (2 code can not 	 N/A 0.00% (0 0.00% (0 2.13% (8 	Y N N 6.14 ↓	79 inflateBack 80 inflateBack 81 inflateCop Total	End(struct z_stream_s + dnit_(struct z_stream_s y(struct z_stream_s *, st ng mode, the coverag	*) = 40.009 *, s = 13.639 ru = 0.00% 0.65% (ge of the funct	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 deflat 5 deflat 5 deflat 6 deflat 6 deflat 6 deflat	2_little(unsigned long, const un te(struct z_stream_s *, signed int) ite8ound(struct z_stream_s *, un te construct z_stream_s *, un te testing mode, the coverage of erage &	 100.00% 0.00% (0 0.00% (0 3.05% (1 the function i 'zlib_merg 	 100.00% 0.00% (0 0.00% (0 1.93% (4 including Asm 	 100.00% 0.00% (0 0.00% (0 1.67% (2 code can not 	 N/A 0.00% (0 0.00% (0 2.13% (8 be measure 	Y N N 6.14 ↓	79 inflateBack 80 inflateBack 81 inflateCop Total	End(struct z_stream_s + dnit_(struct z_stream_s y(struct z_stream_s *, st ng mode, the coverag	*) = 40.009 *, s = 13.639 ru = 0.00% 0.65% (ge of the funct	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
crc32 deflat deflat deflat deflat deflat tal deflat	2_little(unsigned long, const un te(struct z_stream_s *, signed int) iteBound(struct z_stream_s *, un teConstitute terms + the stating mode, the coverage of erage recage information of	 100.00% 0.00% (0 0.00% (0 3.05% (1 the function i 'zlib_merg 	 100.00% 0.00% (0 0.00% (0 1.93% (4 including Asm 	 100.00% 0.00% (0 0.00% (0 0.00% (2 code can not 	 N/A 0.00% (0 0.00% (0 2.13% (8 be measure 	Y N 6.14 ↓ d.	79 inflateBack 80 inflateBack 81 inflateCop Total	End(struct z_stream_s * Init_(struct z_stream_s v(struct z_stream_s *, st ng mode, the coveraç	*) = 40.009 *, s = 13.639 ru = 0.00% 0.65% (ge of the funct	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 deflat 5 deflat 5 deflat 6 deflat 6 deflat 6 deflat	2_little(unsigned long, const un te(struct z_stream_s *, signed int) iteBound(struct z_stream_s *, un teConstitute terms + the stating mode, the coverage of erage recage information of	 100.00% 0.00% (0 0.00% (0 3.05% (1 the function i 'zlib_merg nction 	= 100.00% = 0.00% (0 = 0.00% (0 = 0.00% (0 1.93% (4 including Asm ged' proje	= 100.00% = 0.00% (0 = 0.00% (0 = 0.00% (2 1.67% (2 code can not ect Stateme	N/A 0.00% (0 0.00% (0 0.00% (0 2.13% (8 be measure ent ✓	PY N 6.14 v d. Branch	79 inflateBack 80 inflateBack 81 inflateBack 91 inflateCop Total © In target testi	End(struct z_stream_s ' init_(struct z_stream_s y(struct z_stream_s *, st ing mode, the coverage the coverage Function Call	+) = 40.009 +, s = 13.639 ru = 0.00% 0.65% (ge of the funct Function	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 deflat tal deflat tal deflat tal deflat tal deflat tal deflat tal deflat	2_little(unsigned long, const un te(struct z_stream_s * signed int) iteBound(struct z_stream_s *, un teresting mode, the coverage of erage & verage information of Target Fu	= 100.00% = 0.00% (0 = 0.00% (0 3.05% (1 the function i 'zlib_merg nction ct interna	= 100.00% = 0.00% (0 = 0.00% (0 1.93% (4 including Asm ged' proje	= 100.00% = 0.00% (0 = 0.00% (0 = 0.00% (2) 1.67% (2) code can not ect Stateme = 0.00%	N/A ● 0.00% (0. ● 0.00% (0. ● 0.00% (0. 2.13% (8) be measure ent (0 ● 0	PY N 6.14 V d. Branch V 0.00% (0	79 inflateBack 80 inflateBack 81 inflateCop Total C In target testi	End(struct z_stream_s ' sinit_(struct z_stream_s y(struct z_stream_s *, st ng mode, the coverac Function Call 	+) = 40.009 +, s = 13.639 ru = 0.00% 0.65% (ge of the funct Function	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 deflat deflat deflat tal cover 5 cover 4 5 6	2_little(unsigned long, const un te(struct _stream_s *, signed int) te8ound(struct z_stream_s *, un testing mode, the coverage of erage & rerage information of Target Fu _tr_stored_block(struct _tr_tally(struct interna adler32(unsigned lor	= 100.00% (0 = 0.00% (0 = 0.00% (0 3.05% (1 'zlib_merg inction ct interna al_state *, ng, const	= 100.00% (C = 0.00% (C = 0.00% (C 1.93% (4 including Asm ged' proje [= 100.00% (0 = 0.00% (0 = 0.00% (0 1.67% (2 code can not ect Stateme = 0.00% = 0.00% = 98.05%	N/A 0.00% (0 0	PY N 6.14 v d. Branch 0.00% (0 0.00% (0 37.50% (79 inflateBack 80 inflateBack 81 inflateCop Total ImplateCop ImplateCop ImplateCop ImplateCop	End(struct z_stream_s ' init_(struct z_stream_s y(struct z_stream_s +, st ng mode, the coverage Internet in the coverage	•) = 40.009 •, s = 13.639 • 0.00% • 0.65% (ge of the funct Function • N	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 deflat tal cover st cover 4 5 6 7	2_little(unsigned long, const un te(struct z_stream_s *, signed int) te8ound(struct z_stream_s *, un testing mode, the coverage of erage & rerage information of Target Fu _tr_stored_block(struct _tr_tally(struct interna adler32_(unsigned lon adler32_combine(un:	= 100.00% (0 = 0.00% (0 = 0.00% (0 3.05% (1 'zlib_merg nction ct interna al_state *, ng, const signed lor	= 100.00% (C = 0.00% (C = 0.00% (C = 0.00% (C = 1.93% (4 including Asm ged' proje [= 100.00%. = 0.00% (0 = 0.00% (0 1.67% (2 code can not stateme = 0.00% = 0.00% = 0.00% = 0.00% = 100.00% = 100.00% = 100.00%	N/A 0.00% (0 0.00\% (0	Pranch 6.14 v d. Branch 0.00% (0 0.00% (0 37.50% (00.00% (79 inflateBack 80 inflateBack 81 inflateCop Total ImplateCop ImplateCop ImplateCop ImplateCop	End(struct z_stream_s ' init_(struct z_stream_s y(struct z_stream_s +, st ing mode, the coverage intermediate in the coverage intermediate in the coverage intermediate in the coverage intermediate in the coverage intermediate intermediate intermediate intermediate intermediate) = 40.009 , s = 13.639 ru = 0.00% 0.65% (ge of the funct Function N N N	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 deflat fall deflat cover st cover 4 5 6 7 8	2_little(unsigned long, const un tec(struct z_stream_s *, signed lint) tet8ound(struct z_stream_s *, un testing mode, the coverage of erage & rerage information of Target Fu _tr_stored_block(struct _tr_tally(struct internation) adler32_combine(un: bi_flush(struct internation)	 100.00% 0.00% (0 0.00% (0 0.00% (0 0.00% (0 0.00% (0 0.00% (0 1.10% (0.00%) 1.10% (0.00%)<td>= 100.00% (C = 0.00% (C = 0.00% (C = 0.00% (C = 1.93% (4 including Asm ged' proje [_state = unsig = unsig = ng, un =</td><td>= 100.00% (0 = 0.00% (0 = 0.00% (2 code can not cct Stateme = 0.00% = 0.00% = 98.05% = 100.00% = 0.00%</td><td>N/A 0.00% (0 0.00% (0 0</td><td>PY N 6.14 v d. Branch 0.00% (0 37.50% (100.00% (0 30.00% (0</td><td>79 inflateBack 80 inflateBack 81 inflateCop Total © In target testi InflateCop 0.00% (0 0.00% (0 100.00% (0.00% (0 0.00% (0 0.00% (0</td><td>End(struct z_stream_s ' init_(struct z_stream_s y(struct z_stream_s +, st ing mode, the coverage intermediate in the coverage Function Call Co.00% (0 N/A N/A N/A N/A N/A</td><td>) = 40.009 , s = 13.633 ru = 0.00% 0.65% (ge of the funct Function V N N V V N N N</td><td>5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including</td><td>% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.</td><td> = 0.00% = 0.00% = 0.00% 0.00% (0</td><td>(0 Y (0 Y (0 N 0 3.50</td>	= 100.00% (C = 0.00% (C = 0.00% (C = 0.00% (C = 1.93% (4 including Asm ged' proje [_state = unsig = unsig = ng, un =	= 100.00% (0 = 0.00% (0 = 0.00% (2 code can not cct Stateme = 0.00% = 0.00% = 98.05% = 100.00% = 0.00%	N/A 0.00% (0 0.00% (0 0	PY N 6.14 v d. Branch 0.00% (0 37.50% (100.00% (0 30.00% (0	79 inflateBack 80 inflateBack 81 inflateCop Total © In target testi InflateCop 0.00% (0 0.00% (0 100.00% (0.00% (0 0.00% (0 0.00% (0	End(struct z_stream_s ' init_(struct z_stream_s y(struct z_stream_s +, st ing mode, the coverage intermediate in the coverage Function Call Co.00% (0 N/A N/A N/A N/A N/A) = 40.009 , s = 13.633 ru = 0.00% 0.65% (ge of the funct Function V N N V V N N N	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 deflat 4 deflat 4 deflat 5 covel 4 5 6 7 8 9	2_little(unsigned long, const un tectstruct z_stream_s *, signed int) tectstruct z_stream_s *, un testing mode, the coverage of erage S rerage information of Target Fu _tr_stored_block(struct _tr_tally(struct internation) adler32_combine(un: bi_flush(struct internation) bi_reverse(unsigned internation)	 100.00% (0 0.00% (0 0.00% (0 0.00% (0 0.00% (0 1.00% (0 0.00% (0 1.00% (0 1.00%	= 100.00% (C = 0.00% (C)) (C = 0.00% (C)) (C = 0.00% (C)) (C = 0.00% (C)) (C) (= 100.00% (0 = 0.00% (0 = 0.00% (2 code can not code can not = 0.00% = 0.00% = 0.00% = 98.05% = 100.00% = 100.00% = 100.00%	N/A 0.00% (0 0.00% (0 0	PY N 6.14 v d. Branch 0.00% (0 37.50% (100.00% (0 30.00% (0 50.00% (0	79 inflateBack 80 inflateBack 81 inflateCop Total © Intarget testi InflateCop 0.00% (0 0.00% (0 100.00% (0.00% (0 0.00% (0 0.00% (0 0.00% (0 0.00% (0	End(struct z_stream_s ' init_(struct z_stream_s y(struct z_stream_s +, st ing mode, the coverage intermediate in the coverage intermediate in the coverage intermediate in the coverage intermediate in the coverage intermediate intermediate intermediate) = 40.009 , s = 13.633 ru = 0.00% 0.65% c ge of the funct Function N N N Y Y N Y Y	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 deflat 4 deflat 4 deflat 4 deflat 5 cover 4 5 6 7 8 9 10	2_little(unsigned long, const un tec(structstream_s *, signed lint) tec(structstream_s *, signed lint) testing mode, the coverage of erage & rerage information of Target Fu _tr_stored_block(struct _tr_tally(struct interna adler32(unsigned lor adler32_combine(un: bi_flush(struct interna bi_reverse(unsigned i bi_windup(struct inter	100.00% (0 0.00% (0 3.05% (1 'zlib_mere nction ct interna al_state *, ng, const signed loi al_state *) int, signed ernal_state	= 100.00% (C = 0.00% (C) = 0.00% (C = 0.00% (C) =	= 100.00% (0 = 0.00% (0 = 0.00% (0 = 0.00% (2 code can not = 0.00% = 0.00% = 0.00% = 98.05% = 100.00% = 0.00% = 100.00% = 0.00% = 100.00% = 0.00%	N/A 0.00% (0 0.00% (0 0.	PY N 6.14 v d. Branch 0.00% (0 37.50% (100.00% (0 50.00% (0 50.00% (79 inflateBack 80 inflateBack 81 inflateBack 81 inflateCop Total © 0 inflateCop 0 0.00% 0 0.00% 100.00% (0.00% (0.00% (0.00% (0.00% (0.00% (0.00% (0.00% (0 0.00% (0	End(struct z_stream_s ' init_(struct z_stream_s y(struct z_stream_s - init_cistuct z_stream_s -, st ing mode, the coverage intervention Call Function Call 0.00% (0 N/A N/A N/A N/A N/A N/A N/A	 * 40.009 * 40.009 * 5 * 13.633 ru * 0.00% • 0.65% • 0.65% * 0.65% * 0.65% * N * N * N * N * Y * Y * Y * Y * Y * Y 	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat deflat cover cover 4 5 6 7 8 9 10 11	2_little(unsigned long, const un tec(structstream_s *, signed int) tec(structstream_s *, signed int) teconditutet _stream_s *, un terage & rerage information of Target Fu _tr_stored_block(struct _tr_tally(struct interna adler32(unsigned lor adler32_combine(un: bi_flush(struct interna bi_reverse(unsigned i bi_windup(struct inter build_bl_tree(struct inter	100.00% (0 0.00% (0 3.05% (1 3.05% (1 'zlib_merg nction ct interna al_state *, ng, const signed lor al_state *) int, signed ernal_state	= 100.00% = 0.00% (0 = 0.00% (0) = 0.00% (0.	100.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0.00% 0.00% 0.00% 0.00% 100.00% 100.00% 100.00% 100.00%	N/A 0.00% (0 0.00% (0 0 0.00% (0 0	PY N 6.14 v d. Branch 0.00% (0 37.50% (100.00% (0 50.00% (50.00% (75.00% (79 inflateBack 80 inflateBack 81 inflateBack 81 inflateCop Total © MC/DC 0.00% (0 0.00% (0 0.00% (0 100.00% (0 0.00% (0 0.00% (0 0.00% (0 0.00% (0 50.00% (End(struct z_stream_s ' init_(struct z_stream_s y(struct z_stream_s - init_carried to the coverage init_ coverage Function Call Coverage Function Call Coverage Cover	 • 40.009 • 5 = 13.633 • 13.633 • 0.00% • 0.65% • 0.65% • N • N • N • N • Y 	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat and and and and and and and and and and	2_little(unsigned long, const un te(struct z_stream_s *, signed int) tetestruct z_stream_s *, un testing mode, the coverage of erage & verage information of Target Fu _tr_stored_block(struct _tr_tally(struct internation) adler32_combine(un: bi_flush(struct internation) bi_reverse(unsigned into bi_windup(struct internation) bi_windup(struct internation) bi_tally(struct internation) bi_tally(struct internation) bi_windup(struct internation) bi_tally(struct internation) bi_tally(struct internation) bi_windup(struct internation) bi_tally(struct internation) bi_tally(struct internation) bi_tally(struct internation) bi_tally(struct internation)	 100.00% (0 0.00% (0 0.00% (0 3.00% (1 'zlib_merg nction ct interna al_state *, ng, const signed loi al_state *) int, signed ernal_state nternal_state 	= 100.00% (0 = 0.00% (0 = 0.00% (0 1.93% (4) including Asm ged' proje (1.state unsig unsig unsig) d int) = *) = *) *, stru	100.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0.00% 0.00% 0.00% 0.00% 100.00 71.42% 100.00 100.00	N/A 0.00% (0 0.00\% (0 0.00\% (0 0.00\% (0 0.00\% (0 0.00\% (0 0.00\% (0 0	PY N Contractions Branch Branch 0.00% (0 0.00% (0) 0.00% (0)	79 inflateBack 80 inflateBack 81 inflateBack 81 inflateCop Total © 0 0.00% 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% (0) 0.00% () 0.00% () 0.00% () 0.00% ()	End(struct z_stream_s ' init_(struct z_stream_s y(struct z_stream_s) init_(struct z_stream_s) in)) = 40.009 y, s = 13.633 ru = 0.00% 0.65% (ge of the funct Function V N N N V V V V V V V V V V V V V	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 de	2_little(unsigned long, const un tet(struct z_stream_s * signed int) tetesting mode, the coverage of erage & verage information of Target Fu _tr_stored_block(struct _tr_tally(struct internation) adler32_combine(un) bi_flush(struct internation) bi_windup(struct internation) bi_windup(struct internation) bi_windup(struct internation) bi_windup(struct internation) bi_tallab_tree(struct internation) check_header(struct internation)	= 100.00% (0 = 0.00% (0 = 0.00% (0 3.05% (1 'zlib_merg nction ct interna al_state *, ng, const signed lou al_state *) int, signed ernal_state thernal_state rnal_state gz_stream	= 100.00% (C = 0.00% (C = 0.00% (C 1.93% (4 including Asm ged' proje (1.53% (4) (1.93% (4)) (1.93% (4) (1.93% (4)) (1.93%	= 100.00% (0 = 0.00% (0 = 0.00% (0 = 0.00% (0 = 0.00% (0 = 0.00% = 0.00% = 0.00% = 0.00% = 0.00% = 0.00% = 100.00% = 100.00% = 100.00% = 0.00%	N/A 0.00% (0 0	Y N S.14	79 inflateBack 80 inflateBack 81 inflateBack 81 inflateBack 81 inflateCop Total © 0 0.00% 0 0.00% 0 0.00% 100.00% 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0	End(struct z_stream_s ' init_(struct z_stream_s y(struct z_stream_s) init_(struct z_stream_s) in)) = 40.009 y, s = 13.633 = 0.00% 0.65% (= 0.00%) 0.65% (= 0.00%)	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 deflat 1 de	2_little(unsigned long, const un tete(struct z_stream_s * signed int) tetesting mode, the coverage of erage & rerage information of Target Fu 	= 100.00% (C = 0.00% (C = 0.00% (C 3.05% (1 'zlib_merg inction ct interna al_state *, ng, const signed lon al_state *) int, signed ernal_state rnal_state gz_stream char *, un	= 100.00% (C = 0.00% (C = 0.00% (C 1.93% (4 including Asm ged' proje (1.93% (4) unsig unsig unsig) d int) = *) ate *) isigned	= 100.00% (0 = 0.00% (0 = 0.00% (0 = 0.00% (0 = 0.00% (0 = 0.00% = 0.00% = 0.00% = 0.00% = 0.00% = 100.00% = 100.00%	N/A 0.00% (0 0	PY N 6.14 V 6.14 V 6.14 V 6.14 V 6.14 V 6.14 V 0.00% (0 0.00% (0 0.00% (0 0.00% (0 0.00% (0.00% (0.00% (0.00% (0.00% (0.00% (0.00% (0.00% (0 0.00% (0.00% (0.00% (0 0.00% (0.00% (0.00% (0 0.00% (0.00% ())	79 inflateBack 80 inflateBack 81 inflateBack 81 inflateCop Total InflateCop 0 0.00% 0 0.00% 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 N/A N/A	End(struct z_stream_s ' init_(struct z_stream_s y(struct z_stream_s) init_(struct z_stream_s) in)) = 40.009 y, s = 13.633 = 0.00% 0.65% (= 0.00%) 0.65% (= 0.00%)	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50
2 crc32 3 deflat 4 deflat 1 defla	2_little(unsigned long, const un tet(struct z_stream_s * signed int) tetesting mode, the coverage of erage & verage information of Target Fu _tr_stored_block(struct _tr_tally(struct internation) adler32_combine(un) bi_flush(struct internation) bi_windup(struct internation) bi_windup(struct internation) bi_windup(struct internation) bi_windup(struct internation) bi_tallab_tree(struct internation) check_header(struct internation)	 100.00% (C 0.00% (C 0.00% (C 0.00% (C 3.05% (1 'zlib_merg ithe function i 'zlib_merg inction ct interna al_state *, ng, const signed lon al_state *, ng, const signed lon al_state *, int, signed char s, un char *, u 	= 100.00% (C = 0.00% (C = 0.00% (C 1.93% (4 including Asm ged' proje (1.93% (4 unsig unsig unsig) d int) = *) ate *) isigned insigne	= 100.00% (0 = 0.00% (0 = 0.00% (0 = 0.00% (0 = 0.00% (0 = 0.00% = 0.00% = 0.00% = 0.00% = 0.00% = 100.00% = 100.00%	N/A 0.00% (0 0	PY N 6.14 V 6.14 V 6.14 V 6.14 V 6.14 V 0.00% (0 0.00% (0) 0.00% (0)	79 inflateBack 80 inflateBack 81 inflateBack 81 inflateCop Total InflateCop 0 0.00% 0 0.00% 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 0.00% 0 N/A N/A	End(struct z_stream_s ' init_(struct z_stream_s y(struct z_stream_s) init_(struct z_stream_s) in)) = 40.009 y, s = 13.633 = 0.00% 0.65% (= 0.00%) 0.65% (= 0.00%)	5 (= 50.00% 5 (= 10.00% (0 = 0.00% 2 0.14% (fon including	% (= 0.00% (0 % (= 0.00% (0 (0 = 0.00% (0 (3 0.00% (0.	= 0.00% = 0.00% = 0.00% 0.00% (0	(0 Y (0 Y (0 N 0 3.50

The [Merge Project Coverage] function may not work properly when two or more different functions in the project to be merged have the same function name.

16.3. Unit Test View

The Unit Test View section is located on the right of the screen. The Unit Test View combines the test and coverage information and provides it in a single view.

😸 zlib	New Preinet N	Export Export Properties Preferences Host●■ Target Y	Global Search (Ctrl+3)	Q 🖻
En Sh	ow Coverage		w Graph 🐁 Call Graph 隆 Funct	tion Call Hierarchy E
compress.c		Unit Test Integration Test		E 👺 🗄 🔹 🔻 🗖
01 02	D016(buf); /* 16 sums unrolled */			 Statement Coverage
03[TF] 04 05) while (n); MOD(adler); MOD(sum2);	Run 🕨	(655 / 0 / 283) 938	43.0 9
86	}	type file, function, test, state, issue		(1710)000
97 98	/* do remaining bytes (less than NMAX, still just one modulo) */	Name	Result	0
88 29[T]	<pre>/* do remaining bytes (less than NMAX, still just one modulo) */ if (len) {</pre>			Coverage
10[TF]	while (len >= 16) {	> Z > _tr_align(struct internal_state *)	(1 / 0 / 12) 13	72.7% (32/44)
11	len -= 16;	> Z _tr_flush_block(struct internal_state *, char *)	(0 / 0 / 12) 12	54.7% (23/42)
12 13	D016(buf);	> <pre>_tr_init(struct internal_state *)</pre>	(1 / 0 / 0) 1	100.0% (11/11)
13	buf += 16;	> <pre>_tr_stored_block(struct internal_state *, char *)</pre>	(0 / 0 / 8) 8	100.0% (11/11)
15[TF]	while (len) {	> <pre>_tr_tally(struct internal_state *, unsigned int)</pre>	(0 / 0 / 3) 3	22.2% (2/9)
16	adler += *buf++;	> adler32(unsigned long, const unsigned char *)	(9 / 0 / 4) 13	98.0% (101/103)
L7	<pre>sum2 += adler;</pre>	✓ ✓ adler32_combine(unsigned long, unsigned long)	(11 / 0 / 0) 11	100.0% (18/18)
8	} MOD(adler):	Itest adler32_combine(unsigned long, unsigned long, s:		
9	MOD(sum2);	> V bi_flush(struct internal_state *)	(5 / 0 / 6) 11	100.0% (9/9)
1	}	> bi_reverse(unsigned int, signed int)	(3 / 0 / 2) 5	100.0% (5/5)
2	,	> di_windup(struct internal_state *)	(3 / 0 / 5) 8	100.0% (7/7)
3	/* return recombined sums */	> <pre>> build_bl_tree(struct internal_state *)</pre>	(0 / 0 / 8) 8	100.0% (11/11)
4	<pre>return adler (sum2 << 16);</pre>	> <pre>build_tree(struct internal_state *, struct tree_desc_s *)</pre>		100.0% (42/42)
5] 6	ł	> deck_header(struct gz_stream *)	(21 / 0 / 2) 23	44.4% (20/45)
	/**	> Compress(unsigned char *, unsigned long *)	(3 / 0 / 0) 3	100.0% (1/1)
	JLong ZEXPORT adler32 combine(adler1, adler2, len2)	> Compress2(unsigned char *, unsigned long *)	(6 / 0 / 0) 6	95.2% (20/21)
9	uLong adler1;	> CompressBound(unsigned long)	(3 / 0 / 0) 3	100.0% (1/1)
0	uLong adler2;	> Compress_block(struct internal_state *, struct ct_data_s		77.5% (62/80)
1	<pre>z_off_t len2;</pre>	> Copy_block(struct internal_state *, char *)	(0 / 0 / 5) 5	100.0% (9/9)
2	unsigned long sum1;	> Crc32(unsigned long, const unsigned char *)	(6 / 0 / 1) 7	30.4% (7/23)
4	unsigned long sum2;	> Crc32_big(unsigned long, const unsigned char *)	(9 / 0 / 1) 10	100.0% (38/38)
5	unsigned rem;	> Crc32_combine(unsigned long, unsigned long)	(7 / 0 / 2) 9	100.0% (28/28)
6		> 🗹 crc32_little(unsigned long, const unsigned char *)	(9 / 0 / 1) 10	100.0% (36/36)
7	/* the derivation of this formula is left as an exercise for the reader */	> deflate(struct z_stream_s *, signed int)	(27 / 0 / 5) 32	31.4% (56/178)
8 9	<pre>rem = (unsigned)(len2 % BASE); sum1 = adler1 & 0xffff;</pre>	> 🗹 deflateBound(struct z_stream_s *, unsigned long)	(6 / 0 / 0) 6	77.7% (7/9)
9	sum1 = adleri & 0xtttt; sum2 = rem * sum1:	> 🗹 deflateCopy(struct z_stream_s *, struct z_stream_s *)	(0 / 0 / 3) 3	21.8% (7/32)
	MOD(sum2);	> deflateEnd(struct z_stream_s *)	(19 / 0 / 7) 26	94.1% (16/17)
2	<pre>sum1 += (adler2 & 0xffff) + BASE - 1;</pre>	> deflateInit2_(struct z_stream_s *, signed int)	(20 / 0 / 0) 20	84.4% (49/58)
3	<pre>sum2 += ((adler1 >> 16) & 0xffff) + ((adler2 >> 16) & 0xffff) + BASE - rem</pre>	> 🗹 deflateInit_(struct z_stream_s *, signed int)	(5 / 0 / 0) 5	100.0% (1/1)
[TF]	if (sum1 > BASE) sum1 -= BASE;	> 🗹 deflateParams(struct z_stream_s *, signed int)	(12 / 0 / 0) 12	95.2% (20/21)
[TF] [TF]	<pre>if (sum1 > BASE) sum1 -= BASE; if (sum2 > (BASE << 1)) sum2 -= (BASE << 1);</pre>	> deflatePrime(struct z_stream_s *, signed int)	(5 / 0 / 0) 5	80.0% (4/5)
оция ј 7(теј	if (sum2 > (BASE << 1)) sum2 -= (BASE << 1); if (sum2 > BASE) sum2 -= BASE;	> deflateReset(struct z_stream_s *)	(5 / 0 / 0) 5	94.1% (16/17)
3 1	return sum1 (sum2 << 16);	> 🗹 deflateSetDictionary(struct z_stream_s *, const unsigned	i (7 / 0 / 7) 14	92.3% (24/26)
9]	}	> deflateSetHeader(struct z_stream_s *, struct gz_header_s		83.3% (5/6)
0	×	> deflateTune(struct z stream s *, signed int)	(5 / 0 / 0) 5	88.8% (8/9)
	< >>	\ A doflato fact(ctruct internal ctate * cinned int)	(1 / 0 / 10) 11	61 10/ coore o

Toolbar menu in the Unit Test View

Toolbar icon	Description
Expand All	Shows all test view tree.
E Collapse All	Hides all test view trees.
Sort by Unique Test Name	Sort by unique test name.
👺 Show as Unique Test Name	Displays by unique test name.
E Total	Displays all the tests.
■ ^B Failure/Error	Displays only the tests that the test result is Failure/Error.
Failure	Displays only the tests that the test result is Failure.
📾 Error	Displays only the tests that the test result is Error.
F Success	Displays only the tests that the test result is Success.
E Function changed	Displays only the tests that the function to be tested has been changed.
🕒 Run not guaranteed	Displays only the tests that the test execution is not ensured.
+ Host/Target result different	Displays only the test that the host result and the target result are different.

Pull-down menu in the Unit Test View

	🕀 🕞 📮 😓 🕒 🔹 🗖
	Show coverage at all elements
~	Add description when duplicate test case
Ð	Insert Default Values for Empty Input
	Delete All Test case Descriptions
(!)	Check Integrity

Menu name	Description
Show coverage at all elements	Displays the coverage in all items. (function, test, test case)
Add description when duplicate test case	Adds original test case number on duplicated test case.
Insert Default Values for Empty Input	When input data are empty, inserts default values.
Delete All Test case Descriptions	Delete test case descriptions all together.
Check Integrity	Checks integrity about functions and stubs. Then, reset test if it needs.

• Insert Default Values for Empty Input

Variable type	Default value	
string	(empty string)	
without string	0	

For [Test reconfiguration], refer to [Source Code Modification and Test Reconfiguration]

Dashboard in the Unit Test View

Unit Test	Integration Test	•	👺 E 🕶 🗖 🗖
Dura N		~	Statement Coverage
Run 🕨	ļ	(655 / 0 / 283) 938	43.0%
type file, function, te	est, state, issue		

Menu	Description
Run 🕨	Runs unit tests.
(878 / 0 / 359) 1237	Shows success, failure, error and total of test case.

✓ Statement Coverage	Selects the coverage type to be displayed in the Unit Test View. (statement, branch, MC/DC, function call)	
43.7% (1952/4466)	Shows the selected kind of full coverage.	

Search

In the Unit Test View, you can search functions, tests and test cases by file, function, test, status (success, failure, error) and issue name.

Unit Test	Integration Test	E	🛃 🗄 🔹 🗢 🗖
		~	Statement Coverage
Run 🕨		(655 / 0 / 283) 938	43.0% (1719/3995)
type file, function, to	est, state, issue		

Status search keyword

In the Unit Test View, you can filter the tests that are displayed by the status search keyword. It provides searchable keyword recommendation and auto-complete function.

Keyword	Description
%TEST_SUCCESS%	Test success
%TEST_FAILURE% %TEST_ERROR%	Test failure/error
%TEST_FAILURE%	Test failure
%TEST_ERROR%	Test error
%TEST_HAS_NOT_FUNCTION%	Function changed
%TEST_NOT_GUARANTEE%	Run not guaranteed
%TEST_RESULT_DIFFERENT%	Host/Target result different

* 'Run not guaranteed' is a case where test code is generated, but there is no guarantee that it will run normally. This is the case when using a type that is difficult to specify as a parameter or return value.

Structure of the Unit Test View

The Unit Test View presents the hierarchy structure of [Function] > [Test] > [Test case]. If there are more than 100 test cases in a test, they are grouped in groups of 100 to represent a group.

Unit Test	Integration Test		
			✓ Statement Coverage
Run 🕨		(655 / 0 / 283) 9	43.0% (1719/3995)
pe file, functior	n, test, state, issue		
Name		Result	Coverage
🔽 adler32	2_combine(unsigned long, unsigned long)	(11 / 0 / 0) 11	100.0% (18/18)
🗸 🗹 🗆 test	adler32_combine(unsigned long, unsigned long, si	(11 / 0 / 0) 11	
C	ase 1		
C	ase 2		
C	ase 3		
C	ase 4		
C	ase 5		
C	ase 6		
C	ase 7		
C	ase 8		
C	ase 9		
C	ase 10		
C	ase 11		
🛃 bi flus	<pre>sh(struct internal_state *)</pre>	(5 / 0 / 6) 11	100.0% (9/9)
	<pre>bi_flush(struct internal_state *)_0</pre>	(5 / 0 / 6) 11	1001070 (5) 5)
_	ase 1 Signaled		
C	ase 2		
C	ase 3 Signaled		
	ase 4 Signaled		
	ase 5		
	ase 6 Signaled		
	ase 7 Signaled		
	ase 8 Signaled		
	ase 9		
	ase 10		
	ase 11		
	erse(unsigned int, signed int)	(3 / 0 / 2) 5	100.0% (5/5)
_	dup(struct internal_state *)	(3 / 0 / 5) 8	100.0% (3/3)
	bl_tree (struct internal_state *)	(0 / 0 / 8) 8	100.0% (11/11)
	<pre>tree(struct internal_state *, struct tree_desc_s *)</pre>	(0 / 0 / 7) 7	100.0% (42/42)
_	header(struct gz_stream *)	(21 / 0 / 2) 23	44.4% (20/45)
	<pre>55(unsigned char *, unsigned long *)</pre>	(3 / 0 / 0) 3	100.0% (1/1)
	ess2(unsigned char *, unsigned long *)	(6 / 0 / 0) 6	95.2% (20/21)
	ssBound(unsigned long)	(3 / 0 / 0) 3	100.0% (1/1)
	ss block(struct internal state *, struct ct data s		77.5% (62/80)
	lock(struct internal state * char *)	(0 / 0 / 5) 5	100.09/ (0/0)

Item icons in the Unit Test View

Item icon	Description
None	Function
test	Test
case	Test case, Test case group

Item status color in the Unit Test View

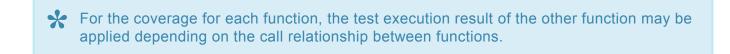
The execution information of functions, tests, and test cases in the Unit Test View is represented by colors.

Color	Description
Green	Function/Test: If the execution result of all the test cases under it is success. Test case: The result of execution is success.
Blue	Function/Test: If only test cases with failure and no test cases with errors exist under it. Test case: The result of execution is failure.
Red	Function/Test: If test cases with error exist under it. Test case: The result of execution is error.

Orange	Function/Test: All test cases have not been executed under it and test cases that are not guaranteed to run exist. Test case: Run not guaranteed.
Black	Function/Test: All test cases under it have not been executed. Test case: Not been executed.

Function

The functions provide the function name, the coverage and the test case execution results (success, failure, error and total).



Double-click on a function to check the location of the function in the Source Code Editor.

Unit Test Integration Test		
Run 🕨		 Statement Coverage
itun P	(655 / 0 / 283) 93	43.0%
		(1719/3995
pe file, function, test, state, issue		
Name	Result	Coverage
adler32_combine(unsigned long, unsigned long)	(11 / 0 / 0) 11	100.0% (18/18)
✓ ✓ test adler32_combine(unsigned long, unsigned long,	si (11 / 0 / 0) 11	
case 1		
case 2		
case 3		
case 4		
case 5		
case 6		
case 7		
case 8		
case 9 case 10		
case 11		
v ✓ bi_flush (struct internal_state *)	(5 / 0 / 6) 11	100.00/
<pre>v v test bi_flush(struct internal_state *)_0</pre>	(5 / 0 / 6) 11	100.0% (9/9)
case 1 Signaled		
case 3 Signaled		
case 4 Signaled		
case 5		
case 6 Signaled		
case 7 Signaled		
case 8 Signaled		
case 9		
case 10		
case 11		
<pre> bi_reverse(unsigned int, signed int) </pre>	(3 / 0 / 2) 5	100.0% (5/5)
<pre>Juindup(struct internal_state *)</pre>	(3 / 0 / 5) 8	100.0% (7/7)
<pre>build_bl_tree(struct internal_state *)</pre>	(0 / 0 / 8) 8	100.0% (11/11)
<pre>build_tree(struct internal_state *, struct tree_desc_s *</pre>	(0 / 0 / 7) 7	100.0% (42/42)
<pre>check_header(struct gz_stream *)</pre>	(21 / 0 / 2) 23	44.4% (20/45)
compress(unsigned char *, unsigned long *)	(3 / 0 / 0) 3	100.0% (1/1)
> compress2(unsigned char *, unsigned long *)	(6 / 0 / 0) 6	95.2% (20/21)
<pre>compressBound(unsigned long)</pre>	(3 / 0 / 0) 3	100.0% (1/1)
compress_block(struct internal_state *, struct ct_data_	s : (1 / 0 / 7) 8	77.5% (62/80)
conv block(struct internal state * char *)	(0 / 0 / 5) 5	100 00/ 10 100

Context menu of the function

	Create Test Copy Test Delete	Ctrl+C Delete
Î	Show Call Graph Show Control Flow Graph Show MC/DC	Delete
	Host Output Value -> Expected Value	
	Test reconfiguration	
	Set related file Relevant issues settings	
24 23	Import Test Data Export Test Data	>
	Add Stub	
\diamond	Apply Class Code	
	Select Function	
	Deselect Function	
250	Generate coverage report	

Context menu	Description	
Create Test	Creates a test of the selected function.	
Copy Test	Copies tests of the selected function.	
Delete	Deletes tests and test cases of the selected function.	
Show call graph	Shows the function call graph of the selected function.	
Show Control Flow Graph	Shows the control flow graph of the selected function.	
Show MC/DC	Shows MC/DC of the selected function.	
Host Output Value -> Expected Value	Pastes the host output value to the expected value.	
Test reconfiguration	Reconfigure test information.	
Set related file	Selects the files containing that test.	
Relevant issues settings	Associates the selected test with the issue of the management tool.	
Import Test Data	Imports the test data saved in local.	
Export Test Data	Exports the test data to local.	
Add Stub	Adds the stub to the selected test.	
Apply Class Code	Apply the class code generated in the Class Factory View to the	

	function.
Select Function	Selects the checkbox of all the functions selected by a mouse.
Deselect Function	Deselects the checkbox of all functions selected by a mouse.
Generate coverage report	Exports the selected test coverage to the selected path.

