



Release Note

3.4 — Last update: Dec 17, 2020

Suresofttech

Table of Contents

1. 기능 개선.....	1
2. 기능 추가.....	3
2.1. RTV 타겟 C/C++ 프로젝트 추가.....	4
2.2. 소스 코드 변경 후 재사용 기능 추가	5
2.3. 빈 입력값에 기본값 삽입 기능 추가	8
3. 버그 수정 및 기능 변경.....	9
3.1. 버그 수정 목록	10
3.2. 기능 변경 목록	11

1. 기능 개선

C++ 테스트 개선

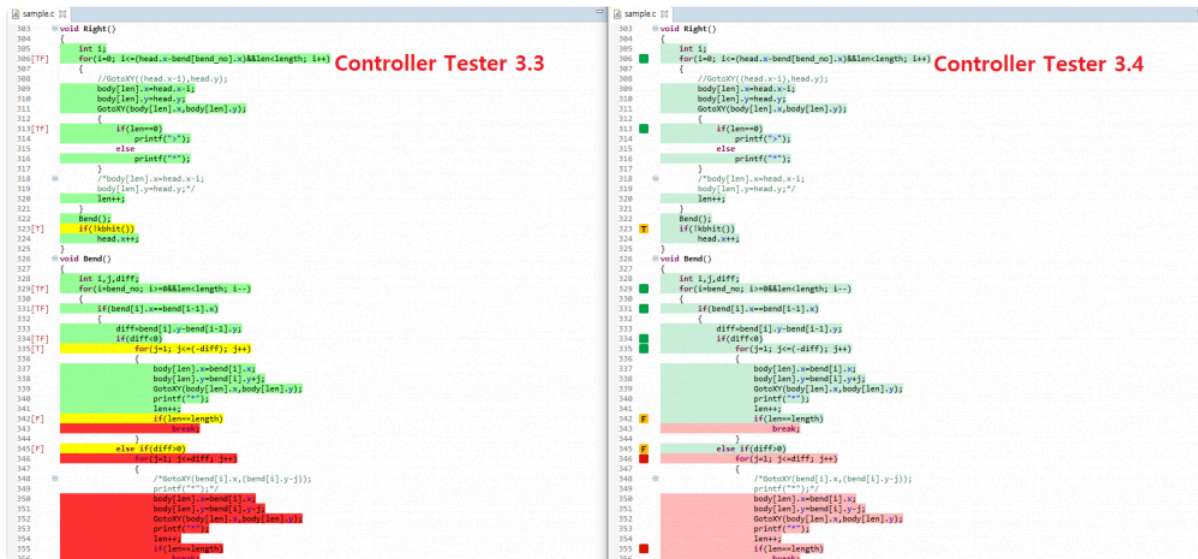
- 테스트 생성 엔진 개선
- 클래스 팩토리 개선

테스트 편집기의 검색 기능 개선

- 구조체의 멤버 변수를 검색할 수 있도록 개선
- 시작 문자열이 일치하지 않아도 검색할 수 있도록 개선

소스 코드 편집기에서 커버리지 표시 방식 개선

커버리지를 표시하는 하이라이트의 색상과 분기의 마커를 다음과 같이 개선



3.3 이전	3.4 이후	설명
[T]	T	분기가 true만 성공한 경우
[F]	F	분기가 false만 성공한 경우
[TF]	■	분기가 true/false 모두 성공한 경우
	■	분기가 커버되지 않은 경우
ex)[TF][T]	■	한 줄에 여러 분기가 있는 경우

- 한 줄에 여러 분기가 있을 때, Controller Tester 3.3 이전에는 분기 결과를 나열하는 형식으로 표시

그 외

- 타깃 테스트에서도 CS_XX_INPUT(OUTPUT) 매크로 사용 가능
- Code Composer Studio 프로젝트가 닫혀있는 경우 command line으로 빌드 가능