For [Test reconfiguration], refer to [Source Code Modification and Test Reconfiguration]

Test

The test provides coverage, test case execution results(success, failure, error, total). Double-click on a test to open its <u>Test Editor</u>.

Context menu of the test

	Add Test Case	
	Generate test cases automatically	>
	Host Output Value -> Expected Value	
	Initialize Test Case	
	Сору	Ctrl+C
	Paste	Ctrl+V
	Duplicate	Ctrl+D
	Duplicate multiple times	
×	Delete	Delete
	Rename	F2
	Test reconfiguration	
	Set related file	
	Relevant issues settings	
2	Import Test Data	>
4	Export Test Data	
	Large test data	>
	Add Stub	
\diamond	Apply Class Code	
	Select Test	
	Deselect Test	
250	Generate coverage report	

Context menu	Description
Add Test Case	Adds the test case to the unit test selected.
Generate test cases	Creates the test case in various ways.

automatically	
Host Output Value -> Expected Value	Pastes the host output value to the expected value.
Initialize Test case	Delete all the test cases.
Сору	Copies the test and test case.
Paste	Pastes the test and test case.
Duplicate	Duplicates the test and test case.
Duplicate multiple times	Duplicates tests and test cases as many as the number entered.
Delete	Deletes the test and test case.
Rename	Modifies the test name.
Test reconfiguration	Reconfigure test information.
Set related file	Selects the files containing that test.
Relevant issues settings	Associates the selected test with the issue of the management tool.
Import Test Data	Imports the test data saved in local.
Export Test Data	Exports the test data in local.
Large test data	Exports it locally or registers the file written by the user to the target test as a test case.
Add Stub	Adds the stub to the selected test.
Apply Class Code	Apply the class code generated in the Class Factory View to the test.
Select Test	Checks all checkbox of the tests selected.
Deselect Test	Checks all checkbox of the tests selected.
Generate coverage report	Exports the selected test coverage to the selected path.

Rule for importing test data
 When importing the test data, if you select multiple test data numbered with the test name, it imports those files by merging them.

For [Test reconfiguration], refer to [Source Code Modification and Test Reconfiguration]

Import test data

You can import the test data in various formats (csv, xlsx, txt, json).

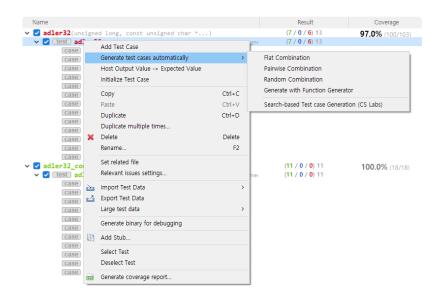
	Add Test Case Generate test cases automatically Host Output Value -> Expected Value Initialize Test Case	>		
	Copy Paste Duplicate Duplicate multiple times	Ctrl+C Ctrl+V Ctrl+D		
×	Delete Rename	Delete F2		
	Set related file Relevant issues settings			
2	Import Test Data	>	2	Basic format data
4	Export Test Data		è	External format data
	Large test data	>	Г	
	Generate binary for debugging		L .	
	Add Stub			
	Select Test Deselect Test			
CSU	Generate coverage report			

- 1. In the context menu of the Unit Test View, click [Import Test Data] and select either [Basic format data] or [External format data].
- 2. If you select [Basic format data], the test data with the format (csv) exported from the Controller Tester is imported.
- 3. If you select [External format data], the test data is imported from the files in various formats (csv, xlsx, txt, json).

Import Test Data		_		×
Import Test Data				
Select the format of the data fi	e, and select the data	file to import.		
File format:				
CT csv file 가져오기 CT excel file 가져오기 CT json file 가져오기 CT multiple test excel file 가져 multiple_test_data_importer_ir multiple_test_data_importer_L(multiple_test_data_importer_v text file 가져오기 <	Description: Controlle File extension: json		ormat. n script lo	cation
Data file				
			Browse	
		ОК	Canc	el

Generate test cases automatically

It creates a test case by using [Flat/Pairwise/Random combination], [Generates with Function Generator], and [Search-based Test case Generation (CS Labs)].



 In the context menu of the Unit or the Integration Test View, click [Generate test cases automatically] and select one among [Flat Combination], [Pairwise Combination], [Random Combination], [Generate with Function Generator] and [Search-based Test case Generation (CS Labs)].

Method	Description
Flat	Combines simply based on variables having the largest number of test data.
Pairwise	Combines so that each selected parameter data is paired at least once with the

	parameter data other than itself. ※ The number of variable partitions must be more than 2 and less than 52.
Random	Combines the test data as many as the number of test cases that the user defines any value between the minimum value and the maximum value for the variable partition of input parameters.
Generate with Function Generator	Creates a test case by using six function types. (Ramps, Random, Range, Sine, Toggle, and Single Value)
Search-based Test case Generation (CS Labs)	Creates a test case by using <u>AVM</u> , which is a local search algorithm. ※ Currently, only C language is supported.

2. If you select [Random combination], the Random combination window is displayed so that the user can enter the number of test cases.

Random Combination			
Number of Test case 5			
	OK	Cancel	

3. If you select [Generate with Function generator], you can select the function type (Ramps, Sine, Random, Toggle, Range, Single value and None) and change the settings of the function selected via the settings of the function generator. In the properties window of the project, you can set the function generator information in [Test] > [Generate test cases automatically] > [Function Generator].

Generate with	h Function Gen	erator	– 🗆 X
Test Parameter	Туре	Function t	Function type: Sine
adler	unsigned I	Sine	
buf_mem[0]	unsigned	None	Amplitude: 0.0
len	unsigned i	None	
			Period: 1 [samples]
			Phase: 0.0 [deg]
			Offset: 0.0
			Sample interval: 0
			Sample count: 0
			Start value: 0.0
			Type Min: 0.0
			Type Max: 4294967295
			Function Generator Setting
			OK Cancel

a. Common settings

You can change the value in [Project] > [Properties] with the setting value common to each function.

Setting	Description
Sample interval	The interval of samples to be sampled from the function.
Number of sample	The number of samples to be sampled from the function(The number of test cases).
Start value	The default value at which the function is started (the values are created based on the start value).
Minimum value of type	The minimum value of variable partition. (If the return value of the function is less than the minimum value of type, the minimum value of type is returned.)
Maximum value of type	The maximum value of variable partition. (If the return value of the function is greater than the maximum value of type, the maximum value of type is returned.)

b. Ramps function

It is a function that creates a pulse by using Pre, Post and Hold values. If the number of samples is greater than the period of the function, the function is called recursively.

Setting	Description
Pre delay	The time to last Pre/Post sample value.
Rise samples	The time to rise from Pre/Post sample value to Hold sample value.
Hold samples	The time to last Hold sample value.
Fall samples	The time to fall from Hold sample value to Pre/Post sample value.
Post delay	The time to last Pre/Post sample value.
Pre/Post delay value	Pre/Post delay sample value.
Hold value	Hold sample value.

c. Random function

It is a function that creates a random value between the minimum value (Min) and the maximum value (Max).

Setting	Description
Min	The minimum value of random range.
Max	The maximum value of random range.

d. Range function

It is a function that creates values that increments or decrements by a certain interval (Step Size) between the minimum value of type and the maximum value of type.

Setting	Description
Step size	The size of increment or decrement value.
Hold	The number of times that the Step size holds.
Rising	The function type that rises by the Step size. Ex) if the step size is 30, 0,30,60
Falling	The function type that falls by the Step size. Ex) if the step size is 30, 100,70,40
Alternate	If the result value of function meets the minimum/maximum value of type, the function type that changes to rise or fall. Ex) if the step size is 30 and the maximum/minimum value are 0~100, 0,30,60,90,60,30,0

e. Sine function

It is a function that creates Sine value. If the number of samples is greater than the period of the function, the function is called recursively.

Setting	Description
Amplitude	The amplitude of Sine function.
Period	The period of Sine function.
Phase	The phase of Sine function.
Offset	The offset of Sine function.

f. Toggle function

It is a function that creates FirstValue and SecondValue repeatedly.

Setting	Description
First value	The first value that is repeated in Toggle function.
Second value	The second value that is repeated in Toggle function.

g. SingleValue function

It is a function that returns only a constant single value.

Setting	Description
Value	The value to be created.

h. None

It does not create a function.

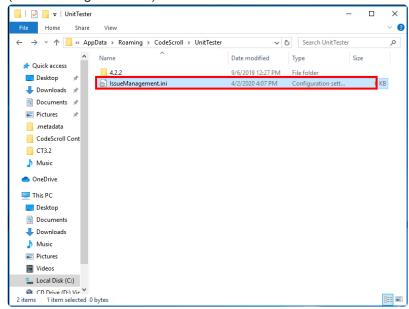
Relevant issues settings

It associates the selected test with the issue in the issue management tool registered. The Controller Tester supports the following issue management tools: JIRA, Trac, Redmine, Mantis, Bugzilla.

1. In the context menu of Unit Test View, click [Relevant issues settings...].

	Add Test Case	
	Generate test cases automatically	>
	Host Output Value -> Expected Value	
	Initialize Test Case	
	Сору	Ctrl+C
	Paste	Ctrl+V
	Duplicate	Ctrl+D
	Duplicate multiple times	
×	Delete	Delete
	Rename	F2
	Set related file	
	Relevant issues settings	
2	Import Test Data	>
4	Export Test Data	
	Large test data	>
	Generate binary for debugging	
	Add Stub	
	Select Test	
	Deselect Test	

2. Enter the information of the issue management tool in the configuration file (IssueManagement.ini).



3. Add the issues to be associated with the test.

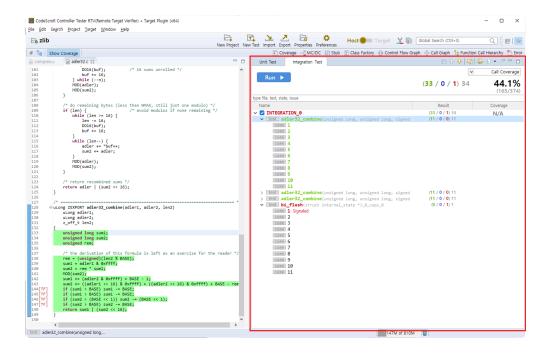
e Set associate issue			\times
Select the issues to associate with the 1 selecte	d tests		
type filter text]	
ZLIB-5		Add.	<
Enter the ID of issue to add			
ZLIB-6] [
OK	(Cancel	
Issue management tool information			
Type: JIRA <u>Open</u>	Confi	guration Fo	older
Address: http://211.116.223.80:8080	Updat	te informat	ion
ОК		Cance	1

4. Check the checked issue list and click [OK].

Set associate issue		×
Select the issues to associate with the 1 selected tests	i.	
type filter text]	
ZLIB-5	Add.	
	Edit.	
	Remo	ve
Issue management tool information		
Type: JIRA Open Confi	guration Fo	lder
Address: http://211.116.223.80:8080 Upda	te informat	ion
ОК	Cance	I

16.4. Integration Test View

The Integration Test View provides the integration test unit that can group the unit tests. The unit tests under the integration test run in order with the context maintained.



Toolbar menu in the Integration Test View

Toolbar icon	Description
Expand All	Shows all test view tree nodes.
E Collapse All	Hides all test view tree nodes.
✤ Move Up Test	Moves the selected test up.
♣ Move Down Test	Moves the selected test down.
Create	Creates an integration test.
Shows as Unique Test Name	Displays with unique test name.
🗄 Total	Displays all the tests.
[™] Failure/Error	Displays only the tests that the test result is Failure/Error.
Failure	Displays only the tests that the test result is Failure.
🗃 Error	Displays only the tests that the test result is Error.
E Success	Displays only the tests that the test result is Success.
E ₂ Function changed	Displays only the tests that the function to be tested has been changed.
E Run not guaranteed	Displays only the tests that the test execution is not ensured.

•	Host/Target result different
---	------------------------------

Displays only the test that the host result and the target result are different.

Pull-down menu in the Integration Test View

🗉 🖻 介 사 🛍 🔛 🗄 📲 🗖 🗖

Show coverage at all elements

- Add description when duplicate test case
- Insert Default Values for Empty Input Delete All Test case Descriptions
- (1) Check Integrity

Menu name	Description	
Show coverage at all elements	Displays the coverage in all items.(integration test, test, test case)	
Add description when duplicate test case	Adds original test case number on duplicated test case.	
Insert Default Values for Empty Input	When input data are empty, inserts default values.	
Delete All Test case Descriptions	Delete test case descriptions all together.	
Check Integrity	Checks integrity about functions and stubs. Then, reset test if it needs.	

• Insert Default Values for Empty Input

Variable type	Default value	
string	(empty string)	
without string	0	

For [Test reconfiguration], refer to [Source Code Modification and Test Reconfiguration]

Dashboard in the Integration Test View

Unit Test Integration Test	
	✓ Call Coverage
Run 🕨	(33 / 0 / 1) 34 44.1% (165/374)
ype file, test, state, issue	

Menu	Description	
------	-------------	--

Run 🕨	Runs integration tests.
(878 / 0 / 359) 1237	Shows success, failure, error and total of test case.
Statement Coverage	Selects the coverage type to be displayed in the Integration Test View. (statement, branch, MC/DC, function call)
43.7% (1952/4466)	Shows the selected kind of full coverage.

Search

In the Integration Test View, you can search the integration tests, tests and test cases by file, function, test, status (success, failure, error) and issue name.

Unit Test Integration Test	🖻 🕆 🖓 🛍 🗎	👺 E 🔹 🗢 🗖
	~	Statement Coverage
Run 🕨	(33 / 0 / 0) 33	43.0%
type file, test, state, issue		

Status search keyword

In the Integration Test View, you can filter the tests displayed by the status search keyword. It provides searchable keyword recommendation and auto-complete function.

Keyword	Description
%TEST_SUCCESS%	Test success
%TEST_FAILURE% %TEST_ERROR%	Test failure/error
%TEST_FAILURE%	Test failure
%TEST_ERROR%	Test error
%TEST_HAS_NOT_FUNCTION%	Function changed
%TEST_NOT_GUARANTEE%	Run not guaranted
%TEST_RESULT_DIFFERENT%	Host/Target result different

Structure of the Integration Test View

The Integration Test View presents the hierarchy structure of [Integartion tes] > [Test] > [Test case].

Suresofttech

Unit Test Integration Test	🗆 🗘 🕹 📫 🛙	
	×	Call Coverage
Run 🕨	(33 / 0 / 1) 34	44.1% (165/374)
type file, test, state, issue		
Name	Result	Coverage
<pre> v V INTEGRATION_0 v Test adler32_combine(unsigned long, unsigned long, signed long, signed long, case 1</pre>	(33 / 0 / 1) 34 (11 / 0 / 0) 11	N/A
case 3 case 4 case 5 case 6		
case 7 case 8 case 9 case 10		
<pre>case 11 test adler32_combine(unsigned long, unsigned long, signed</pre>	(11 / 0 / 0) 11	

Item icons in the Integration Test View

Item icon	Description	
None	Integration Test	
test	Test	
test	Global variable test	
case	Test case, Test case group	

Item status color in the Integration Test View

The execution information of functions, tests, and test cases in the Integration Test View is represented by colors.

Color	Description
Green	Integration Test/Test: If the execution result of all the test cases under it is success. Test case: If the result of execution is success.
Blue	Function/Test: If only test cases with failure and no test cases with errors exist under it. Test case: If the result of execution is failure.
Red	Function/Test: If test cases with error exist under it. Test case: If the result of execution is error.
Orange	Function/Test: All test cases have not been executed under it and test cases that are not guaranteed to run exist. Test case: Run not guaranteed
Black	Function/Test: All test cases under it have not been executed. Test case: Not been executed

Context menu in the Integration Test View

	Create Test		
	Create sub-integration test		
	Сору	Ctrl+C	
	Paste	Ctrl+V	
	Duplicate	Ctrl+D	
	Duplicate multiple times		
	Paste as sub-integration test		
×	Delete	Delete	
	Rename	F2	
	Select Test		
	Deselect Test		

Context menu	Description	
Create Test	Creates the dialog for creating a test	
Create sub-integration test	Creates a sub-integration test	
Сору	Copies the test and test case	
Paste	Pastes the test and test case	
Duplicate	Duplicates the test and test case	
Duplicate multiple times	Duplicates tests and test cases as many as the number entered	
Paste as sub-integration test	Pastes the integration test copied into the integration test selected as a sub- integration test	
Delete	Deletes the test and test case	
Rename	Modifies the test name	
Select tests	Checks all checkbox of the tests selected	
Deselect tests	Checks all checkbox of the tests selected	

Create Integration test and Rename

The integration test can be created in the toolbar menu. The name of the integration test is assigned automatically and can be changed by using [Rename] context menu.

Create

Unit Test	Integration Test					E 🗘 -	4 📬	📴 🗄 🔻 🏸 [
Dun N							*	Statement Coverag
Run 🕨						(33 / 0 / 0)	33	43.0%
type file, test, state,	issue							
Name						Result		Coverage
 INTEGRATI [es] adl. case 1 case 2 case 3 case 4 case 4 case 5 case 6 case 7 case 9 case 1 case 1 	er32_combine(u L 2 3 4 5 5 7 8 9 0	insigned la	ng, unsigned :	long, signed		(33 / 0 / 0) 33 (11 / 0 / 0) 11		100.0% (18/18)
Rename								
Rename Unit Test	Integration Test					日介。	0 😭	₽ • • • •
Unit Test	Integration Test					口 介。	0 😭 V	Statement Coverag
	Integration Test					□ û · (33 / 0 / 0)	~	
Unit Test							~	Statement Coverag
Unit Test Run type file, test, state, Name						(33 / 0 / 0) Result	~	Statement Coverage
Unit Test Run Unit Test Upe file, test, state, Name INTEGRAT	issue Create Test					(33 / 0 / 0) Result (33 / 0 / 0) 33	~	Statement Coverag 43.0% (1719/3995
Unit Test Run type file, test, state, Name	issue Create Test	Rename				(33 / 0 / 0) Result	~	Statement Coverage
Unit Test Run type file, test, state, Name V INTEGRAT Case Case	Create Test Create sub-i		INTEGRATION_0			(33 / 0 / 0) Result (33 / 0 / 0) 33	~	Statement Coverage
Unit Test Run type file, test, state, Name VINTEGRAT Case Case Case	Create Test Create sub-i		INTEGRATION_0)		(33 / 0 / 0) Result (33 / 0 / 0) 33	~	Statement Coverage
Unit Test Run type file, test, state, Name V INTEGRAT Case Case	Create Test Create sub-i Copy		INTEGRATION_0			(33 / 0 / 0) Result (33 / 0 / 0) 33	~	Statement Coverage
Unit Test Run type file, test, state, Name INTEGRAT test adl case case case case	issue Create Test Create sub-i Copy		INTEGRATION_0	OK	-	(33 / 0 / 0) Result (33 / 0 / 0) 33	~	Statement Coverage
Unit Test Run type file, test, state, Name V V INTEGRAT Case Case Case Case Case Case Case Cas	issue Create Test Create sub-i Copy Paste Duplicate		INTEGRATION_0	OK		(33 / 0 / 0) Result (33 / 0 / 0) 33 X 1	~	Statement Coverage
Unit Test Run type file, test, state, Name V INTEGRAT Case Case Case Case Case Case Case Cas	issue Create Test Create sub-i Copy Paste Duplicate Duplicate m		INTEGRATION_0 Delete	OK		(33 / 0 / 0) Result (33 / 0 / 0) 33 X 1	~	Statement Coverage
Unit Test Run type file, test, state, Name V V INTEGRAT Case Case Case Case Case Case Case Cas	issue Create Test Create sub-i Copy Paste Duplicate Duplicate Duplicate m Paste as sub			OK		(33 / 0 / 0) Result (33 / 0 / 0) 33 X 1	~	Statement Coverage
Unit Test	issue Create Test Create sub-i Copy Paste Duplicate Duplicate Duplicate m Paste as sub X Delete		Delete	OK		(33 / 0 / 0) Result (33 / 0 / 0) 33 X 1	~	Statement Coverage

Sub-integration test

You can create a sub-integration test under the integration test by using [Create sub-integration test] context menu or pasting an existing integration test into other integration test using [Paste as sub-integration test] context menu.

Unit Test Integration Test	🖻 🗘 🕂 🖬	
	~	Statement Coverage
Run 🕨	(33 / 0 / 0) 33	43.0% (1719/3995)
type file, test, state, issue		
Name	Result	Coverage
 INTEGRATION_0 test adler32_combine(unsigned long, unsigned long, signed 	(33 / 0 / 0) 33 (11 / 0 / 0) 11	100.0% (18/18)
✓ INTEGRATION_1	(22 / 0 / 0) 22	
 test adler32_combine(unsigned long, unsigned long, signed long, signed long) Case 1 Case 2 Case 3 Case 4 Case 5 	(11 / 0 / 0) 11	

Add a test and change the order of run

There are four ways to add tests to the integration test.

- 1. You can select an integration test and create it by using [Create Test...] context menu.
- 2. You can add by dragging and dropping the test in the Unit Test View to the integration test.
- 3. You can select the test of the Integration Test View and copy/paste it by using the context menu.
- 4. You can select the test in the Unit Test View and copy/paste it by using the context menu.

There are two ways to change the execution sequence for the test.

- 1. You can select a test or test case and change it by moving with a mouse.
- 2. You can select a test or test case and change it by selecting either [Move Up Test] or [Move Down Test] in the toolbar.

	¥	Call Coverage
Run 🕨	(33 / 0 / 1) 34	44.1% (165/374
pe file, test, state, issue		
Name	Result	Coverage
 V INTEGRATION_0 test adler32_combine(unsigned long, unsigned long, signed long, signed case 1 	(33 / 0 / 1) 34 (11 / 0 / 0) 11	N/A
case 2		
case 3 case 4 case 5		
case 6 case 7		
Case) 8 (Case) 9		
case 10 case 11		

The coverage result of integration test

If you select the integration test, the result can be checked in the Coverage.

Unit Test Integration Test	🗉 🗘 🕂 🖬	😓 E 🔹 🍸 🗖
	~	Statement Coverage
Run 🕨	(33 / 0 / 0) 33	43.0% (1719/3995)
type file, test, state, issue		
Name	Result	Coverage
<pre> V VITEGRATION_0 V Vest adler32_combine(unsigned long, unsigned long, signed long) Case 1 Case 2 Case 3 Case 4 Case 5 Case 6 Case 7 Case 8 Case 9 Case 10 </pre>	(33 / 0 / 0) 33 (11 / 0 / 0) 11	100.0% (18/18)

16.5. Coverage View

It shows the coverage of the function to be tested after test has been run as a percentage.

st cov	erage information of 'zlib' project					
	Target Function	Statement	Branch	MC/DC	Function Call	Functi
		•	•	•	-	•
1	_tr_align(struct internal_state *)	•• 72.72%	— 70.00%	40.00%	— 100.00	Υ
2	_tr_flush_block(struct internal_state	= 5 4.76%	62.50%	= 36.36%	— 70.00%	Υ
3	_tr_init(struct internal_state *)	— 100.00%	□ N/A	□ N/A	— 100.00	Υ Y
4	_tr_stored_block(struct internal_stat	— 100.00%	— 100.00%	— 100.00%	— 100.00	Υ
5	_tr_tally(struct internal_state *, unsi	= 22.22%	— 0.00% (— 0.00% (□ N/A	Υ Y
6	adler32(unsigned long, const unsig	98.05%	= 87.50%	••• 75.00%	□ N/A	Y
7	adler32_combine(unsigned long, u	— 100.00%	— 100.00%	— 100.00%	□ N/A	Υ
8	bi_flush(struct internal_state *)	— 100.00%	— 100.00%	— 100.00%	□ N/A	Y
9	bi_reverse(unsigned int, signed int)	— 100.00%	— 100.00%	— 100.00%	□ N/A	Υ
10	bi_windup(struct internal_state *)	— 100.00%	— 100.00%	— 100.00%	□ N/A	Υ Y
11	build_bl_tree(struct internal_state *)	— 100.00%	•• 75.00%	= 50.00%	— 100.00	Y
Fotal		43.02% (28.51% (19.07% (44.11% (100.0

Coverage type provided in Controller Tester

Coverage	Description
Statement	Percentage of source code statement executed by the test.
Branch	Percentage of branch executed by the test. 100% branch coverage includes 100% decision coverage and 100% statement coverage.
MC/DC	Coverage considering the condition coverage and the decision coverage complexly(described in MC/DC view).
Function call	Percentage of the function called by the test among all the functions.
Function	Percentage of functions that have been called at least once in all functions.

Label icon in the Coverage View

lcon	Description
۵	Function changed.
A	Function containing Asm code.

Toolbar menu in the Coverage View

Toolbar icon	Description
E:	Shows coverage for each test.
	Shows full coverage.
(tt)	Shows full coverage(including the external coverage).

Show Coverage

You can check the coverage of the tests in the Coverage View or in the Source Code Editor.

Show coverage in the Coverage View

The Coverage View shows the coverage information of the test or test case selected in the Unit Test View and Integration Test View. If you click [Show full coverage], the coverage information for all tests can be checked by merging them. And if you click [Show full coverage (Include External Coverage)], it shows also the coverage imported externally by merging them.

 The coverage for the items selected in the tree is displayed at the bottom of the Coverage View.

 Total
 43.02% (171...
 28.51% (603/2115)
 19.07% (239/1253)
 44.11% (165/...
 100.00% (114/...)

In the Coverage View, the coverage for each function is displayed for the items selected in the Unit or Integration Test View.

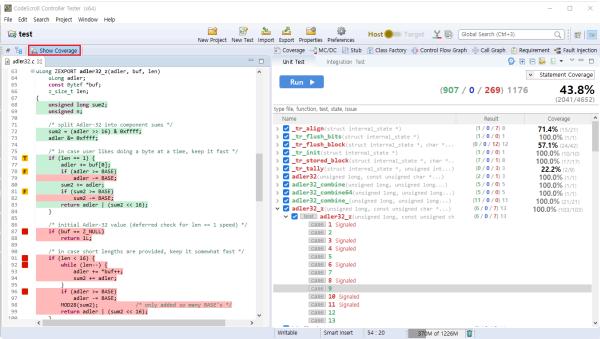
	Target Function	Statement	Branch	MC/DC	Function Call	Functi
	Target Function	- Statement	- Diancii		runcuon can	runcu ▼
1	_tr_align(struct internal_state *)	— 72.72%	— 70.00%	40.00%	1 00.00	-γ
2	_tr_flush_block(struct internal_state	= 5 4.76%	62.50%	= 36.36%	— 70.00%	- Y
3	_tr_init(struct internal_state *)	— 100.00%	□ N/A	□ N/A	= 100.00	Υ
4	_tr_stored_block(struct internal_stat	— 100.00%	= 100.00%	— 100.00%	= 100.00	Υ
5	_tr_tally(struct internal_state *, unsi	= 22.22%	— 0.00% (- 0.00% (□ N/A	Υ
6	adler32(unsigned long, const unsig	98.05%	87.50%	= 75.00%	□ N/A	Υ
7	adler32_combine(unsigned long, u	— 100.00%	— 100.00%	— 100.00%	□ N/A	Υ
8	bi_flush(struct internal_state *)	— 100.00%	= 100.00%	— 100.00%	□ N/A	Υ
9	bi_reverse(unsigned int, signed int)	— 100.00%	= 100.00%	— 100.00%	□ N/A	Υ
10	bi_windup(struct internal_state *)	— 100.00%	— 100.00%	— 100.00%	□ N/A	Υ Y
11	build_bl_tree(struct internal_state *)	— 100.00%	•• 75.00%	= 50.00%	— 100.00	Υ Y
Total		43.02% (28.51% (19.07% (44.11% (100.0

In the Coverage View table, you can the control flow graph, the function call graph and MC/DC for each function via the context menu.

st cov	erage information of	'zlib' project					
	Target Fu	Inction	Statement	Branch	MC/DC	Function Call	Functi
			•	-	-	-	•
1	_tr_align(struct	al		70.00	= 40.00	- 100.00	— Y
2	_tr_flush_block(stru	Show CFG		62.50%	= 36.36%	= 70.00%	- Y
3	_tr_init(struct intern	Show Call G	Graph	⊐ N/A	□ N/A	= 100.00	Y
4	_tr_stored_block(str	Show MC/E	C	= 100.00%	— 100.00%	= 100.00	- Y
5	_tr_tally(struct interr	nal_state *, unsi	= 22.22%	- 0.00% (— 0.00% (□ N/A	- Y
6	adler32(unsigned lo	ng, const unsig	98.05%	= 87.50%	= 75.00%	□ N/A	- Y
7	adler32_combine(ur	nsigned long, u	— 100.00%	— 100.00%	— 100.00%	□ N/A	- Y
8	bi_flush(struct interr	nal_state *)	— 100.00%	— 100.00%	— 100.00%	□ N/A	- Y
9	bi_reverse(unsigned	int, signed int)	— 100.00%	— 100.00%	— 100.00%	□ N/A	- Y
10	bi_windup(struct int	ternal_state *)	— 100.00%	— 100.00%	— 100.00%	□ N/A	- Y
11	build_bl_tree(struct i	internal_state *)	— 100.00%	••• 75.00%	== 50.00%	1 00.00	Y
otal			43.02% (28.51% (19.07% (44.11% (100.0

Show coverage in the Source Code Editor

If you enable [Show Coverage] icon in the main toolbar and click the wanted fucntion/test/test case in the Unit/Integration Test View, it shows which part of the source code is covered by using the color information. The green color indicates that the code is covered, the red color indicates that the code is not covered, and yellow color indicates that the portion of the code is covered only.



In the vertical column on the left of the Source Code Editor, the marker indicate whether the branch statement is true or false.

Marker	Description
Т	The branch is only covered as true.
F	The branch is only covered as false.

The branches are all covered.
The branch is not covered.
A line with multiple branches partially covered.

16.6. MC/DC View

The [MC/DC] view shows the information of MC/DC coverage(Modified Condition/Decision Coverage).

MC/DC coverage

MC/DC is the improved condition/decision coverage allowing each individual conditional expression to independently affect the result of the whole conditional expression without being affected by the other individual conditional expressions and it is stronger than the conditional/decision coverage. If there is no change in the other conditions and it affects the result when the own state has changed, the state can be said to satisfy MC/DC, and the conditions for creating the truth table are as follows.

- The state of decision(decision statement) must satisfy at least once all possible results(true, false).
- All individual conditional expressions belonging to the decision must satisfy at least once all possible results(true, false).
- Each conditional expression belonging to the decision affects independently the result value of decision it belongs to without being affected by the other individual conditional expressions.

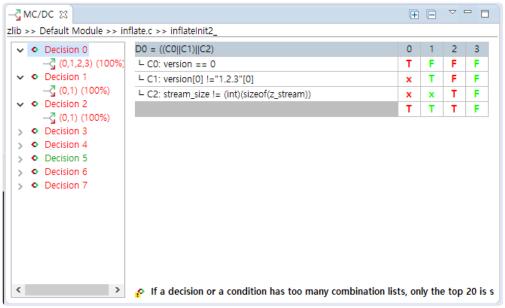
MC/DC view table

• View combination lists satisfying the target coverage The following figure is a view that can identify easily the combinations which need to be covered to achieve the target coverage.

The left section shows the list of Decisions for the selected functions, below which the list of combinations satisfying the target coverage are displayed.

The right section shows the truth table of the selected Decision or combinations and whether or not it is covered. The covered combination is displayed in green and the uncovered combination is displayed in red.

In the Preferences page, you can change the target coverage rate and the screen view mode. And you can select all for the truth table and use 'Ctrl + C' or Copy clipboard function from the context menu.



• View combination lists satisfying the coverage for each condition

The following figure is a view that can identify a one-pair combination satisfying one condition. The left section shows the list of combinations that satisfies the coverage for each condition. The right section shows the truth table for the selected combinations and whether or not it is covered.

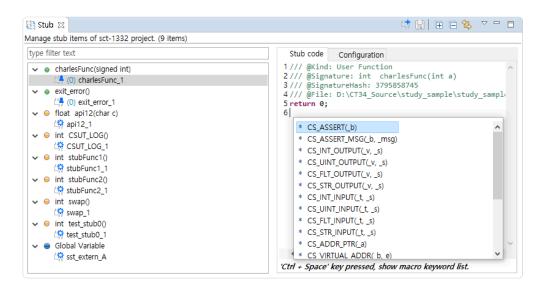
→ MC/DC 🔀		+	Ē	
zlib >> Default Module >> i	inflate.c >> inflateInit2_			
V ODEcision 0	D0 = ((C0 C1) C2)	0	3	
✓	└ C0: version == 0	Т	F	
⇒ (0, 3)	└ C1: version[0] !="1.2.3"[0]	x	F	
✓ - C1	└ C2: stream_size != (int)(sizeof(z_stream))	x	F	
		Т	F	
✓ - C2				
Decision 1				
Occision 2				
> Occision 3				
> • Decision 4				
> > Decision 5				
> Oecision 6				
> Occision 7				
	If a decision or a condition has too many combination li	sts, on	ly the	top 20 is s

16.7. Stub View

In Controller Tester, if there are no libraries used in the target function when executing a test, if the libraries are not yet developed, or if the variables not declared are used, it creates automatically the stubs instead. The stubs are created when running the test and the body is not implemented.

You can add stubs directly in addition to the stubs created in Controller Tester. Multiple stubs can be created for the same function and the stubs created can be linked to each test.

You can control the behavior of each stub by using user macro within the stub. The value used in the user-entered macro can be controlled in Edit a test case tab for the test linked.



Function type

Туре	Description
Function	Function without deficition
Function	Function with definition
Variable	Global variable

Stub type

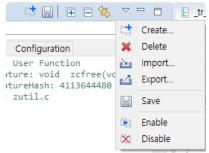
Туре	Description
📫 User stub	User-defined stub
😫 Build stub	Auto-created stub
$\overset{[2]}{\rightrightarrows}$ Old version stub	Stub with 2.3 or less version

Menu

Menu	Description
🕂 Create	Creates stubs.
X Delete	Deletes stubs.
🟜 Import	Imports the stub file exported.
🗳 Export	Exports a stub file.
🔡 Save	Saves the change stub information.
🔄 Enable	Enables stubs.
🖄 Disable	Disables stubs.

Create stubs

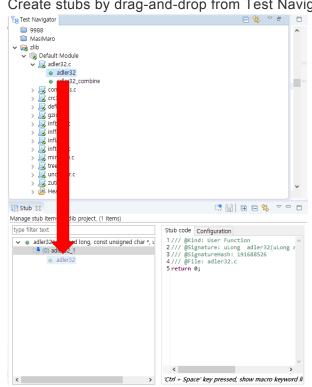
- · Create stubs in Stub view
 - 1. Click the pull-down menu in Stub view and click [Create] menu.



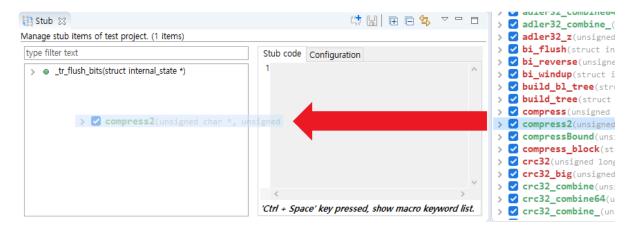
2. Check the functions to be created as stubs and select [OK].

	Function	Path	
-	 ·	P	
1	_tr_align(struct internal_state *)	trees.c	
2	_tr_flush_block(struct internal_state *, char *, unsigned lon		
3	_tr_init(struct internal_state *)	trees.c	
4	_tr_stored_block(struct internal_state *, char *, unsigned lo		_
5	_tr_tally(struct internal_state *, unsigned int, unsigned int)	trees.c	
6	 adler32(unsigned long, const unsigned char *, unsigned i 		_
7	 adler32_combine(unsigned long, unsigned long, signed l 	adler32.c	
8	bi_flush(struct internal_state *)	trees.c	_
9	 bi_reverse(unsigned int, signed int)	trees.c	
10	 bi_windup(struct internal_state *)	trees.c	_
11	build_bl_tree(struct internal_state *)	trees.c	
12	 build_tree(struct internal_state *, struct tree_desc_s *)	trees.c	_
13	 check_header(struct gz_stream *) 	gzio.c	
14	 compress(unsigned char *, unsigned long *, const unsign 		_
15	compress2(unsigned char *, unsigned long *, const unsig	compress.c	

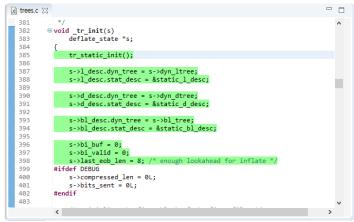
• Create stubs by drag-and-drop from Test Navigator view to Stub view.



• Create stubs by drag-and-drop from Unit Test view to Stub view.



- · Create stubs in source code editor
 - 1. Select the functions to be created as stubs.



2. Right-click it and select [Create Stub] context menu.

👌 trees.c 🛛					
381	*/				^
	■ void _tr_init				
383 384	deflate_s	state "s;			
385	tr stati	- 1 - 1 - K \ -			
386	ur_stat.	🕗 Undo	Ctrl+Z		
387	s->1 de:	1	Contra 2		
388	s->1 de:	Revert File			
389	5-71_ue:	Save	Ctrl+S		
390	s->d des				
391	s->d des	Open With	>		
392		Show In	Alt+Shift+W >		
393	s->bl de	SHOW III	Alt+Shirt+W /		
394	s->bl de	Cut	Ctrl+X		
395					
396	s->bi bu	Сору	Ctrl+C		
397	s->bi_va	Paste	Ctrl+V		
398	s->last			for inflate */	
399	#ifdef DEBU(Source	Alt+Shift+S >		
400	s->compr	Surround With	Alt+Shift+Z >		
401	s->bits	Sunound man	Alconitezy		
402	#endif	Search Text	>	1	
403					\checkmark
	<	📫 Create Stub			>
Stub 🔀		Preferences		C C C C C C C C C C C C C C C C C C C	~ - E

- Create and Link in Test Editor
 - 1. In the test structure of Test Editor, select the stubs or sub nodes.
 - 2. Select [Add New Stub...] button on the right section.

est li	nfo (zlib/_tr_init_test0)						4
est :	Structure		Q 🕀 🛛	Test	Info Edit		
est s	tructure using a tree view and edit	t the informati	on in the te	t. Man	age the stub item	s added to test.	
Nam		In	Out		Name	Description	Add New Stub
	Test global code User code			-			Add Stub
>	Global Variable			-			Remove Stub
>	Test target function						Remove Stub
	User code			_			
~	Stub Host						
	Target						
				-			
				_			

3. Check the functions to be created as stubs and select [OK].

	 Function	Path	
1	 _tr_align(struct internal_state *) 	trees.c	
2	 tr flush block(struct internal_state *, char *, unsigned lon 	trees.c	
3	 tr init(struct internal state *) 	trees.c	
4	 _tr_stored_block(struct internal_state *, char *, unsigned lo 		
5	 _tr_tally(struct internal_state *, unsigned int, unsigned int) 	trees.c	
6	 adler32(unsigned long, const unsigned char *, unsigned i 	adler32.c	
7	adler32_combine(unsigned long, unsigned long, signed l	adler32.c	
8	 bi_flush(struct internal_state *) 	trees.c	1
9	 bi_reverse(unsigned int, signed int) 	trees.c	
10	bi_windup(struct internal_state *)	trees.c	
11	build_bl_tree(struct internal_state *)	trees.c	
12	build_tree(struct internal_state *, struct tree_desc_s *)	trees.c	
13	check_header(struct gz_stream *)	gzio.c	
14	compress(unsigned char *, unsigned long *, const unsign	compress.c	
15	compress2(unsigned char *, unsigned long *, const unsig	compress.c	

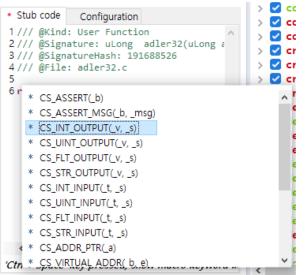
Delete stubs

• Select and right-click unnecessary stubs and select [Delete] menu.

😫 Stub 🐹	🗮 🔛 🖻 🖻 😫 🔻 🗖 🗖
Manage stub items of zlib project. (1 items)	
type filter text	Stub code Configuration
✓ ● adler32(unsigned long, const unsigned char *, u ↓ (3) adler32_1	Stub key: 1707593554463421469
iț (0) ad () Create	Related Files: P:\Example_Project\c\dip\zlib-1.2.
💢 Delete	Test:
🛃 Export	_tr_init(struct internal_state *)_0-Host
💢 Disable	adler32_combine(unsigned long, unsigned long, signed long
	adler32_combine(unsigned long, unsigned long, sign
	< >>
	Description:
	^
	~
< >	Save

Edit stubs

• You can check the macro information by pressing 'Ctrl + Space' in the Stub Code screen. After editing the stub code, press 'Ctrl + S' or [Save] button in the toolbar to save it.



• In the stub setting screen, the description can be edited.

Stub code	* Configuration	
Stub key:	1707593554463421469	
Related Files:	P:\Example_Project\c\dip	lib-1.2.:
Test:		
adler32_com	internal_state *)_0-Host bine(unsigned long, unsigned lo bine(unsigned long, unsigned lo	
<		>
Description:		
adler32() stub	þ	~
		~
		Save

Link between stubs and tests

Each stub can be linked to the host/target test.

If you select "Stub" in the test structure editor, you can edit all the stubs included in the host and the target.

~	Stu	Jb
`	/	Host
		📫 adler32_1
`	/	Target
		📫 adler32_1

If you select "Host" or "Target", you can only edit the stubs in the selected test environment.

• Dragging and dropping into the test editor in Stub view

th Stub 🖾	it 🔛 🖻 🖻 🕏 🔻 🖻	
Manage stub items of zlib project. (1 items)		Test Info (zlib/adler32_combine_test0)
type filter text	Stub code Configuration	The information has been changed after executi
✓ ● adler32(unsigned long, const unsigned char *, u └	Stub key: 1707593554463421469	Test Structure
(0) adler32_2	Related Files: P:\Example_Project\c\dip\zlib-1	2.: Test structure using a tree view and edit the in Name
	Test:	Test global code
	(struct internal_state *)_0-Host	User code
	adultation of the second secon	Global Variable
		> Test target function
	<	> User code
	Description:	Host
		▲ Clifes3211
		✓ Target
		🚔 adler32_1
		v
< >	s	ave
		Test Info Test Case Test Code Configuration

If linked by drag & drop, it is linked to all the targets and all the hosts.

- Link with [Add Stub...] button in Test Editor.
 - 1. Select stubs from the test structure in Test Editor.
 - 2. Select [Add Stub...] button on the right section.

st Info (zlib/compress_test0)					4
est Structure		Q 🕀 E	Test Info Edit		
est structure using a tree view and edit t	he informati	on in the test	Manage the stub ite	ems added to test.	
Name	In	Out	Name	Description	Add New Stub
Test global code				Description	
User code					Add Stub
Global Variable					Remove Stub
> Test target function					Remove stub
User code					
✓ Stub					
Host					
Target					

3. Among the existing stub list, check the stubs to be linked and select [OK].

Add Stub	×
Add Stub Select stubs to add the test. If the stubs of the same function have been alre	ady existed, they are overwritten with the
Input Filter Text	
Stub name	Description
adler32_1	adler32() stub
dler32_2	
adler32_3	
adler32_4	
∃ Stub code	
<pre>1/// @Kind: User Function 2/// @Signature: ulong adler32(uLo 3/// @FignatureHash: 191688526 4/// @File: adler32.c 5 return 0;</pre>	ng adler, const Bytef * buf, uInt len)
<	>
	OK Cancel

Unlink between stubs and tests

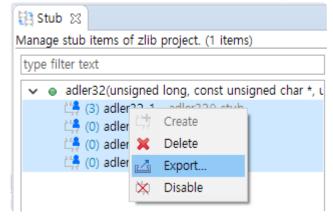
Select the stubs linked in the Test editor and select [Remove Stub] button or select the stubs in the test structure tree and right-click it to unlink them.

E *compress(unsigned char *, unsi 🛛			
Test Info (zlib/compress_test0)			2 è
Test Structure	Q (Test Info Edit	
Test structure using a tree view and edit the info	rmation in the	e test. Manage the stub items a	dded to test.
Name	In Out	Name	Description Add New Stub
Test global code		💻 🐫 adler32_4	
User code		adler32_4	Add Stub
Global Variable			Remove Stub
> Test target function			Kentove Stub
User code			
✓ Stub			
✓ Host			
i adler32_4			
✓ Target			
🛀 adler32_4			
Test Info Test Case Test Code Configuration			

		-	
×	Stub		_
~	×	Disconnect Stub	
	i 🗧 a	dler32_4	_
~	Targe		
	iļ a	dler32_4	

Export stubs

1. Select and the stubs to be exported and right-click and select [Export...] menu.

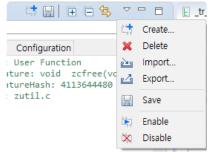


2. Specify the saving location of the stub file in the directory dialog and select [OK].

Save			
 Documents Export Ilib Stubs UnitTests export_stub source 	^		
> 🕹 Downloads > 🍌 Music			
> E Pictures			
Videos Local Disk (C:)	<u>~</u>		
Folder: New folder			
Make New Folder OK Cancel			

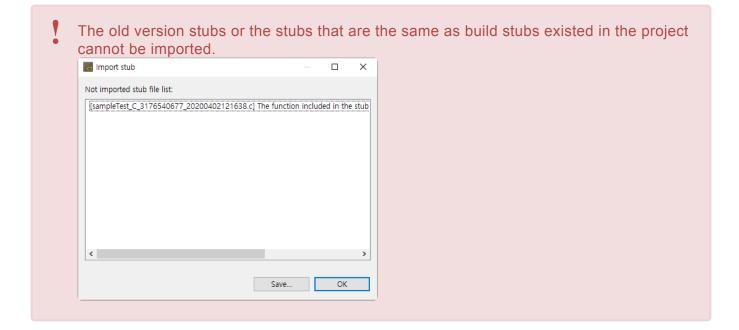
Import stubs

1. Select [Import...] menu from the pull-down menu in Stub view.



2. Select the stub file to be imported and select the [Open] button.

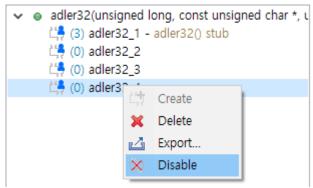
🗧 🔶 👻 🕇 📙 > Thi	s PC > Documents > export_stub			✓ Ö Search ex	port_stub	,P
Organize 🔻 🛛 New folde	r					?
📃 Desktop 🛛 🖈 ^	Name	Date modified	Туре	Size		
🕂 Downloads 🖈	funcA_1_1_382453298_20200402104949_5.c	4/2/2020 10:49 AM	C File	1 KB		
🔮 Documents 🖈	funcB_1_1_365675679_20200402104949_5.c	4/2/2020 10:49 AM	C File	1 KB		
📰 Pictures 🛛 🖈	main_1_1_3935363592_20200402104949_5.c	4/2/2020 10:49 AM	C File	1 KB		
CodeScroll Cont						
CT3.2						
b Music						
Videos						
 OneDrive 						
💻 This PC						
Desktop						
🖶 Documents 🗸						
File na	ime:			~ *.c		~
	1			Op	en Canc	ما



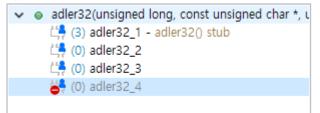
Use/Not use stubs

You can set whether to use the created stubs.

1. Select the stubs and right-click and select [Disable] menu.

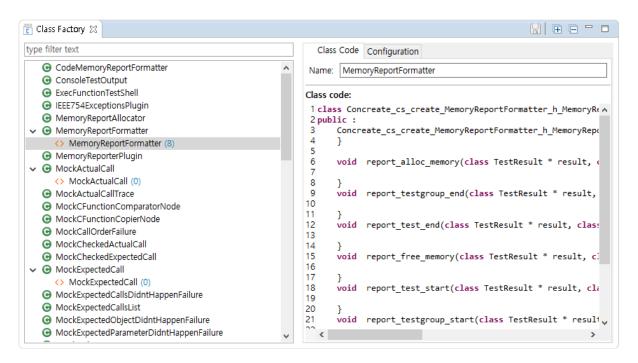


2. The stub image decorator is added and the color of the stub name changes to gray as shown in figure below.



16.8. Class Factory View

When testing C++ source code, you can use class code to create class objects. Since abstract classes cannot be instantiated, Controller Tester automatically generates instantiable class code.



- On the left side of the [Class Factory view], you can see the classes included in the project and the class code for instantiating the class in a tree structure.
- On the right side of the [Class Factory View], you can edit the class code and name in the [Class code] tab, and see the connected tests in the [Configuration] tab.

Toolbar menu of Class Factory view

Item	Description
🗟 Save	Save modification of the class code.
Expand All	Expand all class tree in Class Factory view.
Collapse All	Collapse all class tree in Class Factory view

Icon of Class Factory view

lcon	Description
G Class	Class included in the project.
Abstract class	Abstract class included in the project.
Mock object	Mock object used in test.
<>> Class code	Class code that create class objects to use in tests.

Context menu of class

Item	Description
📌 Create	Create a class code of the selected class.

Context menu of class code

Item	Description
X Delete	Delete selected class codes.

Using class factory

You can use the class factory in two ways.

- Using class codes
- Using mock objects

16.8.1. Using class codes

Create a class code

When analyze the project, a list of all class included in source codes appears in Class Factory view. Controller Tester automatically generates class codes of abstract classes. You can also manually create class code from the context menu of class.

👸 Class Factory 🕱	
type filter text	
G CodeMemon/ReportFormatter	^
🕞 Consol 💎 Create	
 ExecFunctionTestShell 	
IEEE754ExceptionsPlugin	

Apply class code

There are three ways to apply the class code to test.

Apply in Test Editor

 Open the test editor by double-clicking on the test that you want to apply the class code. In the [Test Info] tab, expand the test structure, select the object to which the class code is to be applied, and select [Use class code] in [Test Info Edit] – [Constructor] on the right.

E MemoryReportAllocator::setForm 🔀					
Test Info (class_factory/MemoryReportAlloca	ator_s	setForn	natte	r_test0) 🗳	è
Test Structure		Q 🖪		Test Info Edit	
Test structure using a tree view and edit the inform	nation	in the f	test.	Set to create a object and select member field	^
Name	In	Out	^	Constructor	
Test global code				Use class code	
User code				User code	-
Global Variable				Use class code	
 Test target function 				NormalMemoryReportFormatter::NormalMemoryReportFormatter	0
 o^T object_MemoryReportAllocator : cla 				Function for object return.	_
MemoryReportAllocator::Memory					
 MemoryReportAllocator::setForm. 					
Local Static Variable					
 Parameter/return 					
 o formatter : class Memory 					
✓ ● formatter_mem : clas					
MemoryReportForm					
Before call code					
After call code					
User code					5
Test Info Test Case Test Code Configuration		1			

2. Select class code in [Test Structure]. In [Test Info Edit], check the class code to be applied and save. Select the class code to see the code.

Test Info (class_factory/MemoryReportAllocator_setForma	itter_tes	t0) 🛃 🔛
T 1.01		
Test Structure	E E	Test Info Edit
Test structure using a tree view and edit the information in the test	st.	Select the class code to use for creating class objects.
Name In Test global code In User code In Global Variable In Test target function In Image: Start Strate Start Star	Out ^	MemoryReportFormatter add class code Add class code
Local Static Variable Parameter/return A formatter : class MemoryReporti A formatter_mem : class MemoryReportFormatter Before call code After call code User code Stub C	> *	<pre>1 class Concreate_cs_create_MemoryReportFormatter_h_! ^ 2 public : 3 Concreate_cs_create_MemoryReportFormatter_h_Men 4 } 6 void report_alloc_memory(class TestResult * rn 7 8 } <</pre>

- Select [Go to Class Factory] to open the class code in Class Factory view.
- Click [Add Class code] to add new class code.

Apply by drag-and-drop

You can apply the class code by dragging and dropping the class code to be applied from Class Factory view to the class object in Test Editor.

•	Falameter/Tetum	
~	▲ formatter : class MemoryReportFo	
<> Memo	ryReportAmatatter_ლიდიდ): class Memor	
	 OrmalMemoryReportForma 	
- 4	Betore call code	
	After call code	
User code		
est Info Test Case	Test Code Configuration	
្នំ Class Factory ខ្ល		
ype filter text		
✓ ⊙ MemoryRep	ortFormatter	^
<> Memory	ReportFormatter (7)	
<> Memory	ReportFormatter_2 (0)	
G MemoryRep	oorterPlugin	

Apply using context menu in Test view

1. In Test view, select [Apply Class code] in context menu of functions or tests.

> 🛃 MemoryRep	ant/	Vilocator::setFormatter(MemoryP Create Test	enortForm
		Copy Test	Ctrl+C
	×	Delete	Delete
		Show Call Graph Show Control Flow Graph Show MC/DC	
		Host Output Value -> Expected Value	
		Test reconfiguration	
		Set related file	
		Relevant issues settings	
	2	Import Test Data	>
1	4	Export Test Data	
		Add Stub	
	\diamond	Apply Class Code	
		Select Function	
		Deselect Function	
ſ	C5U	Generate coverage report	

 In [Apply Class code] dialog, select the class code to apply to selected tests and click [Ok]. Expand [Class code] at bottom to see the selected class code.

ype filter text										
~	Class Code(Click to select) MemoryReportFormatter									
eleo	ct All Deselect All									

With [Apply Class code] feature, the same class code is applied wherever the class code can be applied.

Modify class code

You can modify the class code on [Class code] tab of Class Factory view. The modified class code is reflected in all connected tests.

Delete class code

You can delete class code in context menu of Class Factory view.

Suresofttech

	Class Factory 🔀								
typ	e filter text								
~	Ge MemoryReportFormatter MemoryReportFormatter (7) (1) MemoryReportFormatter (7)								
>	MemoryReportFormatter 2 (0) MemoryRep Z Delete Delete MockActualCall MockActualCallTrace								
	MockCFunctionComparatorNode								

It is impossible to delete the class code while it is connected to the test. Before deleting the class code, all connections with the test must be disconnected.

16.8.2. Using mock objects

The following toolchain is available for use with mock objects:

- GCC 6.0 or later
- Visual Studio 2015 or later

Creating and Using mock objects

- 1. Open [Test Editor] by double-clicking a test you want to create a mock object for.
- 2. In the Test Info Tab, expand the test structure tree and select the object to create a mock object. Select [Use mock] at the constructor in the Test Info Edit area on the right.

🕒 *CppUnit::TestDecorator::TestDe ×					
Test Info (poco/CppUnit_TestDecorator_TestDecorator_t	test0))			<u> 2</u> 2
Test Structure		् 🖪		Test Info Edit	
Test structure using a tree view and edit the information in the test.				Set to create a object and select member field	
Name Test global code	In	Out	^	Constructor	
User code				Use mock	*
Global Variable				User code	
 Test target function 				Use class code	
 •^C TestDecorator::TestDecorator(Test *) 				Use mock Function for object return.	
Local Static Variable				Function for object return.	
Parameter/return					
 • test : class CppUnit::Test * 					
🐱 🍳 🔨 test_mem 🗄 class CppUnit::Test					
CS_MOCK_CppUnit_Test					
Before call code					
After cell code			×		

- Selecting [Use mock], the class code of the mock object is generated in the Class Factory View.
- 3. Click the created mock object in the Test Info Tab.

CppUnit::TestDecorator::TestDe $ imes$				
est Info (poco/CppUnit_TestDecorator_TestDeco	orator_tes	t0)		<u>4</u>
Test Structure	C	. 🕀 🖻	Test Info Edit	
est structure using a tree view and edit the information	n in the tes	t.	Write or generate a mock specification.	
Name Test global code	In	Out	Generate Specification Wizard	Show mocking reference
User code			1	^
Global Variable				
 Test target function 				
✓ ▲ TestDecorator::TestDecorator(Test *)				
Local Static Variable				
 Parameter/return 				
✓ 🚯 test : class CppUnit::Test *				
🗸 💧 test_mem 💠 class CppUnit::Test				
CS_MOCK_CppUnit_Test				
Before call code				
After call code				
User code				
✓ Stub			<	Y
Host			-	2
Target			Go	to other mock specification lis

 [Show mocking reference]: You can see the description about specifications of mock objects. 4. You can generate mock specification in the wizard, by clicking [Generate Specification Wizard...] button in the Test Info Edit area.

Generation mock specification	×
Generation mock specification	
① You can specify input and return value or the number of repetitions of function calls. Select the target function to specify the behavior of mock for.	
void Test::run(class CppUnit::TestResult * result)	
 int Test::countTestCases() 	
std::cxx11::string Test::toString()	
OK Cancel	

For details, refer to the <u>Using Mock Objects in C++ Tests</u> page of User Guides.

Check the result of mock objects.

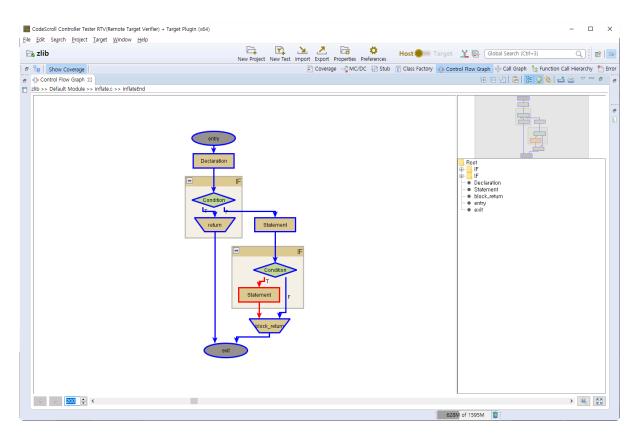
- 1. Click the Run button in the Test View.
- 2. If a mock object does not match the specifications, the test fails and displays the message [Mock Failed] next to the testcase.
- 3. You can check the failure information in [Test Editor] > [Test Case Tab].
 - Expand [► Mock failure] and check why the mock object failed.

 Mock failure 				
Name	Туре	function	Actual function call count doesn't match EXPECT_CALL	\sim
object TestDecorator test	CppUnit::Test	countTestCases()	(csmock_object_TestDecoratortest_mem, countTestCases) Expected: to be called 3 times Actual: called once - unsatisfied and active	
				\sim

16.9. Control Flow Graph View

The [Control Flow Graph] view shows the control flow information for the function selected in graph format.

If you click [Show Coverage] button in the main toolbar, the covered section is displayed in blue color and the uncovered section is displayed in red color.



Toolbar menu

Menu	Description
E Unfold All	Shows all group nodes.
E Fold All	Hide all group nodes.
Show Legend	Shows the legend(the node type in the currently displayed graph).
🔒 Copy to System Clipboard	Copies the currently displayed graph into clipboard.
Show Outline	Displays the graphs in a tree format.
Show Overview	Shows the overview of a graph.
4 Link with Editor	Shows the selected items in the graph in the editor by one-click.
🗳 Export View Content	Exports the contents of view in a report.
Print View Content	Prints the contents of view.

Pull-down menu

Menu	Description
E Unfold All	Shows all group nodes.
E Fold All	Hides all group nodes.
Show Call Graph	Shows the function call graph for the function selected in the current view.
Show Legend	Shows the legend(the node type in the currently displayed graph).
Show Outline	Displays the graphs in a tree format.
Show Overview	Shows the overview of a graph.
4 Link with Editor	Shows the selected items in the graph in the editor by one-click.
Save as Graph Format	Creates the graph model file for the graph displayed on the current screen. Four kinds of formats supported: • Graph Modeling Language XML (*.xgml) • Graph Modeling Language(*.gml) • yWorks Binary Graph Format(.ygf) • Trivial Graph Format(*.tgf)
Save as Image Format	Saves the currently displayed graph as an image format file(jpg, gif).
🛱 Copy to System Clipboard	Copies the currently displayed graph into clipboard.
Preferences	Opens the preferences.

Node Pop-up menu

Menu	Description
Show Call Graph	Shows the function call graph for the function selected in the current view.

History function

The node that had been selected in the current graph can be seen again by using the arrow button(Go back, Go forward) in the lower-left corner of the view.

▲ ▶ 177 ÷ <	• 🔍 🐹
-------------	-------

Zoom out/in function

You can change the zoom out/in ratio by entering a number at the bottom of the view or by adjusting the slider.

4	177 🔹 <	,	۹ 🔛
---	---------	---	-----

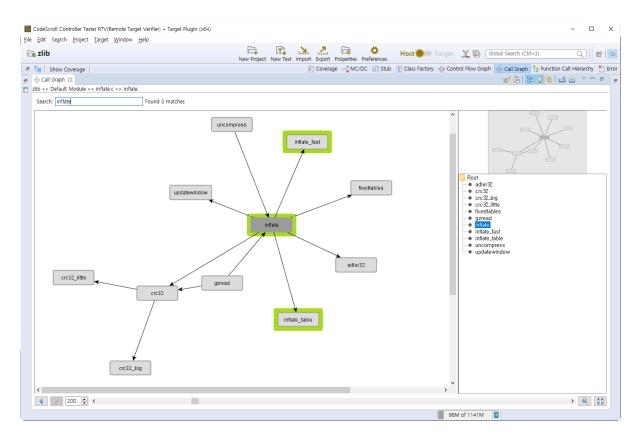
You can reset the zoom out/in ratio by using 🔍 [Initialize zoom out/in ratio] button at the lower right

corner of the view, and change the zoom out/in ratio accordingly to the view size by using 🔀 [Fit to view size] button.

16.10. Call Graph View

The [Call Graph] view shows the function call information for the selected function in a report format. (Ex: When calling function 'B' from function 'A', it is expressed 'edge from node 'A' to node 'B', and it is represented as one edge even if called multiple times.

If you select the node (function), it shows the function call information for that node.



Toolbar menu

Menu	Description
💖 Edit Layout	Changes to make the node position of graph modifiable(it cannot be automatically edited when the graph is updated).
Copy to System Clipboard	Copies the currently displayed graph into clipboard.
E Show Outline	Displays the graphs in a tree format.
Show Overview	Shows the overview of a graph.
5 Link with Editor	Shows the selected items in the graph in the editor by one-click.
Content	Exports the contents of view in a report.
Print View Content	Prints the contents of view.

Pull-down menu

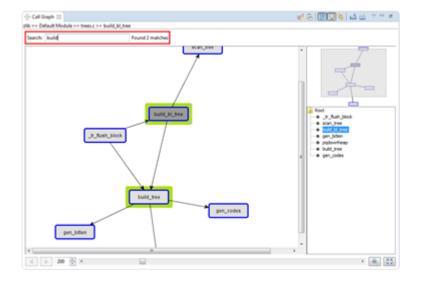
Menu	Description	
Show CFG	Shows the control flow graph for the function selected in the current view.	
Show Outline	Displays the graphs in a tree format.	
Show Overview	Shows the overview of a graph.	
4 Link with Editor	Shows the selected items in the graph in the editor by one-click.	
Save as Graph Format	Creates the graph model file for the graph displayed in the current screen. Four kinds of formats supported: • Graph Modeling Language XML (*.xgml) • Graph Modeling Language(*.gml) • yWorks Binary Graph Format(.ygf) • Trivial Graph Format(*.tgf)	
Save as Image Format	Saves the currently displayed graph as an image format file(jpg, gif).	
🔓 Copy to System Clipboard	Copies the currently displayed graph into clipboard.	
Preferences	Opens the preferences.	

Node Pop-up menu

Menu	Description
Show CFG	Shows the control flow graph for the function node selected.
Expand	Shows the call relationship after the selected function one more step.
Collapse	Hides the call relationship after the selected function.
Set to Start Node	Specifies the start function for checking the call path for two functions.
Set to End Node	Specifies the end function for checking the call path for two functions. Shows all paths possible in the current graph.

Search function

When you enter a search word in the top of the view, the functions with the name containing the entered search word are highlighted.



History function

The node that had been selected in the current graph can be seen again by using the arrow button(Go back, Go forward) in the lower-left corner of the view.

▲ 200 ÷ < □	• 🔍 🔛
-------------	-------

Zoom out/in function

You can change the zoom out/in ratio by entering a number at the bottom of the view or by adjusting the slider.

200 🔹 <	•	۹. 🔛

You can reset the zoom out/in ratio by using ⁽⁴⁾ [Initialize zoom out/in ratio] button at the lower right corner of the view, and change the zoom out/in ratio accordingly to the view size by using ⁽⁵⁾ [Fit to view size] button.

Show/Hide function

By using the menu displayed when right-clicking a node, you can show or hide the edge out from that node. If you select [Hide], it hides the call relationship for all functions called by the function corresponding to the selected node. The Hide menu of the node having no function call information is disabled. If you select [Show], it shows the functions called by the function corresponding to the selected node one more step. Likewise, if there are no call relationships, the Show menu is disabled.

Highlight Path between Nodes function

By selecting the start node and end node, it highlights all paths exited between two nodes so that you can see easily the interesting section in the complex function call relationship.

16.11. Error View

You can check the analysis error information of the project in detail.

rors in the zlib project.		
ype filter text		Dpen log in editor:
	Log locati Error se in compress 2 Test error location error details	<pre>ccl.exe: warning: command line option '-fpermissive' is valid for C++/ObjC++ but not for C In file included from P:\Product_Workspace\ct_3_1_02\zlib\.csda ta\src\build\compress_2.c:7:0: cs_test%compress_test0%User code start: In function 'compress_test0': cs_test%compress_test0%User code start:1:1: error: 'value' undeclared (first use in this function) cs_test%compress_test0%User code start:1:1: note: each undeclared identifier is reported only once for each function it appears in</pre>

Log type

Туре	Description
Build	Errors occurred during build
Statement analysis	Errors occurred during statement analysis
📾 Link	Errors occurred during linking
Pre-processing	Errors occurred during pre-processing

Error type

Туре	Description
Source	Errors occurred int the source file
Stub	Errors occurred in the stub file
Test	Errors occurred in the test editor

Go to the error location

In the context menu, select [Go to error location] to go to the location where the error has occurred so as to correct the error easily.

Show error details

Double-click the error information or select [Show error details...] in the context menu to check the detail information for the error.

	s —	I		×	
Log:	🗟 Build				
Error:	Test				
Log location:	P:\Product_Workspace\ct_3_1_02\zlib\.csdata\log	¥bui			
Message:	'value' undeclared (first use in this function)	< >	Ŷ ₽		
Resolution:					
A compile err	or has been occurred in the test code.				
The test code	is included in the file which the function under test ha	s bee	n defi	ned(t	
To check if th	e types, variables or functions used in the test code car	n be i	used o	r not	
 Please correct the compile error by checking the error message. 					
	rect the compile error by checking the error message. cult to correct it, please contact support@suresoftte e	ch.co	m by a	attacł >	
2. If it is diffi	cult to correct it, please contact support@suresofttee	ch.co	m by a		
 If it is diffi Product inform Product versi Product instation Toolchain: GG Java version: 	cult to correct it, please contact support@suresofttee		m by a		
 If it is diffi If it is diffi Product inform Product versi Product versi Toolchain: GG Java version: 	cult to correct it, please contact support@suresofttee nation: on: 3.0.2.202002210735 llation: C:#Program Files#CodeScroll Controller Tester CC 5.3 (32bit) 1.8.0_202-b08		m by a		

Buttons

Button	Description	
🕆 Previous	Shows the previous error information.	
♣ Next	Shows the next error information.	
🗈 Сору	Copies the error information.	
À Go to Error	Points to the location of the error.	

16.12. Debug Information View

The debug information view shows information to help you determine the cause of an error in a test case.

- · Trace function calls in test cases with errors
- Actual value of variable/expression added to inspect debug information

If you select a test case that has [Inspect Debug Info] in the unit test view, debug information is displayed in the Debug Information view.

👬 Debug Information 🔀			(x)= 🗖 🗍
Test Case (test/func_test0) #1			
Test Case (test/func_test0) #1 Stack trace: Image: Stack trace of the state of the st	List of variable/expression: Variable/Expression p (test1.c: 40)	Value 12319812	Location before line

- The Stack trace shows the function call trace when the test case in error was executed. The location where the error occurred is displayed at the top.
- The variable/expression list shows the actual value of the variable/expression added to inspect the debug information.
- If the actual value of the added variable/expression is not displayed:
 - When [Inspect Debug Info] of the test case is not executed after adding a new variable/ expression

- When the line where the variable/expression was added was not executed when the test case was executed (it was not executed depending on the condition or an error occurred before execution)

Debug Information View toolbar menu

icon description

(x)= List of variable/expression to debug

In the list of debug variable/expression, you can check and remove variable/expression.

This is a list of variable/expression add			
Jnnecessary variable/expression can b	be removed using [Remove]], [Remove All].	
'ariable/Expression	Туре	Location	Remove
(test1.c : func3) line:46	Integer	before line	
(test1.c : func3) line:46	Integer	after line	Remove All
(test1.c : func2) line:27	Integer	before line	
(test1.c : func3) line:40	Integer	before line	

Adding variable/expression to debug

You can add variable/expression to debug in the following ways.

- 1. Double-click the line area in the source code editor
- 2. Select [Add Variable/expression to debug...] from the line area context menu in the source code editor
- 3. After specifying a variable or expression in the source code editor, select [Add Variable/expression to debug...] from the context menu.

When the above operation is performed, the [Add variable/expression to debug] window appears.

d Add varia	ble/expression to debug				
Add variat	le/expression to debug				
	suble-click on the line area to add or remove variables/expressions to debug at that location. add variable/expression only to the shaded area. (□: before line, □: after line)				
Add All	Remove All				
84 int -	test()	A			
85 {	int a, b;				
	a = func(0);				
	b = func(1);				
89	/)				
90	if (a < b) {				
91	return g(0);				
	}				
93					
	g(1);				
	a = a + b;				
96					
	while(a<0) {				
98 99	a;				
100	}				
	return 0;				
102 }	eturn o,				
103					
105		-			
4		4			
* Entering	an invalid variable/expression can cause compilation errors or runtime errors.				
-	· · ·				
Variable/Ex	pression:				
Type: 🔘 Ir	nteger 🔘 Unsigned Integer 🔘 Double 🔘 String				
When ye	ou select a test case, you can inspect the debug info at once.				
	(OK Cancel			

You can double-click on the line area to add or remove variable/expression to debug at that location.

marker	description
•	Variable/expression added before the line
Variable/expression added after the	

You can add markers only in the shaded area.

	4 int 5 {	test()	*
	6	int a, b;	
8	7	a = func(0);	
8	8	b = func(1);	
8	9		
9	0	if (a < b) {	
9	1	return g(0);	
9 9 9	2	}	
9	3		
9	4	g(1);	
9		a = a + b;	
9			
9		while(a<0) {	_
– 9		a;	
9		}	
10			_
10		return 0;	
10	2 }		
10	3		
			Ŧ
		8	

If you can add a variable/expression to debug both before and after the line in the shaded area, a dialog will pop up asking where to add it.

Suresofttech

dd variable/expression location to debug	
Please select a variable/expression location to add.	
📝 before line	
after line	
Select All	Deselect All
ОК	Cancel

You can add or remove variables/expressions anywhere you can add to the target function through the [Add All], [Remove All] buttons at the top of the source viewer.

After specifying the variable/expression and type and clicking the [OK] button, you can see that the marker has been added to the source code editor.

.0	tes	t1.c 2	×			
Г	83					
	84	Θ	int	: test()		_
	85		{			
	86			int a, b;		
	87			a = func(0);		
L	88			b = func(1);		
	89					_
	90			if (a < b) {		
	91			return g(0);		
	92			}		
	93					
C	94			g(1);		
	95			a = a + b;		
	96					
	97			while(a<0) {		
	98			a;		-
	99			}		
	100					
	101			return 0;		
	102		}			-
	103		4			-
			4		P	

If the test case is selected in [Unit Test View], you can immediately perform [Inspect Debug Info] while adding a variable/expression to be debug.

X Entering an invalid variable/expression can cause compilation errors or runtime errors.						
Variable/Expression: val						
Type: Integer Unsigned Integer Double String						
✓ Inspect debug info: func(signed int)_0 #5						
	OK Cancel					

If you enter an invalid variable/expression, a compilation error or a runtime error may occur when performing [Inspect Debug Info].

16.13. Fault Injection View

The Fault Injection View provides the feature for inserting the code needed to test additionally into the specific area of the function to be tested.

1. Drag and drop the function that you want to inject a fault from the Test Navigator View or the Unit Test View to the Fault Injection View.

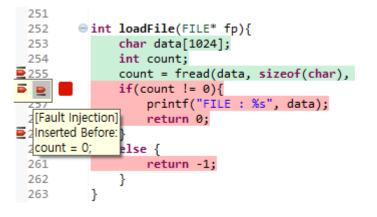
I CodeScroll Controller Tester RTV(Remote Target Verifier) + Targ	get Plugin (x64)		- 🗆 ×
Sample_code	📸 😰 🊵 🚰 📬 🏟 Host 🌒 Target 👱 🕅	Global Search (Ctrl+3)	Q) 🖻 🖃
💫 Show Coverage	🗈 Coverage 🚽 MC/DC 🔃 Stub 👸 Class Factory 👍 Control Flow	Graph 📲 Call Graph 💼 Require	ement 📲 Fault Injection
🔓 Test Navigator 📄 😫 🔍 🗖 🗖	Unit Test Integration Test	æ E	😽 E 🔹 🔍 🖻 🗖
> 😹 faultTest		×	Statement Coverage
✓ Sample_code ✓ Default Module ✓ befault injection ✓ be fault injection ✓ sudy_sample (2)	Run 🕨	(82 / 0 / 20) 102	68.3% (119/174)
 Get study_sample Study_sample 	type file, function, test, state, issue		
✓ G study_sample	Name	Result	Coverage
Subjection Subjection	<pre>>> mcdError(signed int, signed int) >>> of prest(signed int) >>> of stubFunc0(signed int) >>> of stubFunc1() >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></pre>	$\begin{array}{c} (2 \ / 0 \ / 2 \) 5 \\ (3 \ / 0 \ / 0 \) 5 \\ (3 \ / 0 \) 5 \\ (3 \ / 0 \) 5 \\ (3 \ / 0 \) 5 \) (3 \ / 0 \) 5 \\ (3 \ / 0 \) 5 \) (3 \ / 0 \) 5 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0 \) (3 \ / 0$	100.0% (22) 60.0% (4/10) 95.8% (21/24) 93.7% (15/16) 100.0% (4/4) 100.0% (4/4) 100.0% (4/4) 100.0% (4/4) 100.0% (5/5) 88.2% (15/17) 36.9% (17/46) 88.2% (5/5) 100.0% (4/1) 100.0% (4/1) 100.0% (4/1) 100.0% (4/1)
testMe3(signed int)	> 196M of 7841	M	

2. In the tree structure on the left of the view, click the line that you want to inject a fault and, on the right of the view, enter the user code to insert before/after the line. The number of the line where user code is written is underlined.