- CAVM 코드 생성 옵션 처리

2. 기능 추가

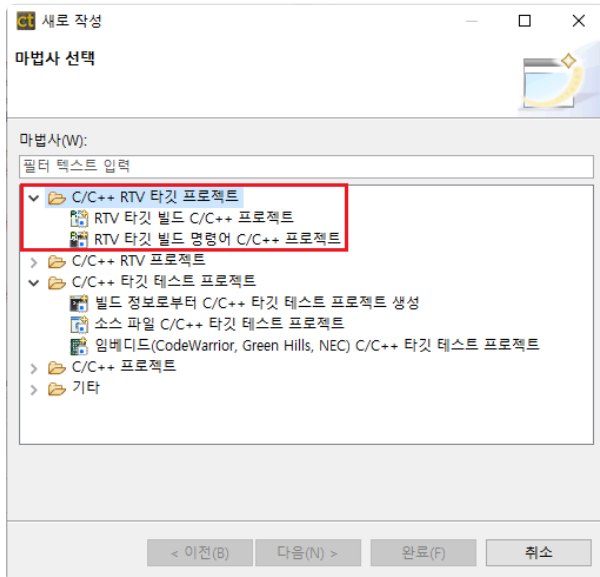
- [RTV 타겟 C/C++ 프로젝트 추가](#)
- [소스 코드 변경 후 재사용 기능 추가](#)
- [빈 입력값에 기본값 삽입 기능 추가](#)

2.1. RTV 타깃 C/C++ 프로젝트 추가

RTV 서버에 존재하는 타깃 소프트웨어의 소스 파일을 테스트하기 위해 RTV 타깃 프로젝트를 생성합니다.

RTV 타깃 C/C++ 프로젝트 생성 기능 추가

- RTV 타깃 빌드 C/C++ 프로젝트
- RTV 타깃 빌드 명령어 C/C++ 프로젝트



RTV 타깃 프로젝트 설정 기능 추가

다음과 같은 과정을 거쳐 RTV 타깃 프로젝트를 만듭니다.

- RTV 서버 설정
- RTV 툴체인 설정
- RTV 타깃 환경 설정
- RTV 타깃 프로젝트 만들기

RTV 타깃 테스트 생성 및 실행

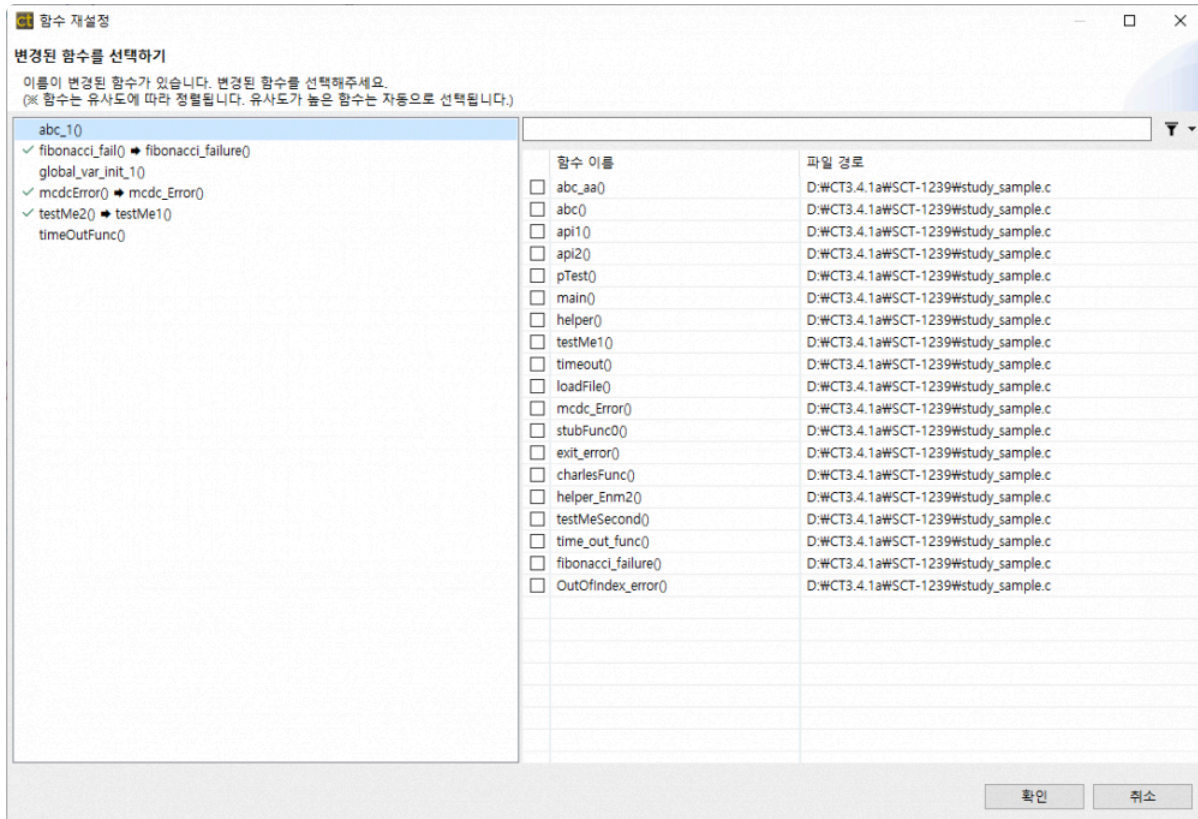
유닛/통합 테스트를 생성하여 호스트/타깃 실행을 할 수 있습니다.