📲 Fault Injection 🔀	▣ □ 3 ☑ 4 ▽ □ □
Inserts code before or after the selected line.	
type filter text	Inserted Before:
278: }	1 count = 0;
279:	
280: return fibonacci_fail(index - 1) +	
 (0) helper(signed int) 	
<pre>266: FILE* fp = fopen("D:\\read.txt", "r</pre>	
267: return fp;	
 (1) loadFile(struct _iobuf *) 	X
253: char data[1024];	
254: int count;	Inserted After:
<pre>255: count = fread(data, sizeof(char), 1</pre>	1 /* test */
✓ <u>256</u> : if(count != 0){	
257: printf("FILE : %s", data);	
258: return 0;	
259: }	
260: else {	
☐ 261: return -1; v	v
< >>	< >

When check checkboxes of lines to insert, the user codes are applied. In the source code editor,

 (before the line),
 (after the line) markers appear at that lines. When mouse is on the markers, the tooltip about fault injection information and written code appears.



Single-line comments (//) and input/output user macros cannot be used.

Toolbar menu in the Fault Injection view

Toolbar icon	Description
Expand All	Expand all function trees in Fault Injection view.
E Collapse All	Collapse all function trees in Fault Injection view.
Show Only Non-empty Fault Injections	View only the line where the fault injection code is written among the fault injection lines.
Show Only Enable Fault Injections	View only active lines among fault injection lines.
Import Fault Injections	Import fault injection information with .xls format.
Substitution Export Fault Injections	Export fault injection information with .xls format.

Context menu in the Fault Injection view

Function nodes and lines can be multi-selected to bring up the context menu.

Context menu of function nodes

×	Remove	Delete
Mer	าน	Description
×F	Remove	Remove the function inserted by users.

Context menu of lines

	Сору	Ctrl+C
Ē	Paste	Ctrl+V
<u>a</u> _	Clean	
\checkmark	Enable	
	Disable	

Menu	Description
🝺 Сору	Copy the information inserted by user before/after the line.
╞ Paste	Paste the information inserted by user before/after the line.
🖉 Clean	Uncheck the checkboxes of the selected lines and remove inserted code.
Enable	Check the checkboxes of the selected lines and activate fault injection.
Disable	Uncheck the checkboxes of the selected lines and deactivate fault injection.

Pull-down menu

Menu	Description	
C Reconfiguring Fault Injection	Note the Reconfiguring Fault Injection	

When modifying the source code, contents of the fault injection may not be properly reflected. In case of modification, contents of the fault injection must be rewritten.

Export Fault Injections

1. Select the [Export Fault Injections] icon from the toolbar menu.

- Fault Injection $ imes$	🕀 🕞 🙆 🔽 📩 😫 🗖	
inserts code before or after the selected line.		
type filter text	Inserted Before:	
> • (1) api2(enum enm1, Enm2)		^
	< >	~
	Inserted After:	
	1	<
	< >	

2. In the directory dialog, specify the save location for the fault injection file (.xls) and then select the [Save] button.

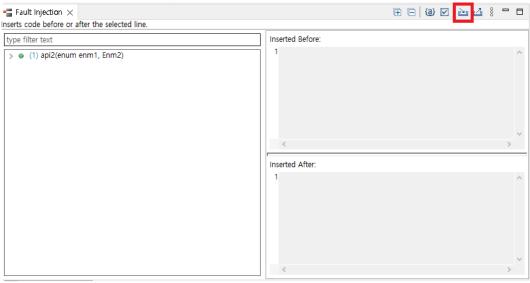
Export Fault Inject	tions				×
\leftrightarrow \rightarrow \checkmark \uparrow	→ This PC → Desktop →	export	ٽ ~		ort
Organize 🔻 Ne	ew folder				::: • ?
Name	^	Date modified	Туре	Size	
		No items match you	search.		
File <u>n</u> ame:	FaultInjection_poco.xls				~
Save as <u>t</u> ype:	*.xls				~
∧ Hide Folders				<u>S</u> ave	Cancel

3. If the export is successful, you can go directly to that path location.

Expo	rt Fault Injections		×
1	The fault injection export is complete.		
		Open Folder Con	nfirm

Import Fault Injections

1. Select the [Import Fault Injections] icon from the toolbar menu.



2. Select the fault injection file to import.

Import Fault Injections					>
$\leftarrow \rightarrow \checkmark \uparrow$ \checkmark This P	C > Desktop > export	ٽ ~	,	h export	
Organize 🔻 New folder					
Name	Date modified	Туре	Size		
FaultInjection_poco.xls	2022-06-02 오전 9:57	Microsoft Excel 97	14 KB		
File name	e: FaultInjection_poco.xls	~	*.xls		~
			Open		Cancel

- 3. If the source code of the fault injection information you want to import is different from the source code of the Controller Tester project, fault injection can be reused to match the changed source code using [Reconfiguration the fault injection to import] feature.
 - a. If the source code shape is different when performing defect injection import, the following dialog will appear.

Fau Fau	It Injection Reconfiguration Warning	×
▲	Resetting is required because the source code information for fault injection to import is different from the source code information for the current project.	
	OK	

b. You can reconfigure the import of fault injection by clicking the [OK] button.

IoadFile(struct _iobuf *)	type filter text		type filter text	
	<pre>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></pre>	^	<pre>v @ (3) loadFile(struct_lobuf *)</pre>	, fp);
		~		
	Inserted After:	~		

- On the left, the source code of the fault injection information you want to import is displayed.
- On the right, the source code of the current project is displayed.

c. If you click the [OK] button in the [Reconfiguration the Fault Injections to Import] window, you can import the reconfigured fault injection information.

If the file and function information of the fault injection information you want to import are different from the project, reconfigure the function is performed first.

- 4. If you have fault injections already created in the same location as the fault injection information you want to import into your project, you need to decide whether to overwrite the fault injection.
 - a. When importing fault injection, if there is fault injection information in the existing location, the following dialog will appear.

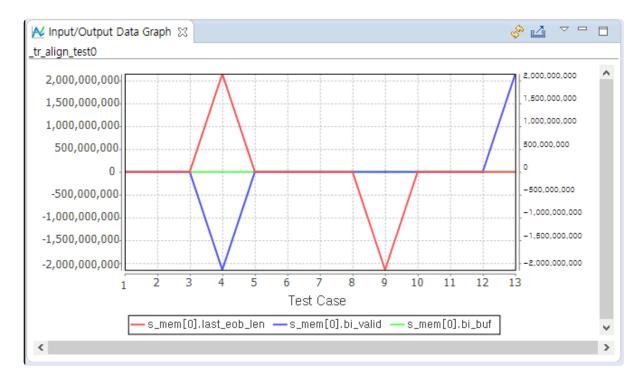
Fault Injection Overwrite Warning	×	(
The fault injection information at the function line position below conflicts with existing fault injection. When overwriting as is, Fault injection information is replaced on a per function basis. Do you want to continue?		
loadFile(struct _iobuf *):252 BEGIN	<	
Overwrite Import excluding conflict functions Cancel		

- The [Fault Injection Overwrite Warning] dialog shows the function, line, and location of the fault injection that does not match the existing information.
- If you select [Overwrite], the existing fault injection information will be overwritten with the fault injection information you want to import.
- If [Import excluding conflict functions] is selected, only fault injection information for functions except for conflict functions is imported.
- Clicking the [Cancel] button cancels the import of fault injections.

Before Controller Tester 3.7, only lines with fault injection code are imported as enabled, as the exported fault injection file did not include enabled or disabled.

16.14. Input/output Data Graph View

The Controller Tester provides the data for input value/expected value/output value in graph format. The horizontal axis of the graph indicates the number of the test case and the vertical axis indicates the data of the test case.



Toolbar icon in Input/output Data Graph view

Toolbar icon	Description
🟜 Refresh	Refreshes the input/output data.
🗳 Save graph	Saves the input/output data.

Click [Save graph] to display the notification window that can enter the path for saving the graph. Enter the path to save, the file name and the file format and click [OK] to save the graph.

Save graph			×
Path: Z:\report\ima	age.jpg		<u>S</u> earch
		ОК	Cancel

Graph settings

Select [Select input/output item] in the pull-down menu to display the Graph settings window.

- For the items selected in the checkbox, the [Input], [Output value] and [Expected value] are displayed in the Input/output Data Graph. However, if the expected values are specified as ~, &, | and ! etc., the expected value is not applied to the Input/output Data Graph view.
- [Number of test cases showed in a view] allows you to specify the number of test cases to be displayed on one screen.

nput		Expect	Output.
☐ int s_mem[0].last_eob_len	s_mem[0].last_eob_len		
unsigned short s_mem[0].bi	s_mem[0].bi_buf	V	
☐ int s_mem[0].bi_valid	s_mem[0].bi_valid		

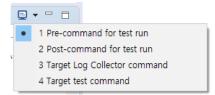
16.15. Console View

Enter the instruction or batch file in [Pre-command for test run] and [Post-command for test run] of [Project] -> [Properties] -> [Test] -> [External Command] and execute the unit test to display the execution result for the external instructions before and after the test execution in the [Console] view.

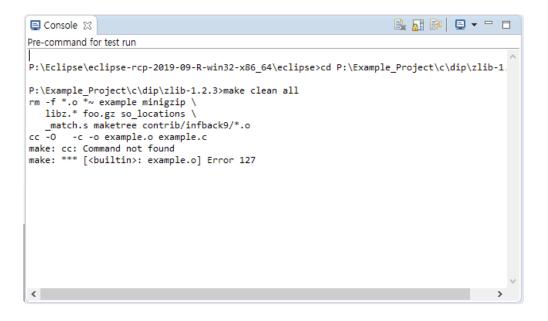
Toolbar icon in the [Console] view

Toolbar icon	Description
Clear Console	Clears the console contents.
Scroll Lock	Locks the scrolling of the [Console] view screen.
Display Selected Console	Shows the selected console.

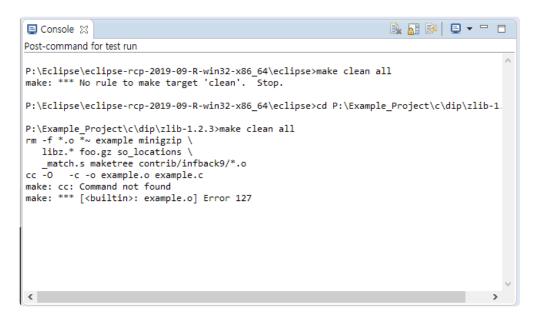
If [Display Selected Console] is selected, the following two menus are displayed: [Pre-command for test run] and [Post-command for test run].



If you select [Pre-command for test run], the execution result for the external command before running the test is displayed in the Console view.



If you select [Post-command for test run], the execution result for the external instructions after the test execution is displayed in the Console view.



16.16. Requirement View

You can import requirements from a requirements management tool or a CSV file into Controller Tester. You can connect imported requirements with tests and export the connected information.

2	Requirem	ent 🛛	d	6 [Ŧ	è 4	
ype	e filter tex	t					
>	🖉 FR11	"Number of tablessss"					
	2 FR12	"Number of seats per table"					
>	🖉 FR13	"List of entrees"					
	2 FR14	"Placing an order updates occupied status"					
>	🖉 FR15	"Placing an order updates number in party"					
	2 FR16	"Placing an order updates a seat's order"					
	2 FR17	"Placing an order updates check total"					
	2 FR18	"Clearing a table resets occupied status"					
>	🖉 FR19	"Clearing a table resets number in party"					
	2 FR20	"Clearing a table resets orders for all seats"					
	2 FR21	"Clearing a table resets check total"					
	2 FR22	"Obtaining check total"					
	2 FR23	"Size of waiting list"					
	2 FR24	"Adding a party to waiting list"					
	2 FR25	"Getting the head of the waiting list"					
	2 FR27	"Adding free dessert"					

Requirement status

Status	Description
🔏 Not connected	A requirement without connected tests
Connected	A requirement with connected tests
Needs Review	A requirement that need to be reviewed

Toolbar Menu in Requirement View

Icon	Description
Connect requirements and tests	Open the [Connect requirements and tests] dialog box.
Expand All	Expand all requirements
E Collapse All	Collapse all requirements
Requirements	Hide unconnected requirements.
🚵 Import	Importing requirements created with a CSV file or a requirements management tool.
🗳 Export	Export requirements to a requirements test coverage report or requirements management tool.

For more information, please refer to the pages <u>Export</u>, <u>Import</u>.

pull-down menu in Requirement View

Icon	Description
Export requirements traceability (csv)	Exporting connection information between requirements and tests.
Export requirements traceability [Including all test cases] (csv)	Export connection information between requirements and tests, including all test cases.
E Import requirements traceability	Importing connection information between requirements and tests.
Export Requirements Traceability Information for V-SPICE	Export connection information between requirements and tests in V-SPICE's xml file format.

Connect tests and requirements

Requirements can be connect to test/test case/integration test.

• Drag a requirement from the Requirements View and drop it on the target you want to connect to in the Test View.

> Zcfree(void *, void *)	(1 / 0 / 0) 1	100.0% (2/2)
> 🗹 zcalloc(void *, unsigned int)	(3 / 0 / 0) 3	100.0% (2/2)
> 🛃 zError(signed int)	(5 / 0 / 0) 5	100.0% (1/1)
> v updatewindow(struct z_stream_s *, const unsigned (ch (0 / 0 / 3) 3	13.3% (4/30)
> vncompress2(unsigmadrRelGar"List Ofrentgees" long *)	(3 / 0 / 0) 3	94.4% (34/36)
> 🗸 uncompress(unsign2d ER14 "Placing:ap.order.updates occup		100.0% (1/1)
Tr_static_init() FR15 "Placing an order updates numb		N/A
> Syncsearch (unsigned FR16 "Placing an order updates a seat		100.0% (13/13)
> Slide_hash(struct@nfelfha"Placing an order updates check	total" (1 / 0 / 2) 3	100.0% (13/13)
🔉 🔽 send tree(struct internal_tate * struct of data		11 0% (40/400)
🖻 Requirement 🕱	6 🗉	- 🛛 🗠 🗠 🔻 🗖
type filter text		
> 2 FR11 "Number of tablessss"		
 		
FR12 "Number of seats per table		
 FR12 "Number of seats per table FR13 "List of entrees" 		
 ⊘ FR12 "Number of seats per table > ₩ FR13 "List of entrees" ⊘ FR14 "Placing an order updates occupied status" 		
 ⊘ FR12 "Number of seats per table > @ FR13 "List of entrees" ⊘ FR14 "Placing an order updates occupied status" > @ FR15 "Placing an order updates number in party" 		
 ⊘ FR12 "Number of seats per table > @ FR13 "List of entrees" ⊘ FR14 "Placing an order updates occupied status" > @ FR15 "Placing an order updates number in party" ⊘ FR16 "Placing an order updates a seat's order" 		

• Drag an item from the test view and drop it on the target you want to connect to in the Requirements View.

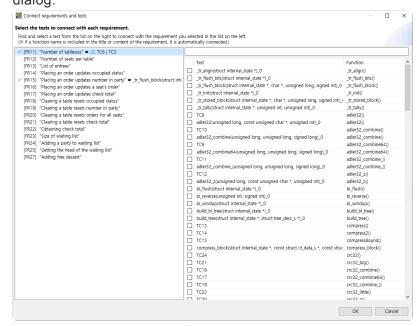
Na	me		Result	Coverage	^
> 🔽	<pre>> zlibVersion()</pre>		(1 / 0 / 0) 1	100.0% (1/1)	
> 🔽	<pre>zlibCompileFlags()</pre>		(3 / 0 / 0) 3	40.4% (19/47)	
> 🔽	<pre>zcfree(void *, void *)</pre>		(1 / 0 / 0) 1	100.0% (2/2)	
> 🔽	<pre>zcalloc(void *, unsigned int)</pre>		(3 / 0 / 0) 3	100.0% (2/2)	
> 🔽	zError(signed int)		(5 / 0 / 0) 5	100.0% (1/1)	
> 🔽	<pre>updatewindow(struct z_stream_s *, cons</pre>	t unsigned ch	(0 / 0 / 3) 3	13.3% (4/30)	
> 🔽	<pre>uncompress2(unsigned char *, unsigned</pre>	long *)	(3 / 0 / 0) 3	94.4% (34/36)	
> 🔽	<pre>uncompress(unsigned char *, unsigned 1</pre>	ong *)	(3 / 0 / 0) 3	100.0% (1/1)	
> 🔽	tr_static_init()		(9 / 0 / 0) 9	N/A	
> 🔽	<pre>syncsearch(unsigned nt *, const unsig</pre>	ned char *	(10 / 0 / 0) 10	100.0% (13/13)	
> 🔽	<pre>slide_hash(struct irnal_state *)</pre>		(1 / 0 / 2) 3	100.0% (13/13)	
5 🔽	send tree(struct innal state * _str	uct ct data s	(4 / 0 / 7) 11	11 0% (40/400)	~
🔁 Re	quirement 🛛 💙		🔏 🖽 🖻	🖹 🖻 🖆 🔻 🖻	
type fi	iter text	(1 / 0 / 0) 1			
> 8	FR11 "Number of tablessss"	(3 / 0 / 0) 3			
0	FR12 "Number of seats per table"	(5 / 0 / 0) 5			
> 8	FR13 "List of entrees" const unsigned ch	(0 / 0 / 3) 3			
0	FR14 "Placing an order updates occupied status"	(3 / 0 / 0) 3			
> 8	FR15 "Placing an order updates number in party"	(3 / 0 / 0) 3			
2	FR16 "Placing an order updates a seat's order"				
	FR17 "Placing an order updates check total"				
0	The final of the state of the s				
0	FR18 "Clearing a table resets occupied status"				

· Connecting via the Connect requirements and tests dialog

1. Select [Connect requirements and tests] from the toolbar menu.

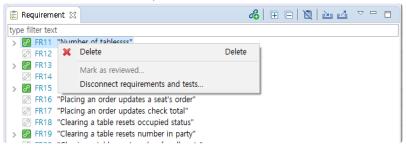


2. You can connect requirements and tests through the [Connect requirements and tests] dialog.



Disconnect requirements and tests

You can disconnect requirements and tests from the context menu in the Requirements View.

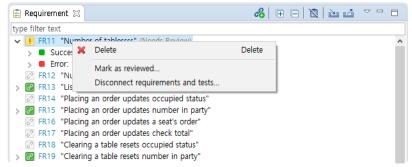


Review of tests connected to requirements

1. When a requirement is re-imported, it will be marked as [Needs Review] if the title or description has changed.



2. After reviewing the requirements and connected tests that need to be reviewed, you can mark them as reviewed with the context menu.



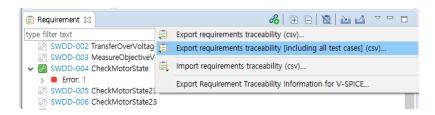
Export requirements traceability

You can export the connection information between requirements and tests to a CSV file.



Export requirements traceability [Including all test case]

You can export the connection information between requirements and tests to a CSV file, including all test cases.



Import requirements traceability

After entering traceability information in the CSV file created by [Export requirements traceability], you can import it through the [Import requirements traceability] menu.



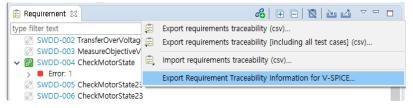
· Requirements traceability file description

Row	Description
TEST_TYPE	Test: 0 Integration Test: 1 Test Case: 2
TEST_NAME	Integration Test: Name Test or Test case: a unique name for the test
TESTCASE_NO	Test case: number Non Test case: 0
REQ_KEY	The key of the requirement to be connected or connected to the test
TEST_ID	Integration Test: Name Test: The name displayed in the test view Test case: The name displayed next to the number in the test view

The unique name of the test can be found by selecting 'Show as Unique Test Name' from the toolbar menu of the test view or by searching.

Export requirement traceability information for V-SPICE

Connection information between requirements and tests can be exported to an xml file conforming to the V-SPICE format.



Open in Requirements Management Tool

Requirements imported from the requirements management tool or tests exported to the requirements management tool can be viewed in the requirements management tool through the [Open with Browser]

context menu.

R	TEST-151 Usr			
~	Not Exect	₩	Open with Browser	
	ExecFi	×	Delete	Delete
0	TEST-152 Use			
0	TEST-156 Use		Mark as reviewed	
0	TEST-157 Use		Disconnect requirements and tests	
2	TEST-159 SVG	PITT	STRUCT THEFT THE TOURWING ALCEDIANCE C	

17. Analysis Perspective

The Analysis Perspective provides a UI for analysis information that can be confirmed from the test results.

The components that make up the analysis perspective are:

- <u>Metrics View</u>
- Metrics Chart View
- <u>Metrics Top Chart View</u>
- Metrics Bar Chart View
- <u>Metrics Diagnosis Chart View</u>
- <u>Unused Function View</u>
- Source-Header Relation View
- Global Variable Relation View
- Eunction Call Hierarchy View
- Source Code Editor section

Views that are not included by default can be opened from [Window]> [Show View]> [Other...] on the top menu.

17.1. Metrics View

The Metrics view shows the result that measures the metric for the project. It shows the summary information for the overall metric and the metric information for each items included in the project (module, file, class, function).

🔤 Metrics 🙁				• 🗳 👜 🗢 🗖
Metrics of 'zlib_merged' project				
✓ Project Summary				
Total Files	40 (13/27)			
Analyzed Files	23 (13/10)			
Number of Functions	113			
Line of Code	7018			
Comments Ratio	52 %			
✓ File Metric				
Metric: pLOC ~ 0 - 100000				
File name	pLOC	LOC	FICRO	^
Compress.c	80 lines	46 lines	50 %	
📄 zlib.h	1358 lines	192 lines	569 %	
zconf.h	333 lines	252 lines	27 %	v .
23 of 23 rows << << <	1 of 1 page >>>>>>	Go to 1 🔹 page 🖂		

lcons

lcon	Description
٢	Module (a logical portion that divides the project for each function)
	File (Source file and header file belonging to the project)
Θ	Class
•	Function

Toolbar menu

Menu	Description
Show Diagnosis	Displays a diagnostic image of the value in the left of the metric value.
😼 Change Mode	Changes the view mode (module, \square file, \bigcirc class, \bullet function).
🗳 Export View Content	Exports the contents of view as a report.
Print View Content	Prints the contents of view.

Pull-down menu

Menu	Description	
Sort by	Sorts based on the selected item (even if clicking the column name in the table, it is sorted equally).	
Columns	Changes the order and width of the columns (allows you to change the order by dragging	

	and dropping the table's column name).
Configure diagnosis	Opens the diagnostics preferences page (allows you to set the diagnostic step and the ranges or images for each metric).
Set to Hide/ Show metric	Sets whether to hide or show the metrics.
Metric View Option	Sets the number of lines per page in the Metric view.

Filtering

ics of 'zlib_merged' project			
Project Summary			
Function Metric			
letric: FUNST v 20 + 10000	0 → Filter matched 42 of 114 items		
Function name	FUNST	FUCYC	·
gz_open	61	in 17	
destroy	23	l 13	
gzread	68	20	
check_header	1 39	l6	
gzwrite	20	9 6	
o gzseek	1 54	23	
do_flush	22	🥥 8	
main	35	l4	
c			>

The filtering allows you to view only those items within the desired range for the specific metric.

Select a metric, enter the range value and click [Filtering] button in the right section to see those values corresponding to the entered information.

If you want to return to the status before filtering, click [Initialize the Filtering Criteria] button on the right side of [Filtering] button.

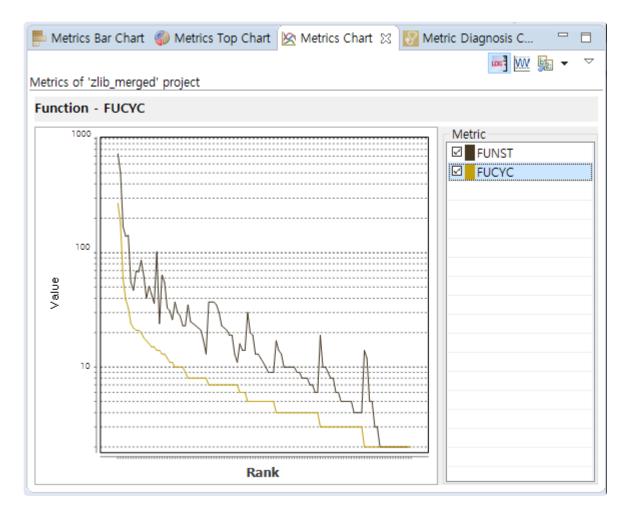
Paging

Metrics 🛛			🛐 💀 🔻 🖾 🖄 🔍 🗖
etrics of 'zlib_merged' project			
Project Summary			
Function Metric			
Metric: FUNST v 0 - 100000			
Function name	FUNST	FUCYC	^
bi_windup	7	🧼 3	
adler32	1 01	l3	
adler32_combine	15	🥥 5	
compress2	19	9 4	
compressBound	👍 1	🥥 1	
compress	🚹 1	🥥 1	
uncompress	<u>iii</u> 22	🥥 6	
gzdopen	<u>6</u> 4	2	
<	2 4 o	A 1	>
100 of 114 rows << << 1	of 2 page >>>>> Go to 1	page 🤕	

The Paging allows you to view many items conveniently.

17.2. Metrics Chart View

The Metrics Chart shows the metric view in a line chart. The x-axis is each item, and the y-axis is the metric value of the item. You can compare the distribution of values between different metrics each other in the project.



Toolbar menu

Menu	Description
Show Log Scale	Changes the y-axis of chart to log units.
We Show Normalized Metric	Shows the standardized metric value.
🔤 Change Mode	Changes the view mode (module, \square file, \odot class, \bullet function).

Pull-down menu

Menu	Description
Sort by	Sorts based on the selected item (even if clicking the column name in the table, it is sorted equally).
Set to Hide/Show	Sets whether to hide or show the metrics.

metric	
Metrics View Options	Sets the connection function with Metric view.

Select Metric

If you select the checkbox left to the metric name in the right table, the checked metrics are displayed in the chart. If you click the metric name in the checked metric, the items are sorted based on that metric value.

17.3. Metrics Top Chart View

The Metrics Top Chart view shows the top items of the metric value in a pie chart. You can see the ratio of the top items in the total.

nction Metrics		Metric: FUNST
op 10 Highest 'FUNST' Values		
	Function name	FUNST
	Other	1569.00
	inflate	735.00
	inflateBack	499.00
	deflate	168.00
	inflate_fast	141.00
	inflate_table	138.00
	adler32	101.00
	send_tree	85.00
	gzread	68.00
	deflate_slow	67.00
	compress_block	63.00

Menu	Description
🌆 Change Mode	Changes the view mode ($\textcircled{\mbox{$\stackrel{\bigcirc$}{$}$}}$ module, $\textcircled{\mbox{$\stackrel{\otimes$}{$}$}}$ file, $\textcircled{\mbox{$\bigcirc$}}$ class, $\textcircled{\mbox{$\bullet$}}$ function).

Pull-down menu

Menu	Description
Metrics Hide/Show Setting	Sets whether to hide or show the metrics.
Metrics View Options	Sets the number of top items to be displayed and sets whether to show the sum of the resting items.

The selection of the pie chart and the selected item in the table are connected together.

17.4. Metrics Bar Chart View

The Metrics Bar Chart view shows the metric values in a bar chart. You can check the value size for each item in the bar chart.

etrics of 'zlib_merged'	project				
unction Metrics			Metric:	FUCYC	V
Function name	FUCYC	Ý			^
inflate				270.0	
inflateBack			178.00		
 deflate 	58.00				
inflate_table	38.00				
inflate_fast	32.00				
gzseek	23.00				
zlibCompileFlags	21.00				
deflate_slow	20.00				
gzread	20.00				
send_tree	19.00				
gz_open	17.00				
check_header	16.00				
deflate_fast	15.00				
gen_bitlen	14.00				
main	14.00				
adler32	13.00				Ļ

lcons

lcon	Description
٢	Module (a logical portion that divides the project for each function)
	File (Source file and header file belonging to the project)
Θ	Class
۲	Function

Toolbar menu

Icon	Description
WW Show Normalized Metric	Shows the standardized metric value.
齄 Change Mode	Changes the view mode ($^{}$ module, $$ file, $$ class, $$ function).

Pull-down menu

Icon	Description
Metrics Hide/Show Setting	Sets whether to hide or show the metrics.
Metric View Options	Sets the number of items to be displayed.

17.5. Metrics Diagnosis Chart View

The Metric Diagnosis Chart view shows the number of items included in the diagnostic range in a pie chart. You can check the number of items within each diagnostic range in the project.

🖶 Metrics Bar Chart 🏼 🌍 Metrics Top Char	t 🛛 🖄 Metrics Chart	Metric Diagnosi	s C 🔀	
Metrics of 'zlib_merged' project				
Function Diagnosis		Ν	letric: FU	ICYC ~
Distribution of Diagnosis	Diagnosis Table			
	Message	Range	Image	Count
	Simple(without	[0.00, 10.00)	۲	92
	More complex(m	[10.00, 20.00)	Ç,	13
	Complex(high risk)	[20.00, 50.00)	4	6
	Greatly complex([50.00, max)	*	3
	None Value			0

Toolbar menu

Menu	Description
hange Mode	Changes the view mode (module, \square file, \bigcirc class, \bullet function).

Pull-down menu

Menu	Description
Configure Diagnosis	Opens the diagnostics preferences page (allows you to set the diagnostic step and the ranges or images for each metric).
Metrics Hide/ Show	Sets whether to hide or show the metrics.
Metric View Options	Sets whether to display the items having no diagnostic value.

The selection of the pie chart and the selected item in the table are connected together.

17.6. Unused Function View

The Unused Function view shows the functions included in the selected project by classifying them according to the specified criteria. It shows the currently unused functions and the criteria for classifying can be added.

™ Unused Function ×		i 🔒 🛱 🕴 🗖 🗖
Total 15 functions - in 'study_sample' project		
Name	Module	File ^
✓ 2 [™] Unused (15 items)		
 OutOfIndex_error 	Default Module	study_sample.c
abc	Default Module	study_sample.c
● api1	Default Module	study_sample.c
• api2	Default Module	study_sample.c
charlesFunc	Default Module	study_sample.c
exit_error	Default Module	study_sample.c
helper	Default Module	study_sample.c
helper_Enm2	Default Module	study_sample.c
loadFile	Default Module	study_sample.c
• main	Default Module	study_sample.c
 mcdcError 	Default Module	study sample c *

Menu	Description
Save as PNG	Save unused function view as image.
🛱 Copy to System clipboard	Copy unused function view image to system clipboard.
🛱 Copy Text to System clipboard	Copy unused function view content to system clipboard.

lcons

lcon	Description
<u> </u>	Classification criteria
<u></u>	Function group
۲	Function

Pull-down menu

Menu	Description
Function View Options	Sets the number of functions to be displayed for each group.

17.7. Source-Header Relation View

Source-Header Relation view shows <include> and #include relations between the source file and the header file

1 ■ Source-Header Relation 🔀	() 🐎 📲 🗖
13 files included by 'zutil.c' - in 'zlib' project	
🗸 🖻 zutil.c	
🔓 _mingw.h	
📓 msvcrtver.h	
📓 stddef.h	
📓 string.h	
📓 w32api.h	
📓 strings.h	
h zlib.h	
📓 sdkddkver.h	
h zconf.h	
📓 wchar.h	
📓 errno.h	
n zutil.h	
📓 stdlib.h	

lcons

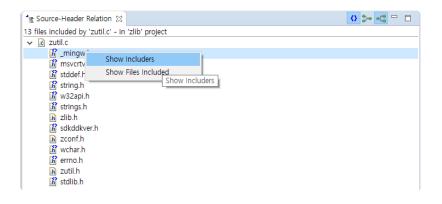
lcon	Description	
. c	Source file	
.h	Header file	
ĥ	System header file	

Toolbar menu

Menu	Description
Show System Header	Shows the system header.
Show Includers	Shows the files that includes the selected file.
Show Files Included	Shows the file included in the selected file.

Show Includers, Show Files Included

You can display the files that includes the selected file or the files that the selected file is included by using the menu displayed when right-clicking a file in the view.



17.8. Global Variable Relation View

The Global Variable Relation view shows the function where the global variable is defined and used. If a function is selected, it shows the global variables defined or used in the function, and if a global variable is selected, it shows the functions where the global variable is defined or used.

\ddagger Global Variable Relation $ imes$		🖬 🔓 🛱 🗖 🗖					
'Left' - 8 relations in project 'Sample2_copy'							
Name	Definition	Use					
✓ ● Left							
bend		✓					
bend_no		✓					
body		✓					
body	✓						
head		✓					
Ien		✓					
Ien	✓						
length		✓					

Menu	Description
Save as PNG	Save global variable relation view as image.
Copy to System clipboard	Copy global variable relation view image to system clipboard.
🛱 Copy Text to System clipboard	Copy global variable relation view content to system clipboard.

lcons

lcon	Description	
۲	Function	
۲	Global Variable	

Show Relation

You can view the relation for the selected items again by using the menu displayed when right-clicking the related item in the view.

'Left' - 8 relations in project 'Sample2_co	py'	
Name	Definition	Use
✓ ● Left		
bend		✓
bend_po		✓
body Show Relation		✓
body	 Image: A set of the set of the	
head		✓
😑 len		 Image: A set of the set of the
len	v	
length		✓
_		

17.9. Function Call Hierarchy View

The Function Call Hierarchy View shows the information for the function called by the selected function or the called function in the hierarchy structure.

You can change the call information for the selected function by using the context menu that appears when selecting and right-clicking the function displayed in the hierarchy structure and double-click it to open the editor for the source file that the function is defined and to go to the location of the selected function. (However, not included if it is a system function)

all Hierarchy 🗙 🕂 🖶 🖬 🛱 🛱 🗖 🗖
ht' - in 'Sample2_copy'
10
XY(signed int, signed int)
tO
f(const char *,)
-

Fui	nction Call Hiera	rchy ×			\oplus)+e •(*e		a G		
s f	rom 'Right' - in 'S	Sample	2_copy'							
•	Right()									
>	Bend()									
>	GotoXY(signe	d int. s	ianed int)							
	++ kbhit()	-	· · · ·							
	+ printf(const cl	et <mark>e</mark> Sh	iow Callee Hierarchy							
	s f	s from 'Right' - in 'S Right() S Bend() S GotoXY(signed H kbhit()	s from 'Right' - in 'Sample Right() > Bend() > GotoXY(signed int. si ++ kbhit()	 Bend() GotoXY(signed int. signed int) 	s from 'Right' - in 'Sample2_copy' ● Right() > ● Bend() > ● GotoXY(signed int. signed int) ++ kbhit() ■ Show Caller Hierarchy	s from 'Right' - in 'Sample2_copy' ● Right() > ● Bend() > ● GotoXY(signed int. signed int) ++ kbhit() \$+• Show Caller Hierarchy	s from 'Right' - in 'Sample2_copy' Right() Bend() GotoXY(signed int. signed int) H kbhit() 	s from 'Right' - in 'Sample2_copy' ● Right() > ● Bend() > ● GotoXY(signed int. signed int) ++ kbhit() \$+• Show Caller Hierarchy	s from 'Right' - in 'Sample2_copy' Right() > Bend() > GotoXY(signed int. signed int) + kbhit() Show Caller Hierarchy	s from 'Right' - in 'Sample2_copy' Right() Bend() GotoXY(signed int. signed int) Herrichy

Menu	Description
++ Show System Function	Sets whether to indicate the call information for the system function in a hierarchical structure.

Show Caller Hierarchy	Shows the functions that call the selected functions in a hierarchical structure.
• 🛱 Show Callee Hierarchy	Shows the functions called by the selected function in a hierarchical structure.
🕼 Save as PNG	Save function call hierarchy layer view as image.
Copy to System clipboard	Copy function call hierarchy view image to system clipboard.
Copy Text to System clipboard	Copy the Function Call Hierarchy View contents to system clipboard as text

17.10. Source Code Editor section

The source code editor section is a pre-reserved area, located on the left of the screen.

CodeScroll Controller Tester RTV(Remote Target Verifier) + Target Plugin (x64)		- 🗆 ×
	Test Import Excort Properties Preferences Host 🌒 Target 👱 💱 Global Search (Ctrl+3)	Q) 🖻 🖃 🏶
F B Show Coverage	🗈 Coverage 🚽 MC/DC 🔃 Stub 🦹 Class Factory 🚭 Control Flow Graph 👘 Call Graph 🦹 Fun	ction Call Hierarchy 🎦 Erro
le adler32.c ☆ 🖓 🖓	Unit Test Integration Test	🖹 👺 🗄 🔹 🌣 🗖
50 } while (0) ^		✓ Statement Coverage
52. # define NOD(a) a %- BASE 53. # define NOD(a) a %- BASE 54. #endif 55	Run ► (18 / 0 / 6) 24	4 2.9%
56 /**/	type file, function, test, state, issue	
57 ⊕ uLong ZEXPORT adler32(adler, buf, len) 58 uLong adler;	Name Result	Coverage
<pre>59</pre>	Cases 3 Signaled Cases 4 Signaled Cases 6 Cases 6 Cases 7 Cases 9 Signaled	
<pre>67 aller å= 0xfff; 68 69 /* in case user likes doing a byte at a time, keep it fast */ 70[rf] if (len = 1) { 71 adler += buf[0]; 72[r] if (adler → EASE) 73 adler → EASE 74 sum2 += adler; 75[r] if (sum2 → EASE) 76 sum2 == BASE;</pre>	Case 10 Case 11 Case 11 Case 12 Case 13 Case 14 Case 14 Case 14 Case 15 Case 14 Case 16 Combine(unsigned long, unsigned long) (11/0/0)11 C Case 2000 Case 14 Case 2000 Case 14	100.0% (18/18)
77 return adler (sum2 << 16); 78 } 79	Test Info (zlit/adler32_test0)	ى ئە ئە
<pre>80 /* a initial Adler-32 value (deferred check for len == 1 speed) */ 81[F] if (buf == Z_NULL)</pre>	Test Structure 🔍 💽 🔁 Test Info Edit	
82 return 1L;	Test structure using a tree view and edit the information in the test. Change the notation edit the partition	^
84 /* in case short lengths are provided, keep it somewhat fast */ 85[rr] if (len x i0) { 86[rr] while (len) { 87 adler +* buffsj	Name In Out Set Numeral System ✓ Parameter/return 0	System
89 }	v o buf : const Bytef *	
90[TF] if (adler >= BASE) 91 adler -= BASE:	✓ • [0] : Bytef Min ~ Max	Add
92 MOD4(sum2); /* only added so many BASE's */ 93 return adler (sum2 << 16); 94 }		Add
95 96 /* do length NMAX blocks requires just one modulo operation */	o len : unsigned int ✓ o [®] returnVar : uLong "0"	Delete
99 7 06 40 mg of 1 MRXX (99 1 m + MRXX) (90 1 m + MRXX / 16; /* MRXX is divisible by 16 */ 109 do (Default
	Test Info Test Case Test Code Configuration	
	257M of 1038M	

18. File Menu

In File menu, Create a new project or module, Editor-related operation, Refresh, Change Workspace, Import and Export, View Properties of the selected item and Exit Tool etc. can be carried out.

File	Edit	Search	Project	Target	Window
	New				>
	Close			C	Ctrl+W
	Close	All		Ctrl+Sł	nift+W
	Save				Ctrl+S
D	Save /	All		Ctrl+S	hift+S
	Renan	ne			F2
	Refres	h			F5
	Switc	h Worksp	ace		
2	Impor	t			
4	Expor	t			
	Prope	rties		Alt	+Enter
	Exit				

- New:
 - Create a project
 - Create a module
- Close/Close all/Save/Save all: Editor-related menu
- Rename: You can modify the name of project or module.
- <u>Switch workspace</u>
- Import
- Export
- Exit: Exit tool.

18.1. Create a Module

You can create a new module in the project. The information other than the module name is created the same as the modules already existed in the target project.

New Module			×
Module			
Create a new module.			
Select the project:			
🔀 stub_sample ₩ zlib			
Module <u>n</u> ame:			
	<u>F</u> inish	Cance	el

- 1. In [New] menu, click [Other...], select [Module] in [Other] and click [Next].
- 2. Select the project that you want to create a new module.
- 3. Enter the module name. The module name existed already cannot be used.
- 4. Click [Finish] to create the module.

*	You can also create a module by using the [New] menu displayed when right-clicking a
	project in Test Navigator view.

18.2. Switch Workspace

You can switch the directory of workspace. When you witch the workspace, the tool restarts with the switched workspace.

Switch Workspace)			×
Select a directory as workspace Controller Tester uses the workspace directory to store its preferences and projects				
Workspace: □ Use this as the	C:₩codescroll e default and do not ask again		~	Browse
Clear History		Clone	ОК	Cancel

Click [Browse...] to select the workspace directory to be switched or to enter it manually. You can select the workspace directory selected previously in the list displayed when clicking [$\mathbf{\nabla}$].

If you select the checkbox [Use this as the default and do not ask again], you are not prompted again to select a workspace directory the next time you execute the tool.

If you click [Clone], the selected workspace is copied to the other directory.

If you click [Clear History], you can delete the workspace list.

18.3. Import

Controller Tester provides a feature that import project and data used other version or other PC.

1. Select [File] > [Import] from the main menu or [Import] from the dashboard to open the Import Wizard.

File	Edit Search	Project	Target	Window
	New			>
	Close		C	Ctrl+W
	Close All		Ctrl+Sł	nift+W
	Save			Ctrl+S
ß	Save All		Ctrl+S	hift+S
	Rename			F2
	Refresh			F5
	Switch Workspa	ace		
è	Import			
4	Export			
	Properties		Alt	+Enter
	Exit			

2. Select a item to import and click [Next].

Import -		×
Select Import the project, including project settings and tests.	Ľ	5
Select an import wizard:		
type filter text		
 Coverage Import Coverage File System File System Import Project Preferences ToolChain Virtual Memory Address Requirement Import from CSV file Import from V-SPICE Test Import test Import test using test code file 		
< Back Next > Finish	Cance	I

Coverage	Import Coverage	import coverages in csd format.
General	File System Import project	Import files or projects.
Preferences	ToolChain Virtual Memory Adress	Import toolchains or virtual memory adress exported before.
Requirement	Import from CSV file Import from Polarion Import from V-SPICE	Import requirements from CSV files or requirement management tools.
Test	Import test Import test using test code file	import tests exported before or tests from test codes.

18.3.1. Requirement – Import from CSV file

You can import requirement documents written in CSV file to Controller Tester.

1. In the Import Wizard, select [Requirement] -> [Import from CSV file] and click [Next].

Import	– 🗆 X
Select Import requirements from a CSV file.	Ľ
Select an import wizard:	
type filter text	
 > Coverage > General > Preferences > Requirement Import from CSV file Import from Polarion > > Test 	
< Back Next > Fini	sh Cancel

2. Enter informations for importing requirements and click [Finish].

Import req	_	×	
File Selection			
	uirements_example.csv	В	rowse
Configuratio			
Value separa	tor: ,		~
Encoding:	US-ASCII		~
format:	EXCEL		~
The CSV f	ile contains a header		
Assgin Attrib	ute		
Key:	Кеу		~
Title:	Title		~
Description:	Description		~
	< Back Next >	Finish Ca	ancel

- a. File Selection
 - Enter the path of a file that the requirements are stored.
- b. Configuration
 - Value seperator : Enter a separator that separate each value in the CSV file. Default value is ,(comma).
 - Encoding : Enter a encoding of the CSV file. Default value is US-ASCII.
 - format : Enter a format of the CSV file. Default value is EXCEL.
 - The CSV file contains a header : Check when the CSV file contains a header.
- c. Assign Attribute
 - Key : Select the column that corresponds to the key of the requirements.
 - Title : Select the column that corresponds to the title of the requirements.
 - **Description** : Select the column that corresponds to the description of the requirements.

The information you entered in the previous import is automatically filled in.

18.3.2. Requirement – Import from Polarion

You can import requirement documents from Polarion to Controller Tester.

1. In the Import Wizard, select [Requirement] -> [Import from Polarion] and click [Next].

Import		\Box \times
Select Import requirements from Polarion.		Ľ
Select an import wizard:		
type filter text		
 > Coverage > General > Preferences > Requirement Import from CSV file Import from Polarion > > Test 		
< <u>B</u> ack <u>N</u> ext >	<u>F</u> inish	Cancel

2. Enter the information to connect with Polarion server and click [Next]. You can see the entered password to click o button.

Import re	equirements from Polarion		\times
Import requ	uirements from Polarion		
Enter the in	formation to access the Polarion server.		
Server:	https://almdemo.polarion.com		
Port:	443		
Username:	admin		
Password:	****		O
			8
	< Back Next > Finish	Canc	ol
	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cano	ei

- The information of Polarion's demo server is entered in server and port.
- If you use a local server, enter http://ip_address in server and port number of local server in port.
- 3. When you enter the Project, Document, and Type to import from Polarion, the corresponding requirements list appears. Check the requirements and click the [Finish] button to import the requirements.

Import requirements from Polarion –			
Import requ	irements from Polarion		
Select the in from.	formation of what you want to i	mport requirements	
Project:	TEST(TEST)		~
-			•
Document:	Specification/Catalog Specification	tion	¥
Туре:	Requirement		Y
ID	Title	Description	^
TEST-101	User name must be validate	User name must be validated when a n	ε
TEST-147	Password must be validated	Password must be validated when the u	i I
TEST-209	Password must contain at l	Password must contain at least 8 chara	c
TEST-210			
TEST-146	User name may not contair	User name may not contain spaces	
TEST-152	User name must be longer	User name must be longer than 3 chara	
TEST-151	User name must contain at	User name must contain at least one nu	1
TEST-149	Password must differ from	Password must differ from the user nan	r
TEST-148		Password must contain at least one nur	
TEST-156	User must he informed on t	User must be informed on the login scr >	×
	< <u>B</u> ack <u>N</u> ex	t > <u>F</u> inish Cancel	

- If you do not select the Document, you can also import requirements that are not included in a specific document.
- 4. [Connect requirements and tests] dialog appears where you can connect the imported requirements and tests. If the title or description of the requirement contains the name of the function associated with the test, it is automatically linked. Click [Cancel] to finish [Connect requirements and tests] without linking requirements and tests.

ect the tests to connect with each requirement.				
ind and select a test from the list on the right to connect with the requirement you $\%$ if a function name is included in the title or content of the requirement, it is aut	u sele oma	ected in the list on the left. tically connected.)		
TEST-101] User name must be validated when a new user account is created				
TEST-146] User name may not contain spaces		Test	Function	
TEST-147] Password must be validated when the user wants to change it	_			
TEST-148] Password must contain at least one number	븜	_tr_align(struct internal_state *)_0	_tr_align()	
TEST-149] Password must differ from the user name	븜	_tr_align(struct internal_state *)_0_copy_0	_tr_align()	
[EST-151] User name must contain at least one number	Ц	_tr_flush_bits(struct internal_state *)_0	_tr_flush_bits()	
[EST-152] User name must be longer than 3 characters		_tr_flush_block(struct internal_state *, char *, unsigned long, signed int)_0	_tr_flush_block()	
[EST-156] User must be informed on the login screen when Caps Lock is turned (_tr_init(struct internal_state *)_0	_tr_init()	
TEST-157] User may not be told if the password or user name is wrong		_tr_stored_block(struct internal_state *, char *, unsigned long, signed int)_(
TEST-159] System should meet the following acceptance criteria Use Case Accept	Ц	_tr_tally(struct internal_state *, unsigned int, unsigned int)_0	_tr_tally()	
[EST-160] System must support all following web browsers Browser Support Leve		adler32(unsigned long, const unsigned char *, unsigned int)_0	adler32()	
[EST-209] Password must contain at least 8 characters Password must contain at		adler32(unsigned long, const unsigned char *, unsigned int)_0	adler32()	
EST-210]		adler32_combine(unsigned long, unsigned long, signed long)_0	adler32_combine()	
		adler32_combine(unsigned long, unsigned long, signed long)_0	adler32_combine()	
		adler32_combine64(unsigned long, unsigned long, signed long)_0	adler32_combine64()	
		adler32_combine64(unsigned long, unsigned long, signed long)_0	adler32_combine64()	
		adler32_combine_(unsigned long, unsigned long, signed long)_0	adler32_combine_()	
		adler32_combine_(unsigned long, unsigned long, signed long)_0	adler32_combine_()	
		adler32_z(unsigned long, const unsigned char *, unsigned int)_0	adler32_z()	
		adler32_z(unsigned long, const unsigned char *, unsigned int)_0	adler32_z()	
		bi_flush(struct internal_state *)_0	bi_flush()	
		bi_reverse(unsigned int, signed int)_0	bi_reverse()	
		bi_windup(struct internal_state *)_0	bi_windup()	
		build_bl_tree(struct internal_state *)_0	build_bl_tree()	
		build tree(struct internal state *, struct tree desc s *) 0	build tree()	
	Π	compress(unsigned char *, unsigned long *, const unsigned char *, unsign	compress()	
	Π	compress2(unsigned char *, unsigned long *, const unsigned char *, unsig	compress2()	
	Π	compressBound(unsigned long)_0	compressBound()	
	П	compress_block(struct internal_state *, const struct ct_data_s *, const struct	1 .	
		crc32(unsigned long, const unsigned char *, unsigned int)_0	crc32()	
		crc32_big(unsigned long, const unsigned char *, unsigned int)_0	crc32_big()	
		crc32_combine(unsigned long, unsigned long, signed long)_0	crc32 combine()	
	H	crc32_combine64(unsigned long, unsigned long, signed long)_0	crc32 combine640	
	H	crc32 combined (unsigned long, unsigned long, signed long) 0	crc32_combine04()	
	H	crc32_tombine_(unsigned long, const unsigned long, signed long)_0	crc32_little0	

18.3.3. Requirement – Import from V-SPICE

You can import requirement documents from V-SPICE to Controller Tester.

1. In the Import Wizard, select [Requirement] > [Import from V-SPICE] and click [Next].

Import		
Select Import requirements from V-SPICE.		Ľ
Select an import wizard:		
type filter text		
 Coverage General Preferences Requirement Import from CSV file Import from Polarion Import from V-SPICE Test 		
< Back Next >	Finish	Cancel

Enter the information to connect with V-SPICE server and click [Next]. You can see the entered password to click obutton.

Import re	equirements from V-SPICE	_	_		×
Import requ	uirements from V-SPICE				
Enter the in	formation to access the V-SPICE server.				
Server:	http://				
Port:	38080				
Username:					
Password:					Θ
	Deale Name	. t. l.		Canada	
	< Back Next > Fir	nish		Cance	4

3. When you enter the Project to import from V-SPICE, the corresponding requirements list appears. Check the requirements and click the [Finish] button to import the requirements.

Import r	equirements from	V-SPIC	E			_	_		×
Import req	uirements from	V-SPIC	E						
Select the information of what you want to import requirements from.									
Project:	CN5_TNR_GLB_T	PU							*
Doc Type:	ALL (ALL)								¥
ID		Title				Descript	tion		
CN5_TNR	GLB_TPU_REQ_5	CN5_T	NR_GLB_TF	U_REQ_5	5	CN5_TN	IR_GLE	3_TPUR	EQ.!
CN5_TNR	GLB_TPU_REQ_0	CN5_T	NR_GLB_TF	U_REQ_0)	CN5_TN	IR_GLE	3_TPUR	EQ_(
CN5_TNR	GLB_TPU_REQ_1	CN5_T	NR_GLB_TF	U_REQ_1		CN5_TN	IR_GLE	3_TPUR	EQ_
CN5_TNR	GLB_TPU_REQ_2	CN5_T	NR_GLB_TF	U_REQ_2	2	CN5_TN	IR_GLE	3_TPUR	EQ
CN5_TNR	GLB_TPU_REQ_3	CN5_T	NR_GLB_TF	U_REQ_3	3	CN5_TN	IR_GLE	3_TPUR	EQ
CN5_TNR	GLB_TPU_REQ_4	CN5_T	NR_GLB_TF	U_REQ_4	ļ.	CN5_TN	IR_GLE	3_TPUR	EQ_4
CN5_TNR	GLB_TPU_REQ_6	CN5_T	NR_GLB_TF	U_REQ_6	6	CN5_TN	IR_GLI	3_TPUR	EQ
CN5_TNR	GLB_TPU_REQ_7	CN5_T	NR_GLB_TF	U_REQ_7	7	CN5_TN	IR_GLI	3_TPUR	EQ
CN5_TNR	GLB_TPU_REQ_8	CN5_T	NR_GLB_TF	U_REQ_8	3	CN5_TN	IR_GLE	3_TPUR	EQ
CN5_TNR	GLB_TPU_REQ_9	CN5_T	NR_GLB_TF	U_REQ_9)	CN5_TN	IR_GLE	3_TPUR	EQ_!
	< Ba	ck	Next :	>	Fir	nish		Cance	el l

4. [Connect requirements and tests] dialog appears where you can connect the imported requirements and tests. If the title or description of the requirement contains the name of the function associated with the test, it is automatically linked. Click [Cancel] to finish [Connect

requirements and tests] without linking requirements and tests.

ect the tests to connect with each requirement.				
nd and select a test from the list on the right to connect with the requirement yo If a function name is included in the title or content of the requirement, it is aut				
ST-101] User name must be validated when a new user account is created				_
EST-146] User name may not contain spaces		Test	Function	
ST-147] Password must be validated when the user wants to change it		tr align(struct internal state *) 0	tr align()	
ST-148] Password must contain at least one number	븜	tr align(struct internal state *) 0 copy 0		
ST-149] Password must differ from the user name	븜	_tr_flush_bits(struct internal_state *)_0_copy_0 _tr_flush_bits(struct internal_state *)_0	_tr_align() _tr_flush_bits()	
ST-151] User name must contain at least one number		_tr_flush_block(struct internal_state *, char *, unsigned long, signed int)_0	_tr_flush_block()	
ST-152] User name must be longer than 3 characters	님			
ST-156] User must be informed on the login screen when Caps Lock is turned (님	_tr_init(struct internal_state *)_0	_tr_init()	
ST-157] User may not be told if the password or user name is wrong	님	_tr_stored_block(struct internal_state *, char *, unsigned long, signed int)_(
ST-159] System should meet the following acceptance criteria Use Case Accept		_tr_tally(struct internal_state *, unsigned int, unsigned int)_0	_tr_tally()	
ST-160] System must support all following web browsers Browser Support Leve	Ц		adler32()	
ST-209] Password must contain at least 8 characters Password must contain at ST-2101		adler32(unsigned long, const unsigned char *, unsigned int)_0	adler32()	
51-210]		adler32_combine(unsigned long, unsigned long, signed long)_0	adler32_combine()	
		adler32_combine(unsigned long, unsigned long, signed long)_0	adler32_combine()	
		adler32_combine64(unsigned long, unsigned long, signed long)_0	adler32_combine64()	
		adler32_combine64(unsigned long, unsigned long, signed long)_0	adler32_combine64()	
		adler32_combine_(unsigned long, unsigned long, signed long)_0	adler32_combine_()	
		adler32_combine_(unsigned long, unsigned long, signed long)_0	adler32_combine_()	
		adler32_z(unsigned long, const unsigned char *, unsigned int)_0	adler32_z()	
		adler32_z(unsigned long, const unsigned char *, unsigned int)_0	adler32_z()	
		bi_flush(struct internal_state *)_0	bi_flush()	
		bi_reverse(unsigned int, signed int)_0	bi_reverse()	
		bi_windup(struct internal_state *)_0	bi_windup()	
		build_bl_tree(struct internal_state *)_0	build_bl_tree()	
		build_tree(struct internal_state *, struct tree_desc_s *)_0	build_tree()	
		compress(unsigned char *, unsigned long *, const unsigned char *, unsign	compress()	
		compress2(unsigned char *, unsigned long *, const unsigned char *, unsig	compress2()	
		compressBound(unsigned long)_0	compressBound()	
		compress_block(struct internal_state *, const struct ct_data_s *, const struct	compress_block()	
		crc32(unsigned long, const unsigned char *, unsigned int)_0	crc32()	
		crc32_big(unsigned long, const unsigned char *, unsigned int)_0	crc32_big()	
		crc32_combine(unsigned long, unsigned long, signed long)_0	crc32_combine()	
		crc32_combine64(unsigned long, unsigned long, signed long)_0	crc32_combine64()	
		crc32_combine_(unsigned long, unsigned long, signed long)_0	crc32_combine_()	
		crc32 little(unsigned long, const unsigned char * unsigned int) 0	crc32 little0	

18.3.4. General – File System

You can add the source files to the project by using [Import] – [File System] feature.

1. In the Import Wizard, select [General] -> [File System] and click the [Next] button.

Import	—	
Select Import from File System.		Ľ
Select an import wizard:		
type filter text		
 > > Coverage > General > File System > Import project > > Preferences > > Pequirement > > > Test 		
< Back Next > Finish		Cancel

2. Enter the information of the file to be imported and click the [Next] button.

🐖 Import		_		×
Import				
Select source files.				
Top Directory: C:\Program Files\Sures	oft#CodeScroll Control	er Tester 3.5₩	Bro	wse
Directory	File			
✓ ■ examples ✓ ■ c	type filter text			
> Sample	☑ sample.c			
Text file written list of source files			Brov	vse
X The entered text file is separated	by ';' or newline chara	cter.	Rem	nove
1 items selected				
Into project: test			Bro	wse
< Back	Next > Fin	ish	Cance	1
Item	Descriptio	on		
Top Directory		d'un stam.	- f 41	ne source

ltem	Description
Top Directory	Select the directory of the source files to be imported.
Directory, File	Select the source files to be imported.
Text file written list of source files	Select the directory of the txt file containing a list of source files.(Option)
Into project	Select the project to add the imported source files.

3. Select the module that the source file will be added from the list of modules included in the project and click the [Finish] button.

	Import	:					×
I	File Syste	m					
	Select mo	odules to add.					
		dule to add:					
	🗹 Defa	ult Module				<u>S</u> elect	t All
						<u>D</u> esele	ct All
	<				>		
		< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish		Cance	4
			INCAL >	<u>r</u> man		Cance	

18.3.5. General – Import project

Using the Import Project feature, you can import a project exported from another PC.

1. In the Import Wizard, click [General] > [Import Project] and then click the [Next] button.

Import	
Select Import the project, including project settings and tests.	Ľ
Select an import wizard:	
type filter text	
 > > Coverage > General > File System > Import project > > Preferences > > Requirement > >> Test 	
< Back Next > Finish	Cancel

 Click the [Next] button after entering the exported project directory and selecting the toolchain. If the exported project has toolchain and source code information, Include Toolchain and Include Source Files are automatically selected when you select the directory.

Import		- D X
Import a Project Import the exporte	ad project.	
Project directory:	C:#Users#hikim#Downloads#test_vs2013_tch_src_e	xport_imported2_20220428102501 Search
Project name:	test_vs2013_tch_src_export_imported	
Location:	C:#Users#hikim#Desktop#ct_workspace	
Select Toolchain		
Default	Toolchain	Description
	hyeintoolchain	
	targetToolchain	
	GCC 4.7 (32bit)	자동으로 생성되었습니다.
	GCC 5.3 (32bit)	자동으로 생성되었습니다.
Options Include Toolcha Include Source		Toolchain Setting
	< Back	Next > Finish Cancel

If you don't use the include toolchain option without a suitable toolchain, you need to import the toolchain using the toolchain export/import feature before importing the project.

3. You can check the path information included in the project to be imported. Invalid paths are marked in red and can be modified by clicking with the mouse or pressing the F2 key.

🔣 Import	- • •
Check the paths included in the project	
A Invalid path: 15/15	
Path(click to edit):	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\adler32.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\crc32.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\deflate.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\gzclose.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\gzlib.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\gzread.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\gzwrite.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\infback.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\inffast.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\inflate.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\inftrees.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\trees.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\uncompr.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.11\zutil.c	
C:\Users\seo hyunji\Desktop\target plugin\zlib-1.2.11\zlib-1.2.13\compress.c	
< Back Next > Finish	Cancel

4. If you edit one path, all paths that can be found in the relative path are automatically corrected. The number of modified paths is displayed at the top of the wizard. After editting, click the [Finish] button.

Import	
Check the paths included in the project	
() Invalid path: 0/15	
15 paths have been resolved.	
Path(click to edit):	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\adler32.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\compress.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\crc32.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\deflate.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\gzclose.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\gzlib.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\gzread.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\gzwrite.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\infback.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\inffast.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\inflate.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\inftrees.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\trees.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\uncompr.c	
C:\Users\vagrant\Desktop\source\zlib1211\zlib-1.2.11\zutil.c	
< Back Next > Finish	Cancel

18.3.6. Coverage – Import Coverage

If the version, the coverage type, and the ternary operator option of the coverage file to be imported are the same as those of Controller Tester, the coverage can be imported. For details, please refer to the <u>Coverage Import Guide</u> page of User Guides.

1. In the [Import] wizard, select [Coverage] > [Import Coverage] and click [Next].

Import	$ \Box$ \times
Select Import external coverage to the selected project. (Coverage shared data file, *.csd)	Ľ
Select an import wizard: type filter text	
 > Coverage > General > ≥ Requirement > ≥ Test 	
< Back Next > Fin	ish Cancel

2. Enter the path of coverage file to be imported and click [Finish].

Import coverage Import coverage Select the file for		_		×
File path:	Z:₩export₩zlib.csd		Brow	
	< <u>B</u> ack <u>N</u> ext > <u>F</u> inis	h	Cance)

The imported coverage information can be checked by selecting [Show full coverage (Include External Coverage)] in the toolbar menu of the Coverage View.

	Target Function	Statement	Branch	MC/DC	Function Call	Functi	^
		•	•	•	-	+	
1	_tr_align(struct internal_state *)	•• 72.72%	— 70.00%	4 0.00%	1 00.00	Υ	
2	_tr_flush_block(struct internal_state	== 54.76%	= 62.50%	= 36.36%	= 70.00%	Y	
3	_tr_init(struct internal_state *)	— 100.00%	□ N/A	□ N/A	= 100.00	Y	
4	_tr_stored_block(struct internal_stat	— 100.00%	— 100.00%	— 100.00%	1 00.00	Υ Y	
5	_tr_tally(struct internal_state *, unsi	= 22.22%	— 0.00% (— 0.00% (□ N/A	Υ	
6	adler32(unsigned long, const unsig	98.05%	= 87.50%	••• 75.00%	□ N/A	Υ	
7	adler32_combine(unsigned long, u	— 100.00%	— 100.00%	— 100.00%	□ N/A	Υ Y	
8	bi_flush(struct internal_state *)	— 100.00%	— 100.00%	— 100.00%	□ N/A	Υ	
9	bi_reverse(unsigned int, signed int)	— 100.00%	— 100.00%	— 100.00%	□ N/A	Υ	
10	bi_windup(struct internal_state *)	— 100.00%	— 100.00%	— 100.00%	□ N/A	Υ Y	
11	build_bl_tree(struct internal_state *)	— 100.00%	••• 75.00%	= 50.00%	— 100.00	Υ Y	
Fotal		43.02% (28.51% (19.07% (44.11% (100.0	4

18.3.7. Test – Import test

1. In the Import Wizard, select [Test] > [Import Test] and click [Next].

Import	_	
Select Import test information.		Ľ
Select an import wizard:		
type filter text		
 > Coverage Coverage General Existing Projects into Workspace File System > Preferences > Test Import test Import test using test code file 		
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	h	Cancel

2. Enter the information for the test to be imported and click [Finish].

Import tests		\times
Import tests		
Import project's test informations.		
Import path:	Searc	:h
Unit Test ☑ Test ☑ Test Data		
Integration Test		
Stub O Connected Stub All Stub		
Includes Fault Injection		
Option O If the same stub exists already, it is not imported. If the same stub exists, it is added as a new stub.		
< Back Next > Finish	Cancel	

- a. In [Import path], enter the path to import the test.
- b. In [Unit Test] group, check whether to import [Test] and [Test Data] respectively.
- c. In [Integration Test] group, check whether to import [Test].
 - When importing the integration test, it is imported with the test data in a set.
- d. In [Stub] group, select the stub to be imported among [Connected stubs] and [All stubs].
 - **Connected stubs**: It imports only the stubs associated with the test you want to import.
 - All Stub: It imports all the stubs in the path to import.
- e. In [Includes] group, select whether to import with fault injections.
- f. In [Option] group, select the method to import if stubs exist.
- 3. If there are items that the name has been changed or the file has been deleted when importing the test information, the notification windows is displayed, allowing you to select whether to import the information except for that items. This notification window is displayed only if the information was exported from Controller Tester 2.6.14 or later version at a time. If the information is exported from the other menu and previous version, the notification is not displayed.

import		—		×
File is deleted, or renamed files exist With the exception of this file, Do you want to run the import?				
Z:#exported_test#zlib#UnitTests#syncsearch_test0.utest Z:#exported_test#zlib#UnitTests#updatewindow_test0.utest Z:#exported_test#zlib#UnitTests#st_data_type_test0.utest Z:#exported_test#zlib#UnitTests#tr_static_init_test0.utest Z:#exported_test#zlib#UnitTests#uncompress_test0.utest Z:#exported_test#zlib#UnitTests#zlibCompileFlags_test0.utest Z:#exported_test#zlib#UnitTests#zlibCompileFlags_test0.utest Z:#exported_test#zlib#UnitTests#zcfree_test0.utest Z:#exported_test#zlib#UnitTests#zcfree_test0.utest Z:#exported_test#zlib#UnitTests#zclibCompileFlags_test0.utest Z:#exported_test#zlib#UnitTests#zcfree_test0.utest Z:#exported_test#zlib#UnitTests#zclibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zclibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zclibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zclibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zclibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zclibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zclibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zclibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zclibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTests#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTest#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTest#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTest#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTest#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTest#zlibVersion_test0.utest Z:#exported_test#zlib#UnitTest#zlibVersion_test0.utes				
L	OK		Cance	I

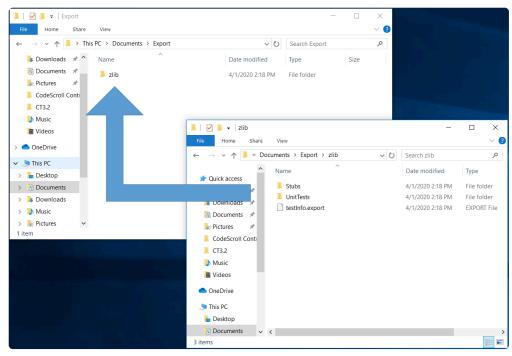
4. If the same test exists in the project, select whether to overwrite the test. Select [Yes] to overwrite and select [No] to cancel the import.

Import tests			_		\times
Import tests					
Import project's te	st informations.				
Unit Test	^t exported_test₩zlib)		<u>S</u> e	arch
e Import					×
There is a te to overwrite	est with the same i ?	name as adler32_c	ombine_test0.ut	est. Do you	ı wish
Yes	Yes To All	No	No To All	Can	cel
Yes	Yes To All	No	No To All	Can	cel
All Stub Option Overwrite exist If the same stu	ing test files witho ib exists already, it ib exists, it is adde	out warning is not imported. d as a new stub.	No To All	Can	cel

5. After the import has completed, it is applied to the selected project.

Import information exported from Controller Tester 2.6.14 or earlier

1. Enter the path to import the test in [Import path].



• If [testinfo.export] file exists in the test path to be imported, select the directory containing the file.

If you import the information exported separately from Controller Tester 2.6.14 or less version, select the higher-level directory containing the information exported.

- 2. In [Unit Test] group, check whether to import [Test] and [Test data] respectively.
- 3. In [Integration Test] group, check whether to import [Test].
 - The test and data must always be in the same directory, and if it is not, the data is not imported.
- 4. In [Stub] group, select the stub to be imported among [Connected Stubs] and [All stubs].
 - For the stubs created in Controller Tester 2.3 version, or the stubs migrated to Controller Tester 2.6, if the same stubs exist in the project, the import cannot be carried out.
 - For the stubs created in Controller Tester 2.6 version, if the same stubs exist, you can select either not to import them optionally or to create as a new stub additionally.
 - There is no feature to link the stubs to the test in Controller Tester 2.3 version, so you need to link directly to the test when migrating to version Controller Tester 2.6 or later.
- 5. In [Option] group, select whether to overwrite the existing test files if they exist and how to handle them when the same stub exists.

18.3.8. Test – Import test using test code file

1. In the Import Wizard, select [Test] -> [Import test using test code file] and click [Next].

import		
Select Create tests from test code files.		Ľ
Select an import wizard:		
type filter text		
 Coverage General Existing Projects into Workspace File System Preferences Test Import test Import test using test code file 		
< <u>B</u> ack <u>N</u> ext > <u>F</u> i	inish	Cancel

2. Click [Add...] and select the test to be imported.

Import test using test code file			×
Import test using test code file			
Create tests using test code files. Please select files.			
Test files:			
Z:WexportWunittestW_tr_align_test0.c Z:WexportWunittestW_tr_flush_block_test0.c Z:WexportWunittestW_tr_stored_block_test0.c Z:WexportWunittestW_tr_tally_test0.c Z:WexportWunittestWadler32_combine_test0.c Z:WexportWunittestWadler32_test0.c Z:WexportWunittestWbi_flush_test0.c Z:WexportWunittestWbi_flush_test0.c Z:WexportWunittestWbi_windup_test0.c Z:WexportWunittestWbi_windup_test0.c Z:WexportWunittestWbi_windup_test0.c Z:WexportWunittestWbi_tree_test0.c Z:WexportWunittestWbild_b_tree_test0.c Z:WexportWunittestWcompress_block_test0.c Z:WexportWunittestWcompress_test0.c Z:WexportWunit	~	<u>R</u> emo	
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish		Cance	I

3. Click [Finish] to apply the selected test to the selected project.

18.3.9. Preferences – Virtual Memory Address

You can import the virtual memory address exported by using the Import function.

1. In the Import Wizard, select [Preferences] > [Virtual Memory Address] and click [Next].

Import	
Select Import Virtual Memory Address	Ľ
Select an import wizard:	
type filter text	
 Coverage General Preferences ToolChain Virtual Memory Address Requirement > Test 	
< Back Next > Finish	Cancel

2. When you select a file(*.csvm) to import, a list of virtual memory addresses included in the selected file appears. Select virtual memory addresses to import from the list of virtual memory addresses.

CT Im	Import virtual memory address							×
Impor	nport virtual memory address							
Select the file to import and the virtual memory address. If a virtual memory address with the same name exists, a number such as '(1)' is added after the setting name.								
Select	files to import: C:\Users	₩jhyeo₩Desktop₩test₩virtualN	1emoryAd	dress.csvm			Sea	rch
List of	imported virtual memory	addresses:						
	Configuration name	Description	:	Status				
	virtual		· ·	-				
	vm (3)			Duplicate name				
	vm (7)			Range overlap				
Selec	t All Deselect All							
Selec	L AII Deselect AII							
Virtual	memory address value ra	nge list:						
Begir	ı		End					
-								
			< Back	Next >	Finish		Cance	1

- Status
 - Duplicate name: If there is a virtual memory address with the same name, the import status will be "Duplicate Name". If you get a virtual memory address that is "Duplicate Name", it will be imported with the displayed setting name.
 - Range overlap: If there are virtual memory addresses in the same range, the import status will be "Range overlap". When you get a virtual memory address that is "Range overlap", the virtual memory address in the same range is imported.
- 3. Click the [Finish]. You can click the link to check the imported virtual memory address on the [Preference] page.



18.3.10. Preferences – ToolChain

You can import the toolchain information exported by using the Import function.

1. In the Import Wizard, select [Preferences] > [ToolChain] and click [Next].

Import	_	
Select Import ToolChain configuration informations.		Ľ
Select an import wizard:		
type filter text		
 > > Coverage > > General > > Preferences > ToolChain > > Requirement > > > Test 		
< Back Next > Finis	h	Cancel

2. If you select the file (*.tch) to be imported, the list of toolchains included in the selected file are displayed. Select the toolchains to be imported from the toolchain list.

t Import			_	D X
Select				
Select the	toolchain and destination locati	ion		
Select the	coordiant and destination locat	01.		
Salact the	import location: Z:\export\exp	ported tob		Browse
		Joned.tcm		DIOWSe
Select the	toolchain to import:			
Default	Toolchain Name	Description	Status in import	
	CPP_TI_TMS320_6000_SoC		Need to configure	
	GCC 4.7 (32bit)		Need to configure	
	GCC 5.3 (32bit)		Need to configure	
	Microsoft Visual Studio 20		Need to configure	
	Microsoft Visual Studio 20		Need to configure	
	Microsoft Visual Studio 20		Need to configure	
	Microsoft Visual Studio 20		Need to configure	
	iar		Need to configure	
Select A	II Deselect All			
Open T	oolchain preference page after w	vizard is finished		
	oolenam preference page arter n			
		< <u>B</u> ack <u>N</u> ext >	<u>F</u> inish	Cancel

3. If the toolchain name is duplicated with the name of the existing toolchain, the import status becomes "Need to modify". You can change the toolchain name to be imported in the [Select how to resolve] window displayed when clicking [Next] button.

Import				×
Resolve conflicts				
The toolchain that have the same name ex Enter the new name.	ist.			
Select how to resolve:				
Toolchain	New Name			
GCC 4.7 (32bit)	GCC 4.7 (32bit)(1)			
GCC 5.3 (32bit)	GCC 5.3 (32bit)(1)			
<u>E</u> xcept				
1				
			_	
<	<u>Back Next > Finish</u>	1	Cance	A Contraction

4. Click [Finish].

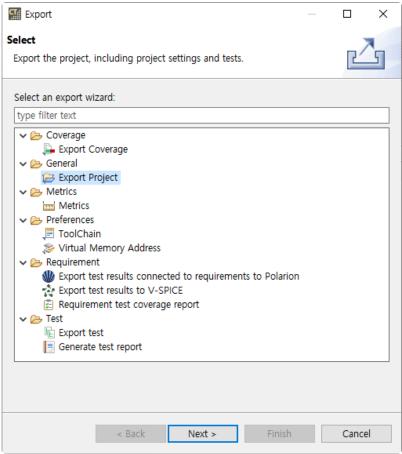
18.4. Export

You can export projects, tests, toolchains, and etc in Controller Tester.

1. Select [File] > [Export] from the main menu or [Export] from the dashboard to open the Export Wizard.

File	Edit Search	Project	Target	Window
	New			>
	Close		C	Ctrl+W
	Close All		Ctrl+Sł	nift+W
	Save			Ctrl+S
ß	Save All		Ctrl+S	hift+S
	Rename			F2
	Refresh			F5
	Switch Workspa	ace		
è	Import			
4	Export			
	Properties		Alt	+Enter
	Exit			

2. Select a item to export and click [Next].



Metrics	Metrics	Export the project's metrics measurement results in a report format.
Requirement	Export test results connected to requirements to Polarion Requirement test coverage report	Export the results of tests associated with requirements to a requirement management tool or to a CSV file.
General	Export project	Export Controller Tester projects.
Coverage	Export Coverage	Export the coverage results in a csd file.
Test	Export test Generate test report	Export tests or generate test reports.
Preferences	<u>ToolChain</u> <u>Virtual Memory Addres</u>	Export virtual memory address or toolchain information.

18.4.1. Metrics – Metrics

It exports the metric result of the project in a report file.

1. In the Export Wizard, click [Metric] -> [Metric] and then click the [Next] button.

Export			
Select Export project's me	tric information to report.		Ż
Select an export wi	zard:		
type filter text			
	< Back Ne	ext > Fir	iish Cancel

2. Select the project and metric type to export metric information and click the [Next] button.

Export Metrics Export project's metric informations to report file. Available Projects: Select All Deselect All
Available Projects:
Select All
Select All
1 of 1 selected.
Metric Type
Function Module
► Module
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Cancel

3. Select the path and report format to export the report and click the [Finish] button.

Export		\times
Export Report		
Choose export destination and extension.		
Use default location Location: C:#Users#x#Export	Brow	'se
File Extension HTML (.html) MS Excel (.xlsx) MS PowerPoint (.pptx) MS Word (.docx) PDF (.pdf)		
Select All Deselect All		
< <u>Back</u> <u>N</u> ext > <u>Finish</u>	Cance	I

18.4.2. Requirement – Export test results connected to requirements to Polarion

You can export test results, which is connected to requirement, to Polarion.

1. In the Export Wizard, select [Requirement] -> [Export test results connected to requirements to Polarion] and click [Next].

Export —		\times
Select		7
Export test results connected to requirements to Polarion.	Ľ	5
Select an export wizard:		
type filter text		
 > Coverage > General > Metrics > Preferences > Requirement Export test results connected to requirements to Polarion Requirement test coverage report > > Test 		
< Back Next > Finish	Cano	el

Enter the information to connect with Polarion server and click [Next]. You can see the entered password to click obutton.

Import re	equirements from Polarion		\times
Import requ	uirements from Polarion		
Enter the in	formation to access the Polarion server.		
Server:	https://almdemo.polarion.com		
Port:	443		
Username:	admin		
Password:	****		Θ
			8
	< Back Next > Finish	Canc	ol
	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cano	ei

- The information of Polarion's demo server is entered in server and port.
- If you use a local server, enter http://ip_address in server and port number of local server in port.
- 3. Enter the information of target to export test result and click [Finish].

Project: TEST(TEST)							~
Document:	Testing/Test Specification						
Гуре:	Test Case						~
ID	Title	Success	Failure	Error	Not Exec	Total	
TEST-101	User name must be validated when a	11	0	8	0	19	
TEST-152	User name must be longer than 3 cha	1	0	0	0	1	
TEST-151	User name must contain at least one i	1	0	0	0	1	
TEST-149	Password must differ from the user na	1	0	0	0	1	
TEST-148	Password must contain at least one ni	1	0	0	0	1	
TEST-147	Password must be validated when the	1	0	0	0	1	
							_

- Project : Select a project to export test results. The project from which the requirements were imported is selected by default.
- Document : Select a document and a content to export the test results. You can export test results without selecting a document.
- Type : Select a type of work item to export the test results.

18.4.3. Requirement – Export test results to V-SPICE

You can export test results to V-SPICE.

1. In the Export Wizard, select [Requirement] -> [Export test result to V-SPICE] and click [Next].

Export			×
Select Export test results to V-SPICE.		Ľ	3
Select an export wizard: type filter text			
 Coverage General Peferences Requirement Export test results connected to requirements to Pol Export test results to V-SPICE Requirement test coverage report Test 	arion		
< Back Next > Finit	sh	Cano	cel

Enter the information to connect with V-SPICEserver and click [Next]. You can see the entered password to click o button.

Import r	equirem	ents from \	/-SPICE					×
Import req	uireme	nts from \	/-SPICE					
Enter the in	formatio	on to acces	s the V-	SPICE serve	er.			
Server:	http://							
Port:	38080							
Username:							 	
Password:								Θ
		< Bac	k	Next >		Finish	Cano	el

3. Enter the information of target to export test result and click [Finish].

roject: WBS	S_TEST						
ID	Title	Success	Failure	Error	Not Exec	Total	
WBS_TEST_	WBS_TEST_REQ_0	1	0	1	0	2	
WBS_TEST_	WBS_TEST_REQ_3	2	0	0	0	2	
WBS_TEST_	WBS_TEST_REQ_6	1	0	1	0	2	
WBS_TEST_	WBS_TEST_REQ_8	1	0	0	0	1	

• Project : Select a project to export test results. The project from which the requirements were imported is selected by default.

If there are no tests connected with a requirement in the selected V-SPICE project, the list of requirements is not displayed. All test information is exported even if the requirements list is not displayed.

18.4.4. Requirement – Requirement test coverage report

You can generate requirement test coverage report in excel format.

1. In the Export Wizard, select [Requirement] -> [Requirement test coverage report] and click [Next].

Export -	_		×
Select Export the requirement test coverage report.		2	5
Select an export wizard: type filter text			
 > Book Coverage > Book General > Preferences > Preferences > Requirement > Export test results connected to requirements to Polarion > Requirement test coverage report > > Test 			
< Back Next > Finish		Cance	I

2. Select the path to export requirement test coverage report and click [Finish].

_				
CT Exp	ort Requirement Test Coverage Report			×
Export	Requirement Test Coverage Report			
	the path to export the requirement test coverage	report		
Select	the path to export the requirement test coverage	report.		
Path:	D:\export		Brow	se
	< Back Next >	Finish	Cance	1

3. You can check the exported report by clicking [Open Folder] .

CT Repo	rt export successed		×
1	The report is generated.		
		Open Folder	Confirm

18.4.5. General – Export project

The project can be exported including the project settings and the test. The exported project can be imported with [Import Project].

1. In the Export Wizard, select [General] > [Export Project] and click [Next].

Export	_	
Select Export the project, including project settings and tests.		Ż
Select an export wizard: type filter text		
< <u>B</u> ack <u>N</u> ext > <u>F</u> inis	h	Cancel

2. Select the project to be exported and the export path and click [Finish]. When you export a project, you can export it including the toolchain and source files.

eff Export Project	 		×
Export Project			
Export the project, including project settings and tests.			
Project:			
☑ ≌study_sample		Select /	
1 of 1 selected. Options Include Toolchain Include Source File			
Path: C:\Users\hikim\Desktop		Searc	ch
< Back Next > Finish		Cancel	I

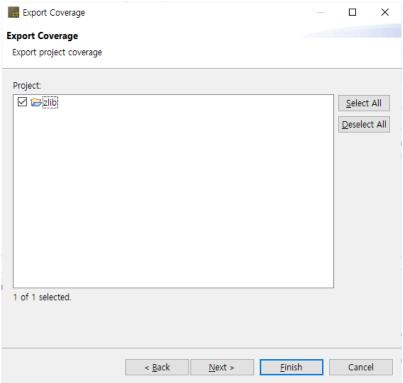
18.4.6. Coverage – Export Coverage

You can export the coverage information measured in the project as a file in Coverage Shared Data (*.csd) format. The exported coverage can be imported with [Import Coverage].

1. In the Export Wizard, select [Coverage] -> [Export Coverage] and click [Next].

Export -		×
Select Export the project's coverage information as a Coverage Shared Data File Format(*.csd).	Ľ	5
Select an export wizard:		
type filter text		
 ✓ ➢ Coverage ➢ Export Coverage > ➢ General > ➢ Metrics > ➢ Preferences > ➢ Test 		
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cance	1

2. Select the project that you want to export coverage information and click [Next].



3. Enter the directory path to export and click [Finish].

Export Coverage	e			_		×
Export Coverage						
Select a directory information(*.csd)	to export the file .	for sharing covera	age			
Directory path:	Z:\export				Brows	se)
						1
	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish		Cancel	

4. A file with the same name as the project name is created.

Z:\\export\		
Name	Size	Modified
[zlib.csd]	4 606	2020-03-31 16:29

18.4.7. Test – Export test

The test information created in the project can be exported at once. Before a project is not analyzed or if there is no test information (Unit/Integration Test, Stub) created, the export cannot be carried out.

1. Select [Test] > [Export Test] in the Export Wizard and click [Next].

Export				
Select Export test information				ß
Select an export wizard:				
type filter text				
 > Coverage > General > Preferences > Test Create Test Report Export test 	t			
< <u>B</u> a	:k <u>N</u> ext	> <u>F</u> inish	1	Cancel

2. Select the project that tests are exported and click [Next].

Export tests			×
Export tests			
Export project's test informations.			
Project:			
🗹 😂 test		Select	All
		Deselect	t All
1 of 1 selected.			
For Faceced.			
< Back Next >	Finish	Cance	1
< bdck Next >	FINISH	Cance	1

3. Select the export path, the unit test items, the integration test items, the stub items, the fault injection and the options, and click [Finish].

Export tests		\times
Export tests		
Export project's test informations.		
Export path: C:#Users#hikim#Desktop#CT교육#01.13	Sear	ch
Unit Test		
☑ Test		
Test Data		
Integration Test		
Test		
Stub		
Connected Stub		
O All Stub		
Includes		
Fault Injection		
Option		
Overwrite existing test files without warning		
Export only checked tests in Unit/Integration Test View.		
< Back Next > Finish	Cance	el 🛛

- a. In [Export path], enter the path to export the test.
- b. In [Unit Test] group, check whether to export [Test] and [Test Data] respectively.

- c. In [Integration Test] group, check whether to export [Test].
 - When exporting the integration test, it is exported with the test data in a set.
- d. In [Stub] group, select the stub to be exported among [Connected stubs] and [All stubs].
 - **Connected stubs**: The stubs connected with the test to be exported. If the stubs have no link with the test or the test to be exported is not selected, it does not export that stubs.
 - All Stub: It exports all the stubs created in the project.
- e. In [Includes] group, select whether to export with fault injections.
- f. In [Option] group, choose whether to overwrite the existing test file when they exist, and whether to export only the tests checked in the Unit/Integration Test view.
- 4. Click [Open Folder] button in the message "Test has been exported successfully." to check the test exported in the folder where it is saved, or click [Confirm] to finish the export.

			-
ct Expo	ort tests		>
j	The test has been exported successfully. Open the path exported?		
		Open Folder	Confirm

From Controller Tester 2.6.14 or later version, when exporting tests, [testinfo.export] file is created in the project directory exported.
 (testinfo.export: the file that records the exported information)

18.4.8. Test – Generate test report

You can create the test result report in various formats by using Export.

1. In the Export Wizard, click [Test] -> [Create Test Report] and click [Next].

Export	_	
Select		
Generate test report.		
<u>S</u> elect an export wizard:		
type filter text		
 > Coverage > General > Preferences > Test Create Test Report Export test 		
< <u>B</u> ack <u>N</u> ext > <u>F</u> inis	h	Cancel

2. Select the project that you want to generate the report and the report type.

Export Export	_		×
Export Test Result			
Select Project and Option			
Project:			
☑ test		Select	
1 of 1 selected. Option File coverage report (format: XLS) Report for each test (format: XLS) Test stub report (format: XLS) Report with external coverage Uncovered function report (format: XLS)]	
< Back Next > Finish		Canc	el

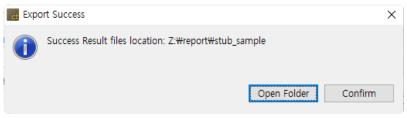
- a. In [Project], select the project that you want to generate the report
- b. In [Option], check each report to be exported.

Report type	Description
File coverage report	Generate reports of covered functions for each files in the selected project.
Report for each test	Generate reports of test result and coverage for each test in the selected project.
Test stub report	Generate a report of stubs created in the selected project.
Report with external coverage	Generates a report containing the external coverage in the selected project.
Uncovered function report	Generate a report of the uncovered function in the selected project.

3. Select the location where the reports are exported and the report file extensions and click [Finish].

Export	_		×
Export Test Result			
Select file type to be exported.			
Use default location			
		Descus	
Location: D:\export		Brows	e
File Extension			
HTML (.html)			
MS Excel (.xlsx)			
MS PowerPoint (.pptx)			
MS Word (.docx)			
PDF (.pdf)			
Select All Deselect All			
< Back Next > Finish		Cance	
C DACK INEXC > PILISH		Cance	

- a. Selecting the checkbox [Use default location] specified "User Account\Export" directory by default.
- b. [File Extension] provides the report file format with html, pdf, docx, pptx and xlsx. The report file extension can be selected in duplicate.
- 1. Click [Open Folder] button in the message "Report has been created" to check the report in the folder where it is saved.



18.4.9. Preferences – Virtual Memory Address

You can export the information of the virtual memory address.

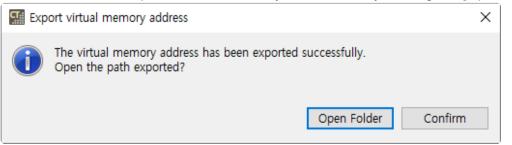
1. In the Export Wizard, click [Preferences] > [Virtual Memory Address] and click the [Next].

Export		×
Select Export Virtual Memory Address	Ľ	5
Select an export wizard:		
type filter text		
 > Coverage > General > Preferences > ToolChain > Virtual Memory Address > Requirement > > Test 		
< Back Next > Finish	Cance	el

2. Select the path to export the virtual memory address and click the [Finish].

Export × Export virtual memory address Select the path to export the virtual memory address. File path: Search Search Search Virtual memory address Search Search						
Select the path to export the virtual memory address. File path: Search	Export			_) X
File path: Search	Export virtual me	mory address				
	Select the path to	export the virtual	memory address.			
	File path:				[Search
< Back Next > Finish Cancel					(
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back Next > Finish Cancel						
< Back IVEXU> PITIST Cancel		- Pack	North	Finish		ancel
		< Back	Next >	FINISH		ancei

3. You can check the exported virtual memory address file by clicking the [Open Folder].



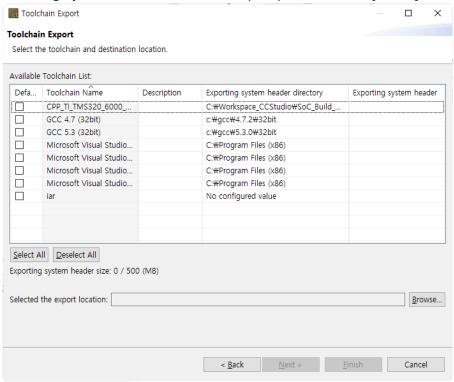
18.4.10. Preferences – ToolChain

You can export the information of the toolchain created.

1. In the Export Wizard, click [Preferences] > [ToolChain] and click [Next].

Export	_	
Select Export ToolChain configuration informations.		ß
Select an export wizard:		
type filter text		
 > Coverage > General > Metrics > Preferences ToolChain > Requirement > > Test 		
< Back Next > Finit	sh	Cancel

2. The list of toolchains registered currently is displayed. Check [Exporting system header] to export including system header. Enter the export path and click [Finish].



19. Edit Menu

In the Edit menu, you can carry out the functions such as Cut, Copy, Paste, Delete and Find/Replace the items selected in the view or editor, or undo or redo the last action performed.

Edit	Search	Project	Target	Window	Help
Y	Undo			Ct	rl+Z
\mathfrak{G}	Redo			Ct	rl+Y
of	Cut			Ct	rl+X
Ð	Сору			Cti	rl+C
Ē	Paste			Cti	rl+V
×	Delete			De	elete
	Select All			Ct	rl+A
	Find/Rep	ace		Ct	rl+F

The menu may vary depending on the selected view or editor.

20. Search Menu

Search Analysis Result

You can search the result created after analysis.

a Search	$ \Box$ \times
🔗 Analysis Result Search 🔗 File Search	
Search string (* = any string, ? = any character):	✓ □ Case sensitive
Search For Global variable Function Any Element	
Scope O Workspace Selected resource	
Customi <u>z</u> e	Search Cancel

Search word

Enter characters to be searched.

The available wildcards are displayed in the Search dialog box.

- "*" Wildcard string: matches the set of characters containing the empty string.
- "?" Wildcard character: matches all characters.

Search target

Select the target to be searched. The global variable, function or all targets can be searched.

Search range

Select the searching range. It can be searched within the workspace or the selected item range.

File search

A file can be searched.

Search –	×
Analysis Result Search 📴 File Search	
Containing text:	
✓ □ Case sensitive	
(* = any string, ? = any character, # = escape for literals: * ? #) Regular express	sion
File name <u>patterns</u> (separated by comma):	
* ~ Ch <u>o</u> ose	
(* = any string, ? = any character, !x = excluding x)	
Derived resources	
Customi <u>z</u> e <u>S</u> earch Cano	el

Text included

Enter a character to be searched. To search a file, the field must be empty. Click [▼] to select the characters searched recently.

The available wildcards are displayed in the Search dialog box.

- "*" Wildcard string: matches the set of characters containing the blank string.
- "?" Wildcard character: matches all characters.
- To search "*", "?" or "\" character, enter a backslash before the characters in order to indicate that you do not use these characters by "\ *" wildcards. (Ex: "\?" or "\ \")

File Name Pattern

Enter the pattern of all file names for the files to be searched by using the specific expression. The wildcards that can use the file name pattern are displayed in the Search dialog box.

- "*" Wildcard string: matches the set of characters containing the blank string.
- "?" Wildcard character: matches all characters.

21. Project Menu

In the Project menu, you can execute the operations for the project (open, close, initialize).

	Proje	ect	Target	Window	Help
			en Projec ose Projec		
	T	Init	ialize Pro	oject	
			nerate co llect proj		ne interface INI file
1		Pro	perties		

Open Project

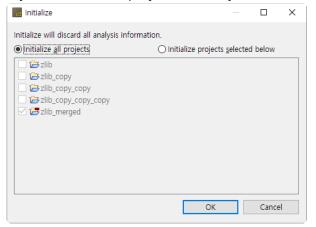
You can open a closed project selected in the Test Navigator view.

Close Project

You can close an open project selected in the Test Navigator view.

Initialize Project

You can initialize all the open projects in the workspace or the selected projects in the project list below. If you initialize the project, all analysis results are cleared.



Generate Command Line Interface INI File

Based on the project information selected in the Test Navigator, you can create the configuration file needed to execute the Controller Tester in CLI environment.

Collect Project Log

You can export the log file of the project selected in the Test Navigator view into the specified path.

Suresofttech

Properties

You can see the information or change the settings for the project selected in the Test Navigator view.

22. Window Menu

In the Window menu, you can open the same new editor as an active editor, open a perspective or a new view, and see the preferences.

Window New		Help	
	New		
	Ope	n Perspective	>
	Show View		>
	Save Perspective As		
	Rese	t Perspective	
5	Close Perspective		6
	Prefe	erences	

New Editor

It copies the selected source code editor or test editor.

Open Perspective

You can select and open the perspective registered in Controller Tester.

Show View

You can select and open the view registered in Controller Tester.

Save Perspective As

Saves the perspective reconfigured by the user as the other name.

Reset Perspective

Resets the perspective into the initial state.

Close Perspective

Closes the perspective currently open.

Preferences

You can check or change the settings currently applied to the tool. For more information, please refer to <u>Preferences</u>

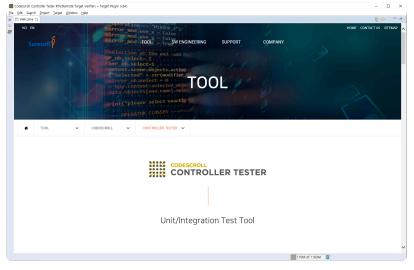
23. Help Menu

In the Help menu, you can find manuals, troubleshooting guides, tutorials, license settings, changing and reverting analysis settings, and information about the installed product.



Welcome

You can see the welcome page that appears when the tool is executed for the first time.



Show Manual

You can see the Controller Tester User Manual.

CONTROLLER TESTER USER Manu	Starting Starting Starting Starti	English		
Before Starting	Befo	re Starting		
Overview	Purpo	se of Document		
Install	This docum	ment provides the information o	on how to use the CodeScroll Control	er Tester Product.
Uninstall				Overview >
Run				•
Set License				
 Set a Toolchain (Analyzer) 				
Auto-registration of Toolchain				
Add a Toolchain				
Edit a Toolchain		Copyright © 2021 Su	iresofttech — Powered by 🔟 Mariula	
Import a Toolchain				
✓ Create a Project				
C/C++ Project with Source Files				
C/C++ Project from Embedded				
C/C++ Project from Visual Studio				

Show Troubleshooting Guides

1

You can see the Con	troller Tester Tr	ouble	eshooting	Guides.
CONTROLLER TESTER Controller Tester	er Troubleshooting Guides	3.6	Search	English
Controller Tester Troubleshooting Guide How to assign values to flexible array members How to increase heap memory in JAVA when memory is low Project DB file ("csp) is damaged due to abornal termination Testrum exe crashes when testing QNX software When a specific function is not displayed in the coverage view if the error message is not displayed after the project analysis fails The screen is cut off when the	er Troubleshooting Guides Controller Tester This is a guide to solving common pro- How to assign values to flexible How to increase heap memory Project DB file ("cs) is damag Testrun exe crashes when test When a specific function is not if the error message is not disp The screen is cut off when the Windows hit flen of tound if INFO (ut hio); runTest.testrun executed When "Extract Toolchain Info" f	Troub oblems you n array memb in JAVA when ed due to ab ing QNX softw displayed in layed after tt Windows disp exit code(10	Ieshooting hay encounter using Cor- ers memory is low normal termination hare the coverage view he project analysis fails lay magnification is not	Guide antroller Tester.
Windows display magnification is not 100% Windows h file not found If INFO [ut hio]: runTest.testrun exit code(105) is displayed in the log and the unit test is not executed When "Extract Toolchain Info" fails Messages display abnormally in the error view	 Messages display abnormally in When testing with source code Errors that may occur during te When cannot find global valuet "C2118 : negative subscript" er SDK version issue when using Issues during installation of Co When cannot save a large cont When workspace is broken 	n the error vi containing C est execution a after import ror occurs du Visual Studio ntroller Teste	++20 items ing integration test uring test execution 2015 toolchain r	

Cheat Sheets

You can run cheat sheets on how to use Controller Tester.

Cheat Sheet Selection	×
Select the cheat sheet to open: Select a cheat sheet from the list:	
Controller Tester Basic Usage	
○ Select a cheat sheet from a file: Browse	
O Enter the URL of a cheat sheet:	~
OK Cancel	

Modify Parser Config

You can modify the parser config by entering the compressed file provided by SureSoft Technologies, Inc.

How to use

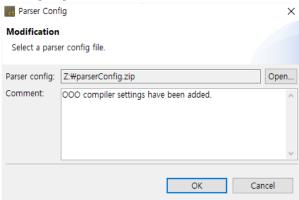
1. Select [Help] > [Parser Config] > [Modification...] menu.

Help	0			
	Welcome		1	C-
?	Help			New Proje
	License			
	Parser Config	>	-8	Modification
ct	About		ŋ	Revert

2. Click [Open...] to select the compressed analysis setting file and enter the changes of analysis settings to [Comment] text as an option.

Parser Config	×
Modification Select a parser config file.	
Parser config:	Open
Comment:	~ ~
	OK Cancel

3. Click [OK].



Format of the parser config file

- The directory [parserConfig] must exist immediately when opening the compressed file as shown in the figure below.
- The extension is "zip".

lame	Size	Packed Size
parserConfig	241 261	40 758

Revert the Parser Config

You can revert the parser config by selecting one among the previous parser configs.

How to use

1. Select [Help] > [Parser Config] > [Revert] menu.

Help	>			
	Welcome		1	C.
0	Help			New Proje
	License			
	Parser Config	>	-8	Modification
ct	About		ஒ	Revert

2. Check the parser config to be reverted in the list of the existing parser configs and select [OK].

Par	rser Config		>
ever	t		
Press	'Ok' button.		
	Date	Comment	
		initial	
	2020-03-31 10:38:15	xxx Compiler analysis settings updated.	
XX Co	ompiler analysis settings up	odated.	1
			•
		OK Cancel	
		c cancer	

Information

You can see the information and version of the product installed.



Click [Installation Details] to see the features installed and plug-in.

24. Troubleshooting

If it conflicts with the security program at installation

Some security programs may stop running the tool installation package. If an error occurred, temporarily stop the security program operation and proceed with the installation again.

If the product does not operate after specifying the workspace when executing it

When executing the product for the first time, the global data of CodeScroll is stored in \APPDATA\ path. If access to this path is restricted, the product may not be operated.

To solve this problem, please use the feature to set the global data path.

Setting method: Enter the global data path to be set instead of –g default (default value) in CodeScroll.ini file. (the same if a case of -g or –global and csc.ini file)

If the -g option is not given, the dialog for setting any global data path when executing the product for the first time is displayed. If canceled here, it is set to the default path.

If it is installed and run in Windows Vista/7

It can only be run with the User Access Control (UAC) features disabled.

If you want to use it without disabling UAC, change the directory during installation so that it is not installed under the Program Files.

In addition, the Windows accounts for installation and execution must be the same and must be installed with an administrator account.

If an analysis is failed or Compile (pre-processing) error occurred

It is the state of source code that the test cannot be run. Please check the compile/link flag and header file/library settings.

You can check the settings in [Preferences] -> [Toolchain settings].

If all settings are correct (if it is possible to build without any problems in the actual development environment (IDE etc.)), please contact SureSoft Technologies, Inc.

If it fails to run all test cases

This is because the test engine operation is blocked by the firewall. The test engine loaded in memory when executing the test must be added as an exception to the firewall setting to run normally.

If it fails to register toolchain automatically

If it fails to register toolchain automatically due to some problem with toolchain information installed on

PC, the toolchain auto-registration feature can be turned off. (--disableAutoToolchain option)

• Add --disableAutoToolchain option to CodeScroll.ini file

Preprocessing error when running tests after connecting stub

When you create and connect a stub and run the tests, the stub may display the error catastrophicer ror:cannot open source file.

This error occurs because the path of the stub file is too long. Windows defines the maximum length of the path as 260, so please rename the workspace and the project to reduce the path.

Etc.

For other issues, please refer to the Troubleshooting Guide .

Technical support

If you find any problems, please contact the Technical support contact below.

- help@suresofttech.com
- +82-2-6472-2800

25. CONTROLLER TESTER Target Plug-in

Introduction

Controller Tester Target Plug-in complements the functionality of the default Controller Tester. Controller Tester allows you to automate testing in your own host environment. In other words, you run the test in a software development environment (normal PC). Controller Tester Target Plug-in allows you to run tests of Controller Tester in a real embedded target environment. This makes it easy to check whether the test results are the same in the host and target environment and to run tests that cannot be executed in the host environment because of factors dependent on the target environment.

25.1. Target Environment Settings

To run tests in a target environment using Controller Tester Target Plug-in, you need to enter information about the target environment.

Controller Tester Target Plug-in builds a test harness using information about the target environment entered by the user and automatically gets the results of running in the target environment.

The target environment can be set in the project properties page or in the new target test project wizard.

Target environment settings

Target environment settings are divided into basic information and detailed settings.

```
▶ GNU Compilers ▶ gcc ▶ 5.3 ▶ others ▶ nodebugger
```

After entering the basic information, detailed settings that require input are displayed.

Analysis	Name	Value
Build	Toolchain Name	Microsoft Visual Studio 2010 (32bit)
Run	Status	This toolchain is supported.
etc.	C Compiler	C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\bin\cleak
ett.	System Header(C Compiler)	C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\ATLMFC\I
	Library(C Compiler)	C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\ATLMFC\L
	C++ Compiler	C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\bin\classed
	System Header(C++ Compiler)	C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\ATLMFC\I
	Library(C++ Compiler)	C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\ATLMFC\L.
	Description:	

Target environment detailed settings

Target environment detailed settings are divided into the analysis, build, run, and etc.

Category	Description
Analysis	The toolchain information is displayed, and in the case of a target test project, the target compiler related settings required for analysis are displayed.
Build	The settings for building the test software are displayed.
Run	Settings for running tests and getting results in the target environment are displayed.
etc.	Other settings are displayed. (Program entry point, etc.)

Required settings for each category are indicated in red. Depending on whether the required settings for each category are entered or not, the behavior when clicking the [Run] button is different as shown

Required settings completed category	Description
None	Test run impossible
Analysis	If you click [Run], it overwrites the original source code with the test code. To perform the test, the user must manually build and run the test.
Analysis, Build	If you click [Run], it builds the test code. To perform the test, the user has to manually run the test on the target.
Analysis, Build, Run	Automatically run tests in the target environment.

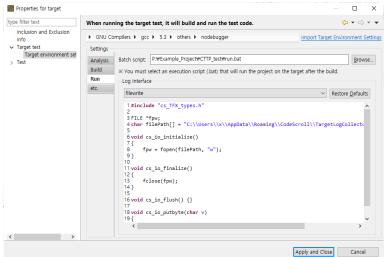
Import target test results

Controller Tester Target Plug-in saves and imports the result of running the test in the target environment in log format.

If you do not use the debugger, you need the settings for saving the target test results(Log interface) and the settings for importing(Target Log Collector – Preferences).

Log interface

Log interface is a setting for saving the test results in the target environment. Log interface is written in the form of a source code that is actually run in the target environment.



Log interface structure

Function	Description
<pre>viod cs_io_initialize()</pre>	Initial function for transfer
<pre>void_cs_io_finalize()</pre>	Transfer end function
<pre>void cs_io_flush()</pre>	Remaining data transfer function
<pre>void cs_io_putbyte(codescroll_byte v)</pre>	1-byte data transfer function

Import target log automatically

The target log, which is the result of the test execution, can be automatically imported through the target log collector. For the target log collector settings, please refer to [Target Log Collector] in <u>Target Log</u> <u>Collector</u> and <u>Target Test Preferences</u>.

Import target log manually

If the target log cannot be imported automatically, you can import the target log manually.

Import from a log file

1. After selecting an analyzed project, click [Target] -> [import Target Test Log] -> [Import from Log File...] in the global menu.

Target Window Help			
👔 Run Target Test			T,
. 😳 Import Target Test Log	>	Import from Log File	ľ
		Import from Target Log Archive	
		Import from Target Log Collector	

2. Select a target test log file and click the [OK] button.

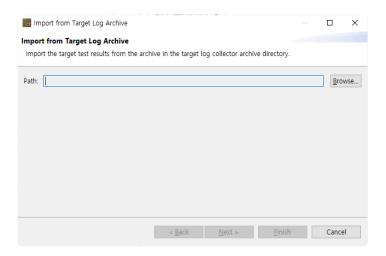
Import a target test log		×
Import a target test log Select the log file of target test.		
Path:		 <u>B</u> rowse
	OK	Cancel

Import from a target log archive

1. After selecting an analyzed project, click [Target] -> [import Target Test Log] -> [Import from Target Log Archive...] in the global menu.

Targ	et Window Help			
٢	Run Target Test			T,
0	Import Target Test Log	>	Import from Log File	ł
_			Import from Target Log Archive	
			Import from Target Log Collector	

2. Select a target log archive file and click the [Next] button.

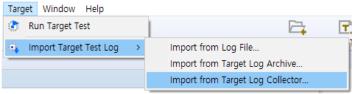


3. Check a target test log file and click the [Finish] button.

elect	target test logs:	Show exclude	ded logs 🗹 Show applied log
	Project	Test code generation time	Status
0	target	2020-04-01 13:57:26	Applied
	target	2020-04-01 13:52:13	Applied
	target	2020-04-01 13:50:45	Applied
	target	2020-03-31 17:43:35	Applied
	target	2020-03-31 17:41:48	Applied
You	a can import the target test logs	s that are neither applied(√) nor excluded(X) o	on next time.

Import from target log collector

- To get logs from the target log collector, you need to select 'Use the default target log collector' from [Preferences] -> [Target Test] -> [Target Log Collector].
- 2. After selecting an analyzed project, click [Target] -> [import Target Test Log] -> [Import from Target Log Collector...] in the global menu.



3. Click the [Yes] button to import all the new target test logs or click the [Show details] button.

	in this get bog	~
1	Do you want to apply the target test logs to project 'target'?	
	Show details Yes	<u>N</u> o

4. If you click the [Show details] button, check the target test log to be imported and click the [OK] button.

ect target test logs:	Show excluded logs Show applie	d l
Project	Test code generation time Status	
target	2020-04-01 13:52:13 Applied	
target	2020-04-01 13:50:45 Applied	
target	2020-03-31 17:43:35 Applied	
target	2020-03-31 17:41:48 Applied	

Result

After the import is completed, you can check the target coverage information in the coverage view.

	rage 🔀					<u>-</u>
irget co	verage information of 'target' project					
	Target Function	Statement	Branch	MC/DC	Function Call	Function
		•	-	•	•	
1	add(double, double)	— 100.00%	— 100.00%	— 100.00%	— 100.00%	Υ
2	divide(double, double)	— 100.00%	— 100.00%	— 100.00%	— 100.00%	Υ
3	longNameFunctionTestlongNameFun	— 100.00%			— 100.00%	Υ
4	longNameFunctionTestlongNameFun	— 100.00%			— 100.00%	Υ
5	longNameFunctionTestlongNameFun	— 100.00%	□ N/A	III N/A	— 100.00%	Υ
6	main()	== 45.45% (= 20.00% (N/A	== 55.55% (Υ
7	multiply(double, double)	= 60.00% (= 50.00% (— 0.00% (0	= 50.00% (Υ Υ
8	subtract(double, double)	— 100.00%	□ N/A	- N/A	— 100.00%	Υ
Total		67.34% (53.84% (50.00% (73.91% (N/A

[Import Target Test Log] may fail because the target test run information and the host test information are not valid if the host test case, source file, function information, etc. are changed before importing the log of the target test result.

25.2. C/C++ Target Test Project

The C / C ++ target test project (hereinafter referred to as the target test project) is a project for target environment testing. Target testing projects do not support host testing.

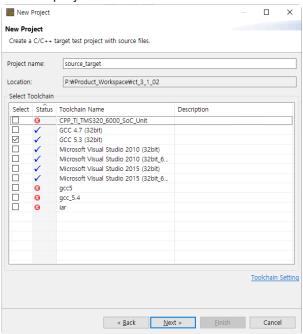
Create a target test project

C/C++ Target Test Project with Source Files

1. Create a project by selecting [File]-> [New]-> [Other ...] in the global menu and clicking [C / C ++ Target Test Project with Source Files].

ct New			×
Select a wizard Create C/C++ target test project with source files.			\$
<u>W</u> izards:			
type filter text			
> C/C++ RTV Project			
 ✓ C/C++ Target Project C/C++ Target Test Project from Embedded(Cod C/C++ Target Test Project with Source Files Create a C/C++ target test project with Build Int > Other 	en Hil	lls, NEG	C)
C/C++ Target Test Project from Embedded(Cod C C/C++ Target Test Project with Source Files Create a C/C++ target test project with Build Im	en Hil	lis, neo	=)

2. Enter a project name and select a toolchain to use.



3. After selecting the basic information of the target environment, enter the detailed settings for each category and click the [Next] button.

-	ironment Settings	
	II out the target environment inform	
GNU Co ettings	mpilers ► gcc ► 5.3 ► others I	please select a debugger Import Target Environment Set
nalysis	Name	Value
	Toolchain Name	GCC 5.3 (32bit)
un	Status	This toolchain is supported.
tc.	C Compiler	C:\u00cefain is supported.
	System Header(C Compiler)	c:\gcc\s.3.0\setuplestimingvcc\setuplestimingv32\starterstarte
	Library(C Compiler)	
	C++ Compiler	C:#gcc#5.3.0#32bit#bin#g++.exe
	System Header(C++ Compiler)	c:#gcc#5.3.0#32bit#lib#gcc#mingw32#5.3.0#include#c++;c:#gcc#
	Library(C++ Compiler)	
	Name:	

4. Select the source files under test and click the [Finish] button.

pp Directory: P:\Example_Project	#c#inain#200-	n. Eile	Brow
	ted by ; or ne	type filter text g adler32.c c compress.c c crs32.c d effate.c g gzlose.c g gzlib.c g gzwrte.c infback.c inflate.c inflate.c inflate.c	Brows Remo

C/C++ Target Test Project from Embedded(CodeWarrior, Green Hills, NEC)

1. Create a project by selecting [File]-> [New]-> [Other ...] in the global menu and clicking [C/C++ Target Test Project from Embedded(CodeWarrior, Green Hills, NEC)].

New		\times
Select a wizard Create a new target test project from CodeWarrior(*.xml), Green Hills MULT (*.gp)) or NEC PM+(*.prw, *.prj).	rı 🦲	2
<u>W</u> izards:		
type filter text		
 > > C/C++ Project > > C/C++ RTV Project > > C/C++ Target Project C/C++ Target Test Project from Embedded(CodeWarrior, Green C/C++ Target Test Project with Source Files Create a C/C++ target test project with Build Information > > Other 	Hills, NE	C)
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Canc	el

2. Enter a project name and select a toolchain to use. After selecting the toolchain, import the embedded project file to test. Currently, embedded projects supported by Controller Tester are CodeWarrior, Green Hills, and NEC.

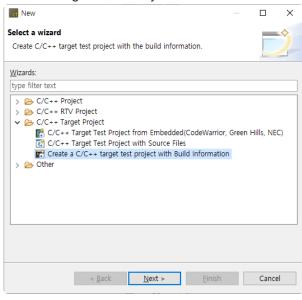
New P	roject)
ew Proj e Create a l		test project from CodeWarrior(*.xml), Green Hi	lls MULTI(*.gpj) or NEC PM+(*.prw,	*.prj).		
roject na	me:	Embedded_Target_Build				
ocation:		P:#Product Workspace#ct 3_1_02				-
elect To	olchain					_
Select	Status	Toolchain Name	Description			-
	()	CPP_TI_TMS320_6000_SoC_Unit	Description			
	√	GCC 4.7 (32bit)				
M	· .	GCC 5.3 (32bit)				
Π		Microsoft Visual Studio 2010 (32bit)				
	1	Microsoft Visual Studio 2010 (32bit 64bit)				
	1	Microsoft Visual Studio 2015 (32bit)				
	1	Microsoft Visual Studio 2015 (32bit_64bit)				
	0	gcc5				
	0	gcc_5.4				
	0	iar				
				т	oolchain !	s
mport Se	etting					
mbedde	d Project F	ile:			Brow	vs
					-	
		< <u>B</u> a	ick <u>N</u> ext > <u>F</u> ini:		Cance	I.

3. After selecting the basic information of the target environment, enter the detailed settings for each category.

ettings	mpilers ▶ gcc ▶ 5.3 ▶ others ▶	please select a debugger Import Target Environment Setti
nalysis	Name	Value
un	Toolchain Name	GCC 5.3 (32bit)
tc	Status	This toolchain is supported.
ц.	C Compiler	C:\u00c8qcc\u00e45.3.0\u00fc32bit\u00fcbin\u00c4qcc.exe
	System Header(C Compiler)	c:#gcc#5.3.0#32bit#lib#gcc#mingw32#5.3.0#include;c:#gcc#5.3.0
	Library(C Compiler)	
	C++ Compiler	C:\gcc\\$.3.0\32bit\bin\g++.exe
	System Header(C++ Compiler)	c:#gcc#5.3.0#32bit#lib#gcc#mingw32#5.3.0#include#c++;c:#gcc#
	Library(C++ Compiler)	
	Library(C++ Compiler)	
		с. Г
	Name:	
	Name:	
	Name:	
	Name:	
	Name:	
	Name:	
	Name:	

Create a C/C++ Target Test Project with Build Information

 Create a project by selecting [File]-> [New]-> [Other ...] in the global menu and clicking [Create a C/C++ Target Test Project with Build Information].



2. Enter a project name and select a toolchain to use. After selecting the toolchain, Specify the path of the created Makefile and build script as the build directory and write the build command. You can choose a script file to set the build-environment from the [Advanced] menu.

oject na	ame:	Target_Build		
ocation:		P:\Product_Workspace\ct_3_1_02		
elect To	olchain			
Select	Status	Toolchain Name	Description	
m	8	CPP_TI_TMS320_6000_SoC_Unit		
	✓	GCC 4.7 (32bit)		
	1	GCC 5.3 (32bit)		
	1	Microsoft Visual Studio 2010 (32bit)		
	1	Microsoft Visual Studio 2010 (32bit_64bit)		
	1	Microsoft Visual Studio 2015 (32bit)		
	1	Microsoft Visual Studio 2015 (32bit_64bit)		
	0	gcc5		
	0	gcc_5.4		
	0	lar		
mport S uild Coi make cl	mmand:		Tools	hain Si
uild Dir	ectory:			<u>B</u> rows
		ct name is determined by combining the entered p lle name extracted from the build information. (e.g		

3. After selecting the basic information of the target environment, enter the detailed settings for each category.

New Pro	oject					×
Target Env	ironment Settings					
-	Il out the target environment infor	mation. (% Red: a required field)				
	mpilers ► gcc ► 5.3 ► others	please select a debugger	Import Tarq	et Enviro	onment !	Setting
Settings						_
Analysis	Name	Value				
Run	Toolchain Name	GCC 5.3 (32bit)				_
etc.	Status	This toolchain is supported.				_
	C Compiler	C:\gcc\5.3.0\32bit\bin\gcc.exe				_
	System Header(C Compiler)	c:#gcc#5.3.0#32bit#lib#gcc#ming	w32₩5.3.0₩inclu	ide;c:₩g	cc#5.3.0)
	Library(C Compiler) C++ Compiler	C:#acc#5.3.0#32bit#bin#a++.exe				
	System Header(C++ Compiler)	c:#gcc#5.3.0#32bit#lib#gcc#ming	w22m5 2.0minclu	dotte		
	Library(C++ Compiler)	c.wgccw5.5.0w52bitwibwgccwining	w52w5.5.0winete	iden c++	r,c.ngcc	n
	Library (C++ Compiler)					
	Name:					
	Description:					
		< Back Next >	Finish		Cance	el
		<u></u>	-			

Toolchains with invalid compiler paths cannot be selected when creating a target test project.

25.3. Target Test Run Settings

You can check the target test execution settings on the target test page in [Project]-> [Properties].

Coverage kind

You can select the type of coverage(Statement, Branch, MC/DC) to be measured for the target test.

Target for coverage measurement

In addition to the functions under test, you can select functions to measure coverage.

All functions called by the target function	Measures coverages for all functions called by the function under test.
The functions called by the previous test run	Includes functions whose coverage was measured from the previous test.

Testcase run

You can choose how to run the test case.

Run at once	Loads and runs all test cases on target at once. Each test case is affected by the previous test case run results.
Run one by one	Repeats loading and running each test case on target. It takes longer than running at once but uses less memory.

25.4. Target Log Collector

The target log collector is a server that collects target test results and sends them to Controller Tester. Collecting target test results methods are largely divided into communication and file scanning. Supported communication protocols and scannable file formats are as follows.

Protocol	TCP, UDP, <u>UART</u>
File format	Target log(text), memory dump(Hex, Intel HEX)

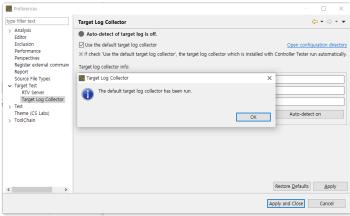
Installation

The target log collector is automatically installed when Controller Tester is installed.

Run

When [Use the default target log collector] is checked in [Preferences] -> [Target Test] -> [Target Log Collector], the target log collector starts collecting target test results.

If you use TargetLogCollector.zip after extracting it, run TargetLogCollector.exe in the unzipped directory at the Windows command prompt.



Settings

It can be set through the 'settings.ini' file in the target log collector installation path, and the options are as follows.

[LogReceiveServer]	Settings for receiving target test results
port	TCP, UDP communication port
protocol	Communication protocol for receiving target test results
timeout	Connection timeout
lastString	String representing the end of the data
serialPort	Serial communication port

port	Target log file transfer port
[LogSendServer]	Settings for communication with Controller Tester
fileExtension	File extensions to scan(Scan all files if blank)
dir	Directory to scan
[ScanLog]	Settings for file scanning
flowControl	Serial communication settings
parity	Serial communication settings
stopBits	Serial communication settings
dataBits	Serial communication settings
baudRate	Serial communication settings
	(Windows: COM#, Linux: /dev/ttyS#)

timeout: If there is no data transmission for the specified number of seconds from the last reception, the received data is saved.
 dir: When a target log file (text, hex) is entered in the specified directory, the target log collector recognizes it.

25.5. Preferences

Target Test

You can set whether to open the target test step notification dialog when the target test is run.

Preferences		_		×
type filter text	Target Test	<		• •
 Analysis Editor Exclusion Performance Perspectives Register external commane Report Source File Types Target Test RTV Server Target Log Collector Test Theme (CS Labs) ToolChain 	Target test run ☐ Prompt for executable target test step Restore	Defaults	Appl	ý
< >				
	Apply and	Close	Cancel	

Target Log Collector

Set target log collector information to use and whether to automatically detect target test results. If you turn on automatic target log detection by clicking the [Atuto-detect on] button, a new target test result is received from the target log collector at a specified automatic detection cycle.

Preferences			– 🗆 X
type filter text	Target Log Collector		← → ⇒ → →
Analysis Editor Exclusion Performance Perspectives Register external commani Report Source File Types Target Test	Auto-detect of targ Use the default target % if check 'Use the default Target log collector info: IP address: Port:	log collector ult target log collector', the target log collector which is installed	Open configuration director with Controller Tester run automatically
RTV Server Target Log Collector > Test Theme (CS Labs) > ToolChain	Auto-detect cycle(sec):		Auto-detect on
< >>			Restore Defaults Apply
			Apply and Close Cancel

Target Log Collector Setting	Description
Auto-detect cycle(sec)	Cycle to check for new target test results
IP	IP on the server or PC where the target log collector runs
Port	Target log collector's communication port

When using the default target log collector, set only the auto-detection interval(sec).

26. CONTROLLER TESTER RTV(Remote Target Verifier)

Introduction

Controller Tester RTV helps you perform automated testing in your Linux environment. After installing the Controller Tester RTV server in your Linux environment and completing the default configuration, you can use the Controller Tester RTV client to perform the RTV test as easily as you would in a hosted environment.

For information on how to install the Controller Tester RTV server in your Linux environment, please contact the Technical support contact on bottom of the <u>Troubleshooting</u> page of this document.

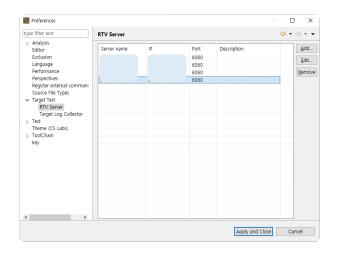
Supported Operating Systems

Server: Linux (RHEL, Ubuntu, Debian, Fedora based) Client: Windows 7/10

26.1. RTV Server Settings

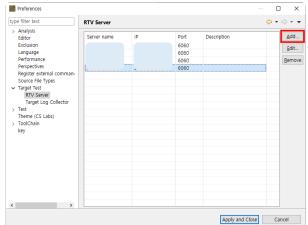
In order to use Controller Tester RTV, you need to add a RTV server with Controller Tester RTV server installed.

The RTV server can be added in [Window] -> [Preferences] -> [RTV Server].



Add a RTV server

1. Click [Add] in [Window] -> [Preferences] -> [RTV Server].

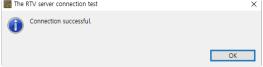


2. In the [Add RTV Server] window, enter the RTV server information.

Add RTV ser	ver		×
Add RTV serve	r erver information		
Server name:			
Port:			
Description:	<u>C</u> onnection Test		
	e e	OK	Cancel

The server name only accepts characters that can be used as Windows file names and can not be modified after adding the RTV server.

3. Click the [Connection Test] button to check the connection status with the RTV server you entered.



4. If the connection test with the RTV server is successful, click the [OK] button to add the RTV server.

add RTV ser	ver X
Add RTV serve	r
Enter the RTV s	erver information
Server name:	RTV_Server
IP:	211.116.222.223
Port:	6060
Description:	RTV server for test
	<u>C</u> onnection Test
	OK Cancel

5. You can see that the RTV server has been added.

type filter text RTV Server				🗢 🖛 🖘 🔹
> Analysis Editor Server name	IP	Port	Description	Add
Exclusion		6060		
Language		6060	RTV Server	<u>E</u> dit
Performance		6060		Remov
Perspectives .		6060		
Register external comman				
Source File Types				
✓ Target Test RTV Server				
Target Log Collector				
> Test				
Theme (CS Labs)				
> ToolChain				
key				
< >				

Edit a RTV server

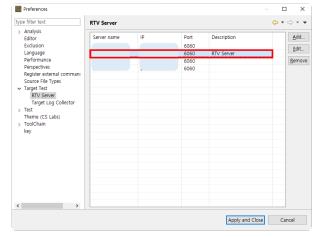
1. Double-click the server in the RTV server list, or select the server and click the [Edit] button.

type filter text	RTV Server				← → ⇒ → →
 Analysis Editor Exclusion Language Performance Perspectives Register external commanics Source File Types Target Test RTV Servieri Target Log Collector > TeolChain key 	Server name	10 	Port 6060 6060 6060 6060	Description Description	Edit.
< >					

2. On the [Edit RTV server] window, edit the server information and then click [OK].

Edit RTV ser	ver	×
Edit RTV serve	r	
Enter the RTV s	erver information	
Server name:	211.116.222.233	
IP:	211.116.222.233	
Port:	6060	
Description:	RTV Server	
	<u>C</u> onnection Test	
	OK Cancel	

3. You can see that the RTV server information has been edited.

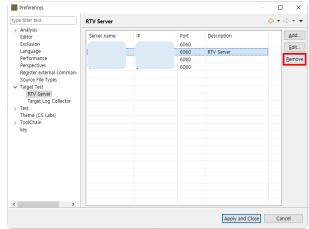


Remove a RTV server

If you no longer use the RTV server, you can remove it.

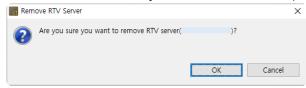
When you remove a RTV server, the toolchain or project resources associated with the RTV server are deleted from the client environment(Can not be deleted if a RTV toolchain added from that RTV server exists).

1. Select a server from the list of RTV servers and click the [Remove] button.



2. Click the [OK] button in the [Remove RTV Server] window(Depending on the size of the RTV

server resource, it may take a minute or two).



3. You can see that the RTV server has been removed.

ype filter text	RTV Server				⇔ - ⇔
 Analysis Solitor Exclusion Language Performance Perspectives Register external commanics Source File Types Target Test Target Log Collector Test Theme (CS Labs) > ToolChain key 	Server name	р р	Port	Description	AddEditBernov

26.2. RTV Toolchain Settings

RTV toolchain

To build and run a test in a RTV environment with Controller Tester RTV, you need to set the toolchain (compiler) information for the source file being tested.

This chapter describes how to add, edit, and delete the RTV toolchain information that has previously extracted from the RTV environment where the Controller Tester RTV Server is installed. (For extracting toolchain information from your RTV environment, please contact the Technical support contact at the bottom of this document <u>Troubleshooting</u> page.)

Add an RTV toolchain

1. Click the [Add RTV Toolchain] button in the [Window] -> [Preferences] -> [Toolchain] window.

type filter text	ToolCl	nain				← → ⇒ →
> Analysis Editor	St	Toolchain Name	Target Co	Conversion Tool	Description	Add Toolchain
Exclusion	SL		-		Description	
Language	1	CPP_TI_TMS320 GCC 4.7 (32bit)	In using	Microsoft Visual		Add <u>R</u> TV Toolchain.
Performance	1	GCC 5.3 (32bit)	In using			Edit
Perspectives	1	Microsoft Visual				<u>_</u> un
Register external comman: Source File Types > Target Test V Test Coverage Export test data External editor	ternal commani		Duplicate			
	1	Microsoft Visual				Remove
		Microsoft Visual	In using			Remove
	ò	gcc5	In using	GCC 5.3 (32bit)		
	ŏ	gcc_5.4	In using	GCC 5.3 (32bit)		
	1	iar		GCC 5.3 (32bit)		
Perspective		101	Hot III do	000 0.0 (020k)		Open Configuration Fo
Type Partition	Detai					
Unit Test View	Detail					
Virtual Address						
Theme (CS Labs)						
> ToolChain						
key						
< >						

If you do not have a RTV server, the [Add RTV Server] window will appear first.

2. If you select the RTV server in the [Add RTV Toolchain] window, you can see the list of RTV toolchain extracted from the RTV server. Select the toolchain you want to add and click the [OK] button.

Add RTV	Toolchain			×
RTV Toolch Select RTV f				
RTV server:	211.116.222.233			~
gj_gcc_	5.4		 	^
☐ gcc6 ☑ gcc_5.4	4			¥
		OK	Cancel	

3. The toolchain information is imported from the RTV server (Depending on the size of the system header, it may take 1 to 2 minutes). When the operation is completed, you can see that the RTV toolchain has been added.

pe filter text	ToolChain					⇔ - ⇒
Analysis Editor Exclusion Language Performance Perspectives	S Toolchain Name ✓ CPP_TI_TMS320_6000_SoC_L ✓ GCC 4.7 (32bit) ✓ GCC 5.3 (32bit) ✓ Microsoft Visual Studio 2010		Target Not in In using In using In using	Conversio Microsoft	Description	<u>A</u> dd Toolchain Add <u>B</u> TV Toolchai <u>E</u> dit
Register external comman Source File Types Target Test Test Coverage Export test data	 Microsoft Visual Studio 2010 Microsoft Visual Studio 2015 Microsoft Visual Studio 2015 Gcc5 gcc, 5.4 	(32bit_64bit) (32bit)	In using In using In using In using	GCC 5.3 GCC 5.3		Duplicate
External editor Perspective Type Partition Unit Test View Virtual Address Theme (CS Labs) > ToolChain key	fiar Detail Toolchain Name C Compiler	gcc_5.4 /usr/bin/	Not in	GCC 5.3		Open Configuration
	C++ Compiler Conversion Toolchain Name Status Description	/usr/bin/ GCC 5.3	g++	oported.		
>	RTV server	211.116.	222.233:60	060		

4. If you select the RTV toolchain added, you can see that the information of the RTV server is displayed on the [Detail].

Edit an RTV toolchain

1. Select the toolchain to be edited and click the [Edit] button.

Edit Too	Ichain —	
oolchain(Method Manage)	
Manage to	olchain information.	
Toolchain	Information	
Name:	gcc_5.4	
Descriptio	n:	
Env Script		Browse
Configurat	ion: Unknown(current usage)	
_		
	**	
Compiler:	/usr/bin/gcc	Browse
Linker:	/usr/bin/gcc	Browse
Archiver:		Browse
System H	eader:	
C:WUsers	#x#AppData#Roaming#CodeScroll#1.1#targetSource#211.116.222.233#f #x#AppData#Roaming#CodeScroll#1.1#targetSource#211.116.222.233#f #x#AppData#Roaming#CodeScroll#1.1#targetSource#211.116.222.233#f	Edit Remove Up Down
<	>	
Libraries:		
		<u>A</u> dd
		<u>E</u> dit
		<u>R</u> emove
utomatica		
	ly extract toolchain information from entered compiler.	Toolchain Inf

- 2. Edit the RTV toolchain information.
- 3. Click the [Finish] button.

Ţ

Duplicate an RTV toolchain

The duplication for an RTV toolchain is not provided.

Remove an RTV toolchain

Select the toolchain to delete and click the [Remove] button.

If you have a project associated with the toolchain you want to delete, you can not use the project normally after deleting the toolchain.

Import/Export an RTV toolchain

The RTV toolchain can not be imported/exported because it can be added through the RTV server.

26.3. Create a RTV Project

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

When building source files in the RTV environment, you can extract the information needed to create the RTV project.

This chapter describes how to create an RTV project by using pre-extracted source file build information(using the Controller Tester RTV Server Utility) in the RTV environment, or extracting information at the same time as building the source file.

To create an RTV project, run the [Create Project Wizard].

Click [File] -> [New] or Shortcut.

File	Edit Search Project	Target Window	v Help
	New	>	C/C++ Project with Source Files
	Close	Ctrl+W	C/C++ Project from Embedded(CodeWarrior, Green Hills, NEC)
	Close All	Ctrl+Shift+W	C/C++ Project from Visual Studio Project
	-		C/C++ Project from Existing CodeScroll Project
	Save	Ctrl+S	Create a C/C++ project with CPI File
	Save All	Ctrl+Shift+S	Create a C/C++ project with Build Information
	Rename	F2	C/C++ Project from RTV Build
	Refresh	F5	C/C++ Project from RTV Build Command
	Switch Workspace		C/C++ Target Test Project with Source Files
ès	Import		C/C++ Target Test Project from Embedded(CodeWarrior, Green Hills, NEC)
4	Export		Create a C/C++ target test project with Build Information
	Properties	Alt+Enter	Ctrl+N Ctrl+N
	Exit		> 🗹 _tr_in:
_			_ > ✓ _tr_st

C/C++ Project from RTV Build

It is selected when creating an RTV project using pre-extracted source file build information (using the Controller Tester RTV Server Utility) in the RTV environment.

1. Enter the project name in [Project name].

New Proje				_		>
ew Project						
Liteate a nev	v project from the RTV	server.				
roject name	RTV_Project_1					
ocation:	D:WDroduct We	orkspace\ct_3_1_02				
		orkspacewci_3_1_02				
Select Toolc	hain	1				
Default	Toolchain Name	RTV server	Description			
	gcc5	211.116.224.99				
	gcc_5.4	211.116.222.233				
				То	olchain	Cott
				10	OICHAIN	Seu
	Γ	< Back Nev	Finish		Cance	al
		< <u>B</u> ack <u>N</u> ext	: > <u>F</u> inish		Cance	e

2. A list of RTV toolchains appears in the [Select Toolchain]. Select the RTV toolchain to use.

3. After selecting the toolchain, then click the [Next] button. You can see the list of source file builds pre-extracted from the RTV server.

New I	Project		- 0	\sim
ew Pro	ject			
elect th	• ne build project on the RTV se	erver.		
	244 446 222 222			
V serve	er: 211.116.222.233			
Build Pr	roject List			
Filter:	Show only creatable project			\sim
		×		
-	Project	Build Time	Build Location	î
	RRAY2]All_Sources	2020-03-31 11:03:37	/home/test/test/test09	
	RRAY2]a.out	2020-03-31 11:03:37	/home/test/test/test09	
	IEWNEWMERGE]All_Sources	2020-03-27 15:15:08	/home/test/test/test05	_
	IEWNEWMERGE]a.out	2020-03-27 15:15:08	/home/test/test/test05	
🗌 [Z	LIB]All_Sources	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
🗌 [Z	LIB]NotLinked	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
	LIB]example	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
	LIB]example64	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
🗌 [Z	LIB]libz.a	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
🗌 (Z	LIB]minigzip	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
🗌 (Z	LIB]minigzip64	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
[5	PRO]All_Sources	2020-03-27 11:43:35	/home/test/test/test03	
< "			4	× .

- Among the generated projects, 'All_Sources_Project Name' is a project that combines all the source files of the modules captured by ctbuild. Since this project is a combined project without considering whether it is built or not, it may not be able to perform the test normally.
- 4. In the [Build Project List], select the item for which you want to create the RTV project and click the [Next] button.

New Project					\times
New Project					
Select the source file.					
RTV server: 211.116.222.	233				
Target Project: [ZLIB]libz.a					
Directory		File			
✓ ■ home		type filter text			
 ■ test ■ test ✓ Zlib-1.2.1 	I	zutil.c gzwrite.c gzclose.c deflate.c gzread.c compres.c inftrees.c trees.c uncompr.c jinfback.c gzlib.c gzlib.c			
15 items selected					
	. Deal-	Nigada		Cance	al
	< <u>B</u> ack	<u>N</u> ext > <u>Finis</u>	n	Cance	ei

5. After selecting source files to be tested, click the [Finish] button to create the project.

C/C++ Project from RTV Build Command

It is selected when creating an RTV project by extracting information at the same time as building the source file. The user builds the source file on the RTV server using the build command and then creates

the RTV project.

1. Enter the project name in [Project name].

	New Proje	ect			_		×			
	New Project									
0	Create a new project from the RTV server.									
P	roject name	RTV_Project_1								
L	ocation:									
-9	Select Toolcl	hain								
	Default	Toolchain Name	RTV server	Description						
		gcc5	211.116.224.99							
		gcc_5.4	211.116.222.233							
							_			
		1		1						
					To	oolchain	Setting			
			< <u>B</u> ack <u>N</u> ext	> <u>F</u> inish		Cance	al l			

- 2. A list of RTV toolchains appears in the toolchain list. Select the RTV toolchain to use.
- 3. After selecting the toolchain, click the [Next] button. The build will be performed on the RTV server from which the selected toolchain was imported.

New Project			×
New Project Please enter a	Build command.		
RTV server:	211.116.222.233		
Build Dircetory			~
Build Comman			
			^
		 	~
Build Log	location		
Location:	P:#Product_Workspace#ct_3_1_02#.metadata	Brow	
	< Back Next > Einish	Cance	4

- 4. Enter the path to build from the RTV server (for example, the directory where the make file is located), and enter the build command.
- In [Build Log], specify the path to the build results file.
 The default location is the .metadata directory of the workspace. If you uncheck [Use default location], you can change the path.
- 6. After entering the required information for the build, click the [Next] button. It builds the source file on the RTV server with the entered build command (depending on the build target, it can take a lot of time).

7. When the build is complete, a list of built projects appears.

	e build project on the RTV se	rver.		
V serve	r: 211.116.222.233			
Duild Dr	oject List			
bullu Pl	-			
Filter:	Show only creatable project			\sim
Target	Project	Build Time	Build Location	^
	RRAY21All Sources	2020-03-31 11:03:37	/home/test/test/test09	
	RRAY2]a.out	2020-03-31 11:03:37	/home/test/test/test09	
	EWNEWMERGE]All_Sources	2020-03-27 15:15:08	/home/test/test/test05	
[] [N	EWNEWMERGE]a.out	2020-03-27 15:15:08	/home/test/test/test05	
[] [Z	LIB]All_Sources	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
[] [Z	LIB]NotLinked	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
[] [Z	LIB]example	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
🗌 [Z	LIB]example64	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
[] [Z	LIB]libz.a	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
[] [Z	LIB]minigzip	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
	LIB]minigzip64	2020-03-27 12:32:07	/home/test/test/zlib-1.2.11	
	PRO]All_Sources	2020-03-27 11:43:35	/home/test/test/test03	
<			4	>

Among the generated projects, 'All_Sources_Project Name' is a project that combines all the source files of the modules captured by ctbuild. Since this project is a combined project without considering whether it is built or not, it may not be able to perform the test normally.

8. In the [Build Project List], select the item for which you want to create the RTV project and click the [Next] button.

New Project	- t					×
New Project						
Select the sou	rco filo					
Select the sou	rce me.					
RTV server:	211.116.222.233					
Target Project:	[ZLIB]libz.a					
Directory			File			
v 🔳 home	9		type filter text			
✓ 🔳 te			Zutil.c			
× 🗉	test zlib-1.2.11		gzwrite.c			
	2110-1.2.11		gzclose.c			
			deflate.c			
			gzread.c			
			inftrees.c			
			✓ trees.c			
			infback.c			
			gzlib.c			
			✓ crc32.c			~
15 items select	ed					
		< <u>B</u> ack	<u>N</u> ext > <u>F</u> inis	sh	Cance	el

9. When creating the RTV project, select the source file to be included and click the [Finish] button to create the project.

RTV build command

To create a project from build information, you must use the ctbuild command on the build command.

ctbuild command

The cs is a utility that automatically extracts the Controller Tester project configuration information of the source files that are built so that the Controller Tester RTV project can be created.

How to use ctbuild

Prefix the existing build command with ctbuild.

• Ex)make \rightarrow ctbuild capture make

ctbuild option

new: Generate the necessary information before extracting build commands. Specify the toolchain with the --toolchain option and the project name with the --project option.

- Ex) ctbuild new --toolchain=gcc5.4 --project=PRJ
- The -- toolchain option is required.

capture: Creates an RTV project by performing build command extraction.

• Ex) ctbuild capture make

convert: Convert a STATIC project to an RTV project.

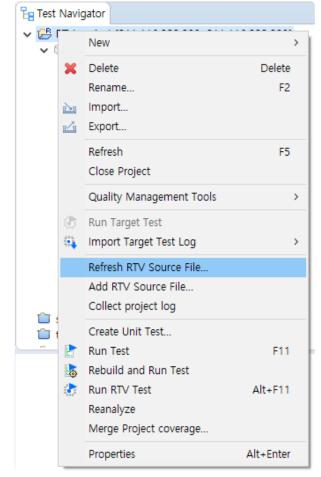
• Ex) ctbuild convert

26.4. Refresh/Add RTV Source Files

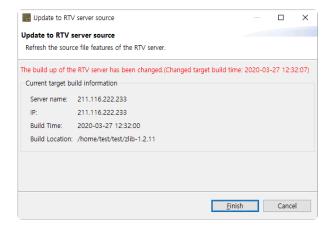
Refresh RTV source file

If the source file is modified and rebuilt on the RTV server, the [Refresh RTV Source File] allows the revised content to be reflected in the project.

- If the source file is modified but not rebuilt, the changed contents will not be reflected.
 If you modify the source files on the RTV server and then rebuild, you must perform a full build.
- 1. In the [Test Navigator] view, click the [Refresh RTV Source File] context menu.



2. When you create an RTV project, you can see the latest build information for the source files you selected (display a change notification if the build has changed). Click the [Finish] button to confirm your changes.



3. After confirming the changes, click the [OK] button.

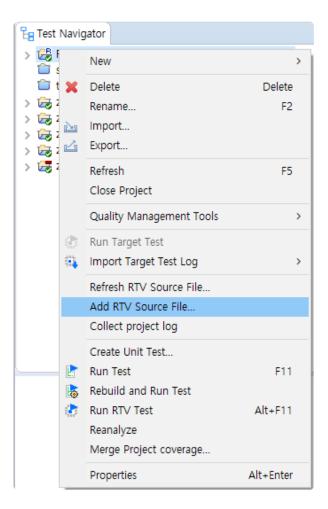
Update to RTV server source	— 🗆 X
Jpdate to RTV server source	
The Source File has been changed to the source file of the RTV server.	
change:	
source file	content
211.116.222.233://home/test/test/zlib-1.2.11/adler32.c	Modified
	OK

Add RTV source file

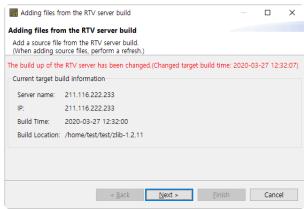
You can add source files to your RTV projects via the [Add RTV Source File] if new source files are added and built on the RTV server, or if only some source files are selected when you create the RTV project.

If the source file is added but not rebuilt, the changed contents will not be reflected. If you build the target server after adding the source file, you must perform a full build.

1. In the [Test Navigator] view, click the [Add RTV Source File] context menu.



2. When you create the RTV project, you can see the latest build information for the selected source file (show change notification if the build has changed). Click the [Next] button to confirm the new file list.



3. In the [New file], select the file you want to add and click the [Finish] button.

C Ac	Iding files from the RTV server build				×
Addir	g files from the RTV server build				
The p	project selected in the table is reflected in the project.				
New 1	ile				
	source file	content		Select A	dl
	211.116.222.233://home/test/test/zlib-1.2.11/inflate.c	New		Clear A	1
	211.116.222.233://home/test/test/zlib-1.2.11/inftrees.c	New		Clear A	
	211.116.222.233://home/test/test/zlib-1.2.11/trees.c	New			
	211.116.222.233://home/test/test/zlib-1.2.11/uncompr.c	New			
	211.116.222.233://home/test/test/zlib-1.2.11/zutil.c	New			
			0/5	Selected	
	< Back Next >	Finish		Cance	el
	L	_			

4. After confirming the changes, click the [OK] button.

— 🗆 🗙
content
New
OK

26.5. RTV Project Information

Test navigator

- 1. The icon for the RTV project is displayed as $\stackrel{\text{\tiny{les}}}{\Longrightarrow}$.
- 2. The RTV server's information ([server name: IP]) is displayed after the name of the RTV project.

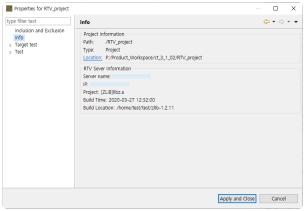
En Test Navigator	
✓	
✓ ⁽ⁱ⁾ Default Module	
> 🔜 adler32.c	
> > compress.c	
> 🔜 crc32.c	
> 🔜 deflate.c	
> 🔜 gzclose.c	
> 🔜 gzlib.c	
> 🔜 gzread.c	
> 🔜 gzwrite.c	
> 🔜 infback.c	
> 🔜 inffast.c	
inflate.c	
. 📄 inftrees.c	
. 📄 trees.c	
🔄 🔲 uncompr.c	
📃 📃 zutil.c	
> 🕒 Headers	

 Source files in the RTV project can not be modified(the source file editor of the RTV project is read-only).

Project property

Info

1. The RTV project displays the RTV server and project information.



2. If the general project has set the toolchain as the RTV toolchain, information about the RTV server from which the RTV toolchain was imported is displayed.

Properties for RTV_project		– 🗆 X
type filter text	Info	↓ ↓ ↓ ↓
Inclusion and Exclusion Info > Target test > Test	Project Information Path: /RTV_project Type: Project Location: P:/Product.Workspace/ct.3_1_02/RTV_project RTV Sever Information Sever Information Sever Information Project: [218]libbca Build Time: 2020-03-27 12:32:00 Build Location: /home/test/test/2lib-1.2.11	
		Apply and Close Cancel

Module/File property

Build

- 1. The RTV project can not change the RTV toolchain that was set when the project was created.
- 2. However, if the RTV toolchain you set originally is deleted, you can change it to another toolchain.
- 3. In general projects, you can change the toolchain to the RTV toolchain. Changing to the RTV toolchain changes all toolchain settings (modules and source files) under the project and allows you to run the RTV tests.
- 4. The project with the RTV toolchain can not change the compiler settings of the source file (the module's linker configuration can be changed).

26.6. Run RTV Test

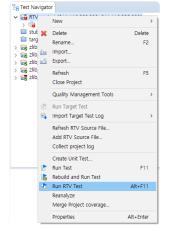
The RTV project allows you to run RTV tests without any additional setup. To test a general project in the RTV environment, you must set the toolchain of the project to the RTV toolchain created from the RTV server to be tested.

Run all RTV tests

It runs the selected tests in the [Unit Test] view and the [Integration Test] view.

RTV tests can be run in the following way.

1. Select the project to be tested, right-click it and click [Run RTV Test].



Run RTV unit test

Runs the selected test in the [Unit Test] view.

Run RTV test

[Run] button In the [Unit Test] view.

Coverage – CMC/DC B Stub Class Factory Control Florences	w Graph 🖧 Call Graph 怜 Fu	nction Call Hierarchy
Unit Test Integration Test		
Run 🕨	(0 / 0 / 0)	 Statement Cover
	Result	Coverage
Name	Result	Coverage
Name adler32(unsigned long, const unsigned char *)	(0 / 0 / 0) 0	Coverage
Name		Coverage - -
Name ✔ adler32(unsigned long, const unsigned char *) ✔ adler32_combine(unsigned long, unsigned long) ✔ adler32_combine64(unsigned long, unsigned long)	(0 / 0 / 0) 0 (0 / 0 / 0) 0	Coverage - - -
Name Z adler32(unsigned long, const unsigned char *) adler32_combine(unsigned long, unsigned long)	(0 / 0 / 0) 0 (0 / 0 / 0) 0 (0 / 0 / 0) 0	Coverage - - - -
<pre>adler32_combine64(unsigned long, unsigned long) adler32_combine_(unsigned long, unsigned long)</pre>	(0 / 0 / 0) 0 (0 / 0 / 0) 0 (0 / 0 / 0) 0 (0 / 0 / 0) 0	Coverage - - - -

Run RTV test case

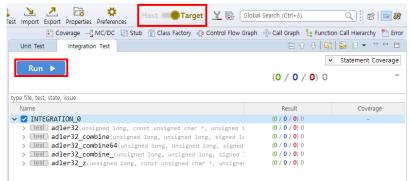
[Run RTV Test Case] context menu (multi-selectable) in the [Unit Test] view.

Unit Test	l l	ntegration Test				- 8
Run 🕨					 Statement Cove 	rage
Kun 🕨				(0 / 0 / 0)	0	-
pe file, functior	n, test	, state, issue				
Name				Result	Coverage	
🗸 🗹 adler3	(uns	igned long, const unsigned ch	ar *)	(0 / 0 / 0) 0	-	
		ler32(unsigned long, const un		(0 / 0 / 0) 0		
C	a	Conv	Ctrl+C			
C		Сору				
C	a	Paste	Ctrl+V			
🗹 adler32	2	Duplicate	Ctrl+D	(0 / 0 / 0) 0	-	
🗹 adler32		Duplicate multiple times		(0 / 0 / 0) 0	-	
🗹 adler32	· ¥	Delete	Delete	(0 / 0 / 0) 0	-	
🗹 adler32	1			(0 / 0 / 0) 0	-	
🗹 compres		Add Test Case		(0 / 0 / 0) 0	-	
🗹 compres		Host Output Value -> Expected Valu	e	(0 / 0 / 0) 0	-	
compres		Target Output Value -> Expected Va	lue	(0 / 0 / 0) 0	-	
🗹 crc32(u				(0 / 0 / 0) 0	-	
🗹 crc32_l	0	Generate binary for debugging		(0 / 0 / 0) 0	-	
crc32_	0	Run RTV Test Case		(0 / 0 / 0) 0	-	
🗹 crc32_0		Run Test Case		(0 / 0 / 0) 0	-	
crc32_0				(0 / 0 / 0) 0	-	
🗹 crc32_1	193	Add Stub		(0 / 0 / 0) 0 (0 / 0 / 0) 0	-	

Run RTV integration test

Runs the selected test in the [Integration Test] view.

[Run] button In the [Integration Test] view.



26.7. RTV Test Results

Coverage

You can change the coverage information (host, target, host/target merged) displayed in each view and editor through the coverage view menu in the top toolbar menu.

You can check the detailed information of each view in <u>Test Perspective</u>.

Test results

You can change the test results through the host/target settings menu on the dashboard.

🕞 zlib	Rew Project	New Test	کر Import	Z Export	Properties	Preferences	Host 🔵 🗩 Target	Clobal Search (Ctrl+3) C
🖉 🗄 Show Coverage			:	Covera	ge 🖂 MC	/DC 🔡 Stub	🕆 Class Factory 🛛 🖓 Con	ntrol Flow Graph 🐁 Call Graph 🔓 Function Call Hierarchy 睯 Error

Stub

The RTV build stubs that are created when running the RTV test is distinct from the build stubs which are created when running the host test.

🔢 Stub 🐹	it 🔛 🕀 🗗 🕏 🔻 🗖 🗖
Manage stub items of zlib project. (2 items)	
type filter text	Stub code Configuration
> adler32(unsigned long, const unsigned char *, u	Stub key: 5822949563779398816
✓	Related Files: P:#Example_Project#c#dip#zlib-1.2.:
ở _tr_align_2 - RTV Build Stub	Test:
	Description:
	RTV Build Stub
	× .
	Save
< >>	5010

Туре	Description
🧬 Target build stub	Auto-created stub during RTV tests

27. CONTROLLER TESTER RTV(Remote Target Verifier) Target Plug-in

Using Controller Tester RTV (Remot Target Verifier) Target Plug-in, software developed in Linux environment can be tested in a real embedded target environment. You can run RTV target after installing Controller Tester RTV server in Linux and setting target environment in Controller Tester. For information on how to install the Controller Tester RTV server in your Linux environment, please contact the Technical support contact on bottom of the <u>Troubleshooting</u> page of this document.

The followings are how to create Controller Tester RTV Target Plug-in project and run a test.

- <u>RTV Server Settings</u>
- <u>RTV Toolchain Settings</u>
- <u>RTV Target Environment Settings</u>
- Create a RTV Target Project
- <u>Refresh/Add RTV Target Source Files</u>
- <u>RTV Target Project Information</u>
- Run RTV Target Test

Supported Operating Systems

Server: Linux (RHEL, Ubuntu, Debian, Fedora based) Client: Windows 7/10

27.1. RTV Server Settings

For RTV server setting to use Controller Tester RTV Target Plug-in, refer to <u>RTV Server Settings</u>.

27.2. RTV Toolchain Settings

For RTV toolchain setting to use Controller Tester RTV Target Plug-in, refer to <u>RTV Toolchain Settings</u>.

27.3. RTV Target Environment Settings

To run tests in a target environment using Controller Tester RTV Target Plug-in, you need to enter information about the target environment as with Controller Tester Target Plug-in. Using information about the target environment entered by the user, Controller Tester builds a test harness in RTV server and run in the target environment. Target log collector can get the test results automatically. The target environment can be set in the project properties page or in the new RTV target test project wizard.

Target environment settings.

Target environment settings are divided into basic information and detailed settings. RTV target project don't need to enter the basic information.

RTV target plugin does not need toolchain selection.

After entering the basic information, detailed settings that require input are displayed.

alysis	Name	Value
ld	language	c
1	Toolchain Name	gcc5.4
	Status	This toolchain is supported.
	C Compiler	/usr/bin/gcc
	System Header(C Compiler)	C:#Users# #AppData#Roaming#CodeScroll#1.1#tar
	Library(C Compiler)	
	C++ Compiler	/usr/bin/g++
	System Header(C++ Compiler)	C:#Users#
	Library(C++ Compiler)	
	Name:	

Target environment detailed settings

Target environment detailed settings are divided into the analysis, build, run, and etc.

Category	Description
Analysis	The toolchain information is displayed, and in the case of a target test project, the target compiler related settings required for analysis are displayed.
Build	The settings for building the test software are displayed.
Run	Settings for running tests and getting results in the target environment are displayed.
etc.	Other settings are displayed. (Program entry point, etc.)

Required settings for each category are indicated in red. Depending on whether the required settings for each category are entered or not, the behavior when clicking the [Run] button is different as shown

Required settings completed category	Description
None	Test run impossible
Analysis	If you click [Run], it overwrites the original source code with the test code. To perform the test, the user must manually build and run the test.
Analysis, Build	If you click [Run], it builds the test code. To perform the test, the user has to manually run the test on the target.
Analysis, Build, Run	Automatically run tests in the target environment.

Target environment detailed settings – Build



Enter a build script path of RTV server into build script area. After entering the build script path, Controller Tester checks validity of the path. If the path is invalid, build script area is displayed in red and if vaild, it is displayed in black.

Settings	
Analysis	☑ Use build script
Build	Build script: /home/ /workspace/RTV_Target_ENmanual/build.bat
Run	
etc.	

Target environment detailed settings – Run

Run setting is divided into batch script area and log interface area. Controller Tester checks validity of batch script area as with build script area. Informations to get results of RTV target test from target environment is entered in log interface area.

Batch script area

Enter a batch script path of RTV server into batch script area as with build script area.

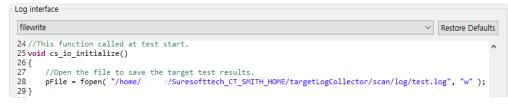
Analysis	Batch script: //home/: . : /workspace/RTV_Target_ENmanual/build.bat	
Build	Log interface	
Run	filewrite	 Restore Defaults
etc.	<pre>//The file is included in all test source files.</pre>	^

Log interface area

The log interface area is a setting to record test result from target environment. The followings are how to get target log into <u>Target Log Collector</u>.

- File
- Socket communication (TCP, UDP)
- Serial communication (UART)

To allow target log collector to get test results during recoding the results in file, you modify a path in log interface into <code>%SMITH_HOME%/targetLogCollector/scan/log/%file_name%</code>.



27.4. Create a RTV Target Project

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

When building source files in the RTV environment, you can extract the information needed to create the RTV project.

This chapter describes how to create an RTV target project by using pre-extracted source file build information(using the Controller Tester RTV Server Utility) in the RTV environment, or extracting information at the same time as building the source file.

To create an RTV project, run the [Create Project Wizard].

Click [File] -> [New] or Shortcut.

Ct New					×
Select a wizard					\$
Wizards:					
type filter text					
	Project arget Project Irget Test Projec Irget Test Projec	t from RTV Build t from RTV Build			
	< Back	Next >	Finish	Cance	1

C/C++ Target Test Project from RTV Build

It is selected when creating an RTV target project using pre-extracted source file build information (using the Controller Tester RTV Server Utility) in the RTV environment.

1. Enter the project name in [Project name] and select the RTV toolchain to use in the [Select Toolchain]. After selecting the toolchain, then click the [Next] button.

Ct New Project						×
New Project Create a new pr	roject from the RTV server.					
Project name: Location:	test D:₩CT34_workspace₩	⊬test				
Select Toolchain Default	Toolchain Name gcc7.5 gcc5.4	RTV server tired tired 	Description	Ī	oolchain	Setting
?		< Back	Next > Fin	ish	Cance	el

2. You can see the list of source file builds pre-extracted from the RTV server. In the [Build Project List], select the item for which you want to create the RTV project and click the [Next] button.

C New Project				×
New Project				
Select the build project on the RTV server.				
RTV server: tired				
Build Project List				
Filter: Show only creatable project				~
show only creatable project				
Target Project	Build Time	Build Location		
[RTV_Target_ENmanual]All_Sources	2020-11-12 17:16:53	/home/ /workspace/RTV_T		
[RTV_Target_ENmanual]NotLinked	2020-11-12 17:16:53	/home//workspace/RTV_T		
?	< Back	Next > Finish	Cano	el

- Among the generated projects, 'All_Sources_Project Name' is a project that combines all the source files of the modules captured by ctbuild. Since this project is a combined project without considering whether it is built or not, it may not be able to perform the test normally.
- 3. After selecting source files to be tested, click the [Next] button.

📴 New Project					×
New Project Select the source file.					
RTV server: tired Target Project: [RTV_Target_ENmanual]All_Sources					
Directory	File				
 ■ home ■ workspace ☑ Workspace ☑ RTV_Target_ENmanual 		filter text ☐ calc.c			
1 items selected					
?	< Back	Next >	Finish	Cano	el

4. After entering the detailed settings for each category and click the [Next] button

ct New Pro	oject		— 🗆 X
Target Envi	ironment Settings		
	nning the target test, it will overv t test manually, or automatically		
RTV targ	et plugin does not need toolchair	selection.	Import Target Environment Setting
Settings			
Analysis	Name	Value	
		c	
Build	language Toolchain Name	gcc5.4	
Run	Status	This toolchain is su	apartad
etc.	C Compiler	/usr/bin/gcc	pported.
	System Header(C Compiler)	C:#Users#i	#AppData#Roaming#CodeScroll#1.1#tar
	Library(C Compiler)	C.#Usels#i	, mappDatamRoarning#Codescroll#1.1#tar
	C++ Compiler	/usr/bin/g++	
	System Header(C++ Compiler		WAppData#Roaming#CodeScroll#1.1#tar
	Library(C++ Compiler)	C.WOSEISW	, mappoatamoanning@codescroimr.rmtar
	Elorary(C++ Compiler)		
	Nama		
	Name:		
	Description:		
(?)		< Back	Next > Finish Cancel
		~ DOCK	Calicer

C/C++ Target Test Project from RTV Build Command

It is selected when creating an RTV target project by extracting information at the same time as building the source file. The user builds the source file on the RTV server using the build command and then creates the RTV target project.

1. Enter the project name in [Project name] and select the RTV toolchain to use in the [Select Toolchain]. After selecting the toolchain, then click the [Next] button.

Ct New Project				_		×
New Project						
Create a new p	roject from the RTV serve	er.				
Project name:	test					
Location:	D:\CT34_workspace	e₩test				
Select Toolchair	n					
Default	Toolchain Name gcc7.5	RTV server	Description			
	gcc5.4	tired				
					ſoolchain	Setting
				-	oorendin	Jetting
?		< Back	Next > Fini	sh	Cance	əl

 Enter the path to build from the target server (for example, the directory where the make file is located), and enter the build command. In [Build Log], specify the path to the build results file. After entering the required information for the build, click the [Next] button. It builds the source file on the target server with the entered build command (depending on the build target, it can take a lot of time).

CI New Project	+					×
	L .					^
New Project						
Please enter a	Build command.					
RTV server:	tired					
Build Dircetory	/home/ /workspace/RTV_Targ	et_ENmanual				~
Build Comman						
gcc calc.c						^
						\vee
Build Log						
Use defaul	t location					
Location:	D:#CT34_workspace#test#.metada	ata			Brov	vse
					Open Di	rectory
?		< Back	Next >	Finish	Cano	el

3. When the build is complete, a list of built projects appears. In the [Build Project List], select the item for which you want to create the RTV target project and click the [Next] button.

C New Project					×
New Project					
Select the build project on the RTV server.					
RTV server: tired					
Build Project List					
Filter:					~
Target Project	Build Time	Build Locati	on		
[RTV_Target_ENmanual]All_Sources	2020-11-12 17:16:53	/home/	/workspace/RTV_T		
[RTV_Target_ENmanual]NotLinked	2020-11-12 17:16:53	/home/	/workspace/RTV_T		
					-11
					-11
					_
					-11
					-11
				-	
0	< Back	Next >	Finish	Cancel	
	< DOCK	NOAL P	1 mail	Carlee	

Among the generated projects, 'All_Sources_Project Name' is a project that combines all the source files of the modules captured by ctbuild. Since this project is a combined project without considering whether it is built or not, it may not be able to perform the test normally.

4. After selecting source files to be tested, click the [Next] button.

C New Project		_		×
New Project Select the source file.				
RTV server: tired				
Target Project: [RTV_Target_ENmanual]All_Sou	rces			
Directory	File			
✓ ■ home ✓ ■ ^{log} → all → al	type filter text			
♥ ■ workspace ♥ RTV_Target_ENmanual	Calc.c			
1 items selected				
?	< Back Next > Fit	nish	Cance	el

5. After entering the detailed settings for each category and click the [Next] button

	et plugin does not need toolchain s	filling out the required fields on the Build, election.	mport Target Environment Sett		
ings					
alysis	Name	Value			
d	language	c			
	Toolchain Name	gcc5.4			
	Status	This toolchain is supported.			
_	C Compiler	/usr/bin/gcc			
	System Header(C Compiler)	C:#Users#4 #AppData#Ro	aming#CodeScroll#1.1#tar		
	Library(C Compiler)				
	C++ Compiler	/usr/bin/g++			
	System Header(C++ Compiler)	C:#Users# AppData#Ro	aming#CodeScroll#1.1#tar		
	Library(C++ Compiler)				
	Name:				

RTV build command

To create a project from build information, you must use the ctbuild command on the build command. For RTV build command, refer to [RTV build command] in <u>Create a RTV Project</u>

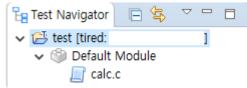
27.5. Refresh/Add RTV Target Source Files

If the source file is modified or added on the RTV server, the [Refresh RTV Source File] and [Add RTV Source File] allow the revised content to be reflected in the project. For Refresh/Add RTV Target Source Files, refer to <u>Refresh/Add RTV Source Files</u>.

27.6. RTV Target Project Information

Test navigator

- The icon for the RTV project is displayed as [™].
- The RTV server's information ([server name: IP]) is displayed after the name of the RTV target project.



• Source files in the RTV project can not be modified(the source file editor of the RTV project is read-only).

Project properties

Info

The RTV target project displays the RTV server and project information.

CI Properties for test		— 🗆 X
type filter text	Info	← ▼ □ → ▼
Inclusion and Exclusion Info > Target test > Test	Project Information Path: /test Type: Project Location: D:/CT34_workspace/test/test RTV Sever Information Server name: tired IP:	
		Apply and Close Cancel

Target test

The RTV target project can set about target test run and RTV environment. For target test run settings, refer to <u>Target Test Run Settings</u> and for RTV environment settings, refer to <u>Target Environment</u> <u>Settings</u>.

Module/file properties

For properties about module or file, refer to <u>RTV Project Information</u>.

27.7. Run RTV Target Test

Test run

The binary file built in Linux environment can be tested in target environment using RTV Target Plug-in. For running the RTV target test, refer to <u>Run RTV Test</u>.

28. DISCOVERY Plug-in

DISCOVERY automatically generates test cases to increase coverage.

The process of executing DISCOVERY is as follows.

- DISCOVERY Install
- DISCOVERY Execution

28.1. DISCOVERY Install

DISCOVERY Install

By using the install package, you can install DISCOVERY as follows:

- 1. Execute DISCOVERY_Installer.exe file.
 - ISCOVERY_Installer.exe
- 2. Set the path where Controller Tester is installed and click [Install].

Install CodeScroll DISCOVERY	×
	DLL
Specify the folder where the CodeScroll is installed,	
To install in this folder, click "Install", Click "Browse" to specify a different folder, or enter the path to the destination folder manually.	e
Folder:	
C:₩Program Files₩Suresoft₩CodeScroll Controller Tester 3,5₩ Browse	
Install Cancel	

If the installation path of Controller Tester is wrong, it will not be installed. If the version of Controller Tester installed is lower than the version of Controller Tester required to install DISCOVERY, it will not be installed.

3. After installation is complete, click [OK] to automatically run Controller Tester.

Install CodeScroll DISCOVERY			\times
			ODESCROLL
Specify the folder where the Co	leScroll is install	ed,	
To install in this folder, click "Ins path to the destination folder ma	tall", Click "Brow nually,	se" to specify a different fo	lder, or enter the
	nstall	×	r I
Folder:	-		
C:₩Program Files₩Suresoft₩	DISCOVI	ERY installation complete.	Browse
		ОК	
		Install	Cancel

28.2. DISCOVERY Execution

If you use the [Test Case Generation using Symbolic Execution] feature, test cases that can increase coverage are automatically generated.

Test Case Generation using Symbolic Execution

- 1. Generate test cases based on symbolic execution for the selected tests.
 - Toolbar in Unit Test View

If you click [Test Case Generation using Symbolic Execution] on the toolbar of Unit Test View, test cases are generated and executed based on symbol execution for the function/ test checked in Unit Test View.

Unit Test	Integration Test	🚱 🕀 🖨 😓 🗄 🔻 🌱 🗖
		Test Case Generation using Symbolic Execution
Run 🕨		(0 / 0 / 0) 0 -

Context menu in Unit Test View

Right-clicking a function/test and selecting the [Test Case Generation using Symbolic Execution] menu generates and executes a test case based on the symbolic execution.

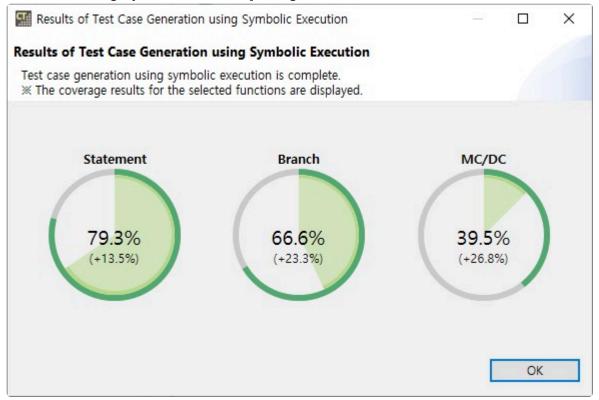
Unit Test In	egration Test		🚱 🕀 🖻 👺 E 🔻 🌱 🗖
			 Statement Coverage
Run 🕨	(933	8 / 0 / 265)	1198 42.8% (1994/4652)
pe file, function, test,	tate, issue		
Name		Result	Coverage
> 🗹 > zlibVersi > 🔽 zlibCompil	Test Case Generation using Symbolic Execution	(1 / 0 / 0) 1 n) 3	100.0% (1/1) 40.4% (19/47)
zcfree(void)	Create Test) 1	100.0% (2/2)
🛛 🗾 zcalloc(voi		3	100.0% (2/2)
> 🗹 zError(sign	Copy Test	Ctrl+C 5	100.0% (1/1)
> 🗹 updatewindo	💢 Delete	Delete 3	13.3% (4/30)
> 🗹 uncompressi	Host Output Value -> Expected Value	3	94.4% (34/36)
> 🗹 uncompress(> 🗹 tr static i	Target Output Value -> Expected Value	3	100.0% (1/1)
<pre>></pre>		10	N/A
Syncsearch(Test reconfiguration	3	0 100.0% (13/13) 100.0% (13/13)
<pre>sinde_nash() send_tree(s</pre>	Set related file	11	44.0% (48/109)
send all tr	Relevant issues settings	8	73.9% (34/46)
Scan tree(s	Relevant issues settings		87.8% (29/33)
read_buf(st	🔄 Import Test Data	> 7	100.0% (14/14)
> 🗹 putShortMSE	🛃 Export Test Data	3	100.0% (2/2)
🗸 🗹 pqdownheap) 6	100.0% (11/11)
🛛 🔽 longest_mat) 6	94.1% (32/34)
> 🗹 lm_init(str	Apply Class Code) 5	100.0% (14/14)
> 🗹 init_block	Select Function) 12	100.070 (10/10)
> 🗹 inflate_tat	Deselect Function) 14	40.370 (05/154)
🛛 🗹 inflate_fa:	Deselect Function) 15	22.470 (37/103)
> 🗹 inflateVali	Generate coverage report) 5	37.5% (3/8)
> 🗹 inflateUnde		(1 / 0 / 0) 1	42.8% (3/7)
> 🗹 inflateSync	Point (struct z_stream_s *)	(1 / 0 / 0)	60.0% (3/5)

2. You can check the progress information after clicking [Test Case Generation using Symbolic Execution].

Progress Information	
Test Case Generation using Symbolic Execution	
Preparing to generate Test Cases using symbolic execution	
	Cancel

When generating a test case based on symbolic execution, it may take a long time to generate test cases.

3. When the test case generation is completed, you can check the coverage through the [Test Case Generation using Symbolic Execution] dialog.



29. CONTROLLER TESTER CLI(Command Line Interface)

You can run Controller Tester in CLI environment.

The process of executing Controller Tester in CLI environment is as follows.

- Install CLI
- Project Import/Export
- <u>CLI Settings</u>
- <u>CLI Execution</u>

29.1. CLI Install

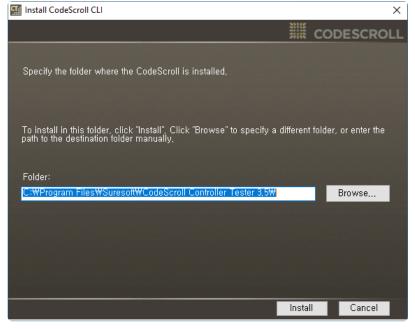
Install CLI

By using the install package, you can install CLI as follows:

1. Run the CodeScroll CLI.exe file.

CT,	CodeScrol	CLI.exe
1000	00000101	021.0/10

2. Enter the path where Controller Tester is installed and click [Install].



If the installation path of Controller Tester is incorrect, it will not be installed.

3. CLI is installed.

🖬 Install CodeScroll CLI		×
	iiii co	DESCROLL
Specify the folder where the CodeScroll is installed,		
To install in this folder, click State 10 Otto 100 on the destination folder Install	≺ ^{""ferent folde}	r, or enter the Browse,
OK		blowse,
	Install	Cancel

29.2. Project Import/Export

You can import and export projects in the CLI environment.

Export command

You can export a project using the -e, --export <project top dir> <global top dir> <exp ort dir> command.

- projectIE.exe path: Controller Tester installation path\plugins\com.codescroll.ut_ut_3.x.x\bin.
- project top dir: export target project directory.
- global top dir: tool global directory.
- export dir: Export output directory.



Export output

- **io**: Folder containing the interface implementation file used for target export. Appears on the global path and does not create it if it does not exist.
- **target_server**: Folder containing RTV target server information. Appears on the global path and does not create if it does not exist.
- factory: Folder used for class factory function. Appears in the project path and does not create if it does not exist.
- pa.ini, ut.ini: Project analysis and test configuration files.
- **build.mk**: RTV project compilation flag file.
- link.mk: RTV project link flag file.
- ctproject.json: Project setting information and exclusion information.
 - infromation: Project information.
 - exclusions: [Project] [Properties] [Inclusion and Exclusion] [Excluded].
 - Inclusions: [Project] [Properties] [Inclusion and Exclusion] [Included].
 - globalExclusions: [Window] [Preferences] [Exclusion].
 - utCoverageExclusions: [Project] [Properties] [Test] [Exclusion of coverage].
 - utUserCodeInsertNodes: Fault Injection view.
 - utClassFactories: Class Factory view.
 - properties: Project setting information.
 - globalProperties: Controller Tester global setting information.
- ctsrc.json: Test target source file information.
 - name: Module name.

- language: Target source language.
- linkFlag: Link flag.
- cmpileFlag: Compilation flag.
- toolchainName: Toolchain name.
- sourceTopDir: Source common parent path.
- src.path: Source file path.
- src.compileFlag: Compilation flag.
- **ctaction.json**: A file that defines actions to be executed for each step.
 - copy: Fields left by the existing copy action migration.
 - beforeCreateProject: Actions executed before project creation.
 - afterCreateProject: Action to be executed after project creation.
 - afterImport: Action to be executed after import is complete.
 - copy action: Description.
 - actionName: Action name.
 - from: Copy destination.
 - to: Path to be copied.
 - isDir: Whether a directory.
- **ctmacro.json**: The macro information inserted in each json file is replaced with the characters in the ctmacro.json file.
 - If you put it in the json file in the form of \${macro name}, it is replaced with the corresponding string when imported.
 - When exporting, the WORKSPACE and GLOBAL paths are created as blanks in the ctmacro.json file.
 - WORKSPACE: The workspace path of the project to be created when exporting.
 - GLOBAL: Global path of PC to be imported.

The WORKSPACE path and GLOBAL path are required.

Import command

- 1. You can import the project using the # -i, --i <import resource dir> command.
 - import resource dir: Import destination directory.
- 2. Import will create a .import_product folder in the destination directory. In that folder, setup files (.cpi, .ini) and .bat files used for project creation and test export import are created.

Create an exportTest folder in the import destination directory and copy the result of [Project] – [Export] – [Test] – [Export Test] to import the test when importing.

C:#Windows#System32#cmd.exe	- 🗆 X
C:\Users\jhyeo\Desktop\project_cli_test>"C:\Program_Ei plugins\com.codescroll.ut_3.5.2\bin\projectlE.exe" -i INFO	iLes#Suresoft#CodeScrol Controller Tester 3.5₩ ∧ "D:₩exportReportTest"
======================================	100%
INFO success create project!! INFO insert project information INFO run copy action	
[BEGIN : Project operation]	
[END : Project operation]	
[BEGIN : Analysis operation]	
[START : PA]	
[END : PA]	
_ [START : ckdb]	
[=====================================	100%
[END : Analysis operation]	
[BEGIN : InitializeTest operation] []	100%

29.3. CLI Settings

Creates a configuration file to run Controller Tester in CLI environment.

Creating CLI configuration file

• If you select [Project] – [Generate command line interface INI file], the projectName_CLI.ini file and path.bat file are created in the path selected by the user.

	Proje	ect	Window	Help
		Ор	en Project	
		Clo	ose Project	5
2	ĪF	Init	ialize Proje	ct
	Generate command line interface INI file			
		Со	llect project	t log
	_	Pro	perties	

If there is no selected project, CLI.ini file and path.bat file that does not contain project information are created.

[Information]

C

Item	Description
workspace	Workspace path.
ct_version	Controller Tester version.

* [Information] automatically fills in the values when creating the CLI configuration file.

[Project]

Item	Description
create	Whether to create a project.
cpi_file	If create=true, the path to the .cpi file for project creation.
name	Project name.
toolchain	Toolchain name (create in case of RTV project).
target_project	Target project name (create in case of RTV project).

The target_project option must be entered in the same way as the target project name in [Project]-[Properties]-[Information]-[RTV Server Information].

[Analysis]

ltem	Description
run	Whether to perform analysis.

[InitializeTest]

Item	Description
run	If true, delete all existing tests.

[CreateTest]

ltem	Description
run	Whether to generate unit tests.
target_function	It is a function name for generating unit tests. Unit tests can be created for one or more functions. When entering multiple functions, they can be separated with a semicolon (;) or Enter. If there is no target_function list, create unit tests for all functions.

[ImportTestInfo]

ltem	Description
run	Whether to fetch test information.
testinfo_file	Path to the testinfo.export file.

[CreateTestCase]

Item	Description
run	Whether to create a test case.
mode	Test case generation mode. (flat, pairwise, random)
random_testcase_count	In case of random, input the number of test cases generated.

[ExecuteTest]

Item	Description
run	Whether to run unit test.

remote_target	RTV run or not.	
run_robust	If an error occurs while running a test, whether to exclude non-executable tests.	

[Report]

Item	Description
run	Whether to generate a report.
output_dir	Report output path. The default output path is project path.
each_test	Whether to generate a report by test.
HTML, XLSX, PDF, DOCX, PPTX, XLS, DOC, PPT	Specify Report Format
URL	URL of the results report accessible from ITS

Individual import of test information

By setting run=false in [ImportTestInfo], [ImportStub], [ImportTest], and [ImportIntegrationTest] can be executed individually.

[ImportStub]

ltem	Description
run	Whether to perform stub import.
stub_file	Path to the stub file.
stub_dir	Path to the stub directory.

[ImportTest]

Item	Description
run	Whether to perform test import.
test_source_file	.utest file path. When entering multiple source file paths, they can be separated with a semicolon (;) or Enter.
test_source_dir	Path to the test source directory. If you enter the test source directory path, perform a full test import included in the directory path.

[ImportTestCase]

ltem	Description
run	Whether to perform test import.
testcase_	Test case directory path. If you enter the test case directory path, import all test cases included in the directory path.

[ImportIntegrationTest]

ltem	Description
run	Whether to perform integration tests and import of test cases.
test_file	.itest file path.
test_dir	Common parent path for .itest file. If the test and test case are in the same directory, perform an import.

Create CPI file

CPI file is a configuration file for creating a project in CLI environment. The project is created only when the [Project] – create option in the CLI.ini file is set to true.

You can set the option to create a project from the CPI template file in the path of Controller Teste r installation path/plugins/com.codescroll.gp.cli_version/cpi.

You cannot create a project in the CLI environment while using the workspace.

Item	Description
KIND	Project creation type setting. General project =1 Get Build Script = 2 Visual Studio Import =3 Import Embedded Project = 4
NAME	Project name.
LOCATION	Location of project creation.
DESCRIPTION	Project description.
LANGUAGE	Project type. Java = 0, C/C++ = 1

SourceFileProject

Item

Description

IUT_LANGUAGE	Target language type. C=0, C++=1, Java=2
DBSCHEMA	Global database schema version.
LINKFLAG	Link flag.
COMMON_COMPILEFLAG	Compilation flag to apply to all TUs.
TOOLCHAIN_NAME	Toolchain name.
BINARY_KIND	Binary type.
SOURCE_TOP_DIR	Source file top level directory.
SOURCE	Absolute path to source file.
COMPILEFLAG	Compilation flag to apply only to source files.
COMPILER_PATH	Path to the compiler to apply only to source files.

VisualStudioPorject

Item	Description
PROJECT_PATH	Visual Studio project file to import.
ENVFILE	Environment variable required for import.

BuildScriptProject

Item	Description
BUILD_COMMAND	Build script to perform the import.
WORKING_DIRECTORY	Working directory to perform import.
ENVFILE	Environment variable required for import.

EmbeddedProject

Item	Description
COMMON_COMPILEFLAG	Compilation flag to apply to all Source Files.
TOOLCHAIN_NAME	Toolchain name.
PROJECT_PATH	Visual Studio project file to import.

29.4. CLI Execution

Controller Tester can be done in CLI environment.

Performing Controller Tester with CLI

- 1. Execute the path.bat file created with [Generate command line interface INI file] to get the path information necessary for CLI execution.
- 2. Execute UtCLI.exe by passing the CLI.ini file edited by the user as an argument.
 - UtCLI.exe path: Controller Tester installation path\plugins\com.codescro ll.ut ut 3.x.x\bin

:₩Users₩jhyeo#Desktop#example>path.bat sWihveoWDesktopWexample>set CODESCROLL_HOME=C:\\Program Files\\Suresoft\\CodeScroll Controller Tester 3.5\\program Files\\CodeScroll.gp.core_1.0.2.20 #Users#jhved#Desktop#example>set CSUT_DLL=C:##Program Files##Suresoft##CodeScroll Controller Tester 3.5##plugins##com.codescroll.ut.core.win32.x86_64_1. #UsersWihyeo#Desktop#example>set CSUT_HOME_BIN=C:##Program Files##Suresoft##CodeScroll Controller Tester 3.5##plugins##com.codescroll.ut_3.5.2##bin## s#jhyeo#Desktop#example>set CSGLOBAL_HOME=C:###Users##jhyeo###AppData###Roaming###CodeScroll###1.1## #Desktop#example>set CODESCROLL_JAVA_HOME=C:###Program Files##Suresoft##CodeScroll Controller Tester 3.5##ire## #jhyeo#Desktop#example>set CODESCROLL_JAVA_BIN=C:##Program Files##Suresoft##CodeScroll Controller Tester 3.5##jre##bin## combest connexample>set_dAva_nome=c:wwwProgram FilestWSuresoftWWCodeScroll Controller Tester 3.5tWjreWW combest connexample>set_path=C:WWProgram FilestWSuresoftWWCodeScroll Controller Tester 3.5tWjreWW:C:WWProgram FilestWSuresoftWWCodeScroll Cont intW:C:WProgram Files(K80)WConnon FilestConceleWJavaWiayaaptiC:WWIndowsW:S:WWIndowsC:WWIndowsC:WWIndowsC:WWForgram FilestWJavaptiC:WProgram FilestWSuresoftWCodeScroll Cont itWWEb Platform InstallerW:C:WProgram Files (x80)WWicrosoft SQL ServerW90WToolstWindowSC:WWIndowsC:WWIndowsC:WWIndowsC:WWIndowsC:WWIndowsC:WWIndowsC:WWForgram FilestWJavaptiC:WProgram FilestWJavaptic:WProgra #jhyeo#Desktop#example>set JAVA_HOME=C:##Program Files##Suresoft##WCodeScroll Controller Tester 3.5##jre### rsWjhyeoWDesktopWexample>"C:WProgram FilesWSuresoftWCodeScroll Controller Tester 3.5WpluginsWcom.codescroll.ut_3.5.2WpinWUtCLl.exe" project_CLl.ini Scroff Controller Tester 3.5mmplugtnsmmcom.codes pdeScroll Controller Tester 3.5mmplugtnsmmcom.codes MCodeScroll Controller Tester 3.5mmpluginsmmcom.co ming#MCodeScroll ##1.1## MSuresoft#MCodeScroll Controller Tester 3.5##jre## rogram Files##Suresoft### codescroll.ut_3.5.2##bin## Guresoft₩₩ ppDat csglhome=C:\\\U00ffubers\\U00f

3. The Controller Tester function set by the user is performed.

[projectproject CLI start!!!]
[BEGIN : Project operation] [END : Project operation]
[BEGIN : Analysis operation] [START : PA] Suresoft (R) Program Analyzer (internal use only) Copyright (C) Suresoft Technologies Inc. All rights reserved.
Build : May 5 2021 20:10:42 REV : ENT 6.2 Windows(Release) {ce4b8c6 2021-05-04 15:42:28 +0900} LIBANALYSIS 6.2 Windows(Release) {ba34ac82 2021-05-03 14:05:21 +0900} LIBSST 6.2 Windows(Release) {ab41074 2021-04-28 14:53:32 +0900}
VALUE1 PHASE21002 VALUE100 log4j:WARN No appenders could be found for logger (com.codescroll.ut.module.cwg.generator.ClassFactoryManager). log4j:WARN No appenders could be found for logger (com.codescroll.ut.module.cwg.generator.ClassFactoryManager). log4j:WARN See http://logging.apache.org/log4j/1.2/fag.html#noconfig for more info. Load JDBC DriverOK [END : PA] [START : ckdb] VALUE1 VALUE100 [END : ckdb] [END : Analysis operation]
[BEGIN : InitializeTest operation] VALUE1 VALUE10 VALUE30 : Delete Test From DB VALUE30 : Delete Test From Dir VALUE70 : Delete Test From Dir VALUE100 [END : InitializeTest operation]

30. EULA(End-User License Agreement)

Software End User License Agreement

Important: This End User License Agreement ("EULA") is a legal agreement between the user (individual or organization) and Suresoft Technologies, Inc. ("SureSoft Tech") which is the manufacturer the software products ("Software product" or "Software") identified by the product identification card or recognition label attached to this EULA. The software product includes computer software, related media, related documents, and "online" or electronic manual. When installing, copying, downloading, backing up, accessing or using the software product, the user must accept the terms of use in the EULA; if the user does not accept the terms of use in EULA, the user may not be licensed for the software product. Unless the user is licensed to use the software product, the user cannot install, copy, download, backup, access or use this software product.

Software Product License

The Software product is protected by intellectual property rights as well as copyrights, program copyrights, and international copyright regulations. The term "computer" used in this agreement means a single computer system, the range of which is described below.

1. License Grant

The EULA grants the user the following rights.

Software Installation and Use. Except for those clearly explained in the EULA, only one copy of software product can be installed, used, accessed, executed or interacted ("execution") on the computer. To install, use or execute the software product on two or more computers, the user must obtain the legal license(s) that corresponds to the number of computers regardless of the interaction of software used or executed on each computer.

Backup Copy. When there is no software product backup copy on the computer, the user can create a single backup copy that corresponds to the computer software part of the software product. The backup copy can be used for the purpose of recording. To create a single backup copy, a backup utility can be used. Except for those clearly stated in this EULA, the user may not make copies of the software product including documents provided with the software product.

2. Other Rights and Limits

Computer or Single Computer System. The Software product's license is applied only to a single computer system. If the single computer system has multiple CPUs, it is considered as a single computer system only when these CPUs are installed to one circuit board and the structure is the symmetric central processing structure that shares one system bus.

Choose language versions. When the software product is included in one or more language versions, it is allowed to use only one language version.

Reverse engineering limitations. De-compilation and decomposition. The user cannot decompile or decompose software the product using reverse-engineering.

Component separation. The software product is allowed as a single product. Users cannot separate the components to use in one or more computers.

Lease. The Software product cannot be leased or lent.

Right Transfer of software product. The user may permanently transfer all rights as part of the sale of hardware or right transfer according to this EULA, it is regarded that the recipient has accepted the terms of use in this EULA.

Expiration. When the user violates the terms and conditions in the EULA, without infringing other rights, Suresoft Technologies can make user's rights expire according to the terms and conditions in this EULA. In such a case, the user must discard all the duplicates and composing parts, including software product that has been already installed.

Registered Trademark. The EULA doesn't grant any rights related to the registered trademark and service trademark of the software manufacturer, the computer manufacturer, and other related suppliers. **Upgrade.** If the software product is upgraded via Suresoft Technologies, the terms of use in the EULA are still valid for the upgraded software and it cannot be separated to use in one or more computers.

3. Copyright

Every right and intellectual propriety right (including all the image, figure, animation, video, audio, music, text, and applet of the software product) of the software product, the enclosed manual, and every copy of the software product are the property of software manufacturer or supplier. All rights and intellectual property rights for the contents that are not built-in the software product but are accessible through the use of the software product are the property of the Company, and this EULA does not grant any rights for such use. When software product includes the manual in a file, only one copy of that file can be printed. The documents accompanied by the software product cannot be copied. All the rights that are not clearly stated are the property of the corresponding software manufacturer and supplier.

4. Double Media Software Product

The Software product can be delivered in one or more media. Regardless of the received media's type or size, only one media can be used in the computer. Other media cannot be executed on another computer. Except for the case of transferring rights permanently (as described above), the user cannot lease or lend the other media or transfer the right to other users.

5. Guarantee

The Software product's guarantee on quality follows the attached warranty. If there is no separate warranty enclosed, Suresoft Technologies guarantees that there was no defect during the production process if it has been normally used for one year from the day of purchase. This does not mean that the software product has no error, and does not exclude the possibility that unexpected result may appear when using the product. Except for the cases where it is clearly stated in the separate warranty agreement, Suresoft Technologies is not liable not only for the inaccuracy, error or defect of the software product but also for the loss and damage produced from these. Suresoft Technologies doesn't take any responsibility for the third party's claims for damage or the user's claims for damage to the third party. The Limited warranty statement, mentioned in the EULA, is still valid even though the user notified Suresoft Technologies or its official representative in advance of the possibility mentioned above.

6. Export limitation

The right transfer of the software product can be carried out only within Korea, and it cannot be exported to other countries.

7. JAVA Support

The Software product includes the support of programs written in JAVA. JAVA technology is not designed as an online control device, or not made, used or sold in dangerous environments that need to error preventive function, such as nuclear facility, aircraft navigation, aircraft control, medicine, medical system, and weapon system that can cause death, injury, or critical physical/environmental damage.

Caution

© Suresoft Technologies, Inc. All copyrights reserved. Printed in Korea Suresoft Technologies and CODESCROLL are trademarks of Suresoft Technologies, Inc.

The product name or trade name mentioned in this document and related document are the trademarks or registered trademarks of the Company having the copyright for the product. The software described in this document is provided according to the usage agreement and non-disclosure agreement.

The software may only be used or copied according to the terms of use in this Agreement.