2.2. 소스 코드 변경 후 재사용 기능 추가

소스 코드에서 함수의 이름이나 리턴 타입, 매개변수의 개수나 이름이 바뀐 경우 무결성 검사를 이용하여 재설정 창을 띄웁니다.

함수 재설정

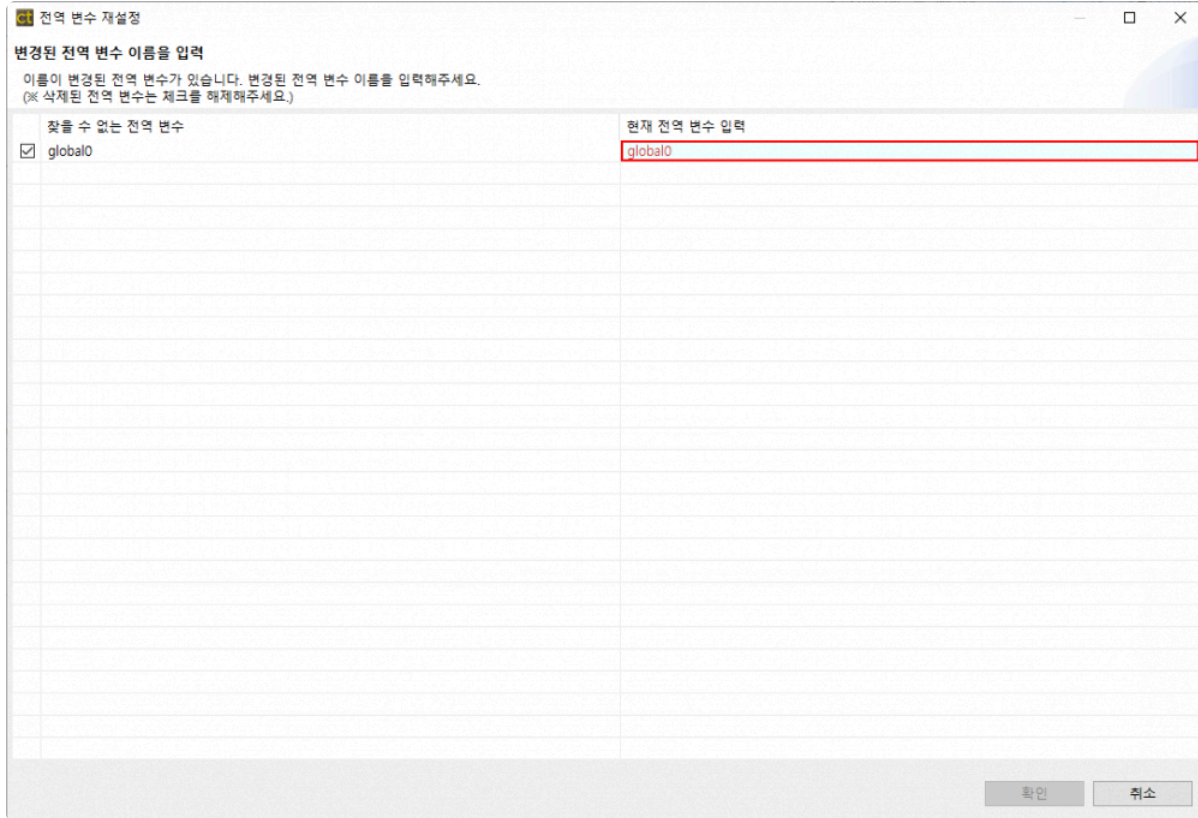
소스 코드에서 테스트 대상 함수나 스텝 대상 함수의 이름을 변경한 경우, [함수 재설정] 기능을 사용할 수 있습니다.



유사도에 따라 함수를 정렬하고 유사도가 높으면 자동으로 선택합니다.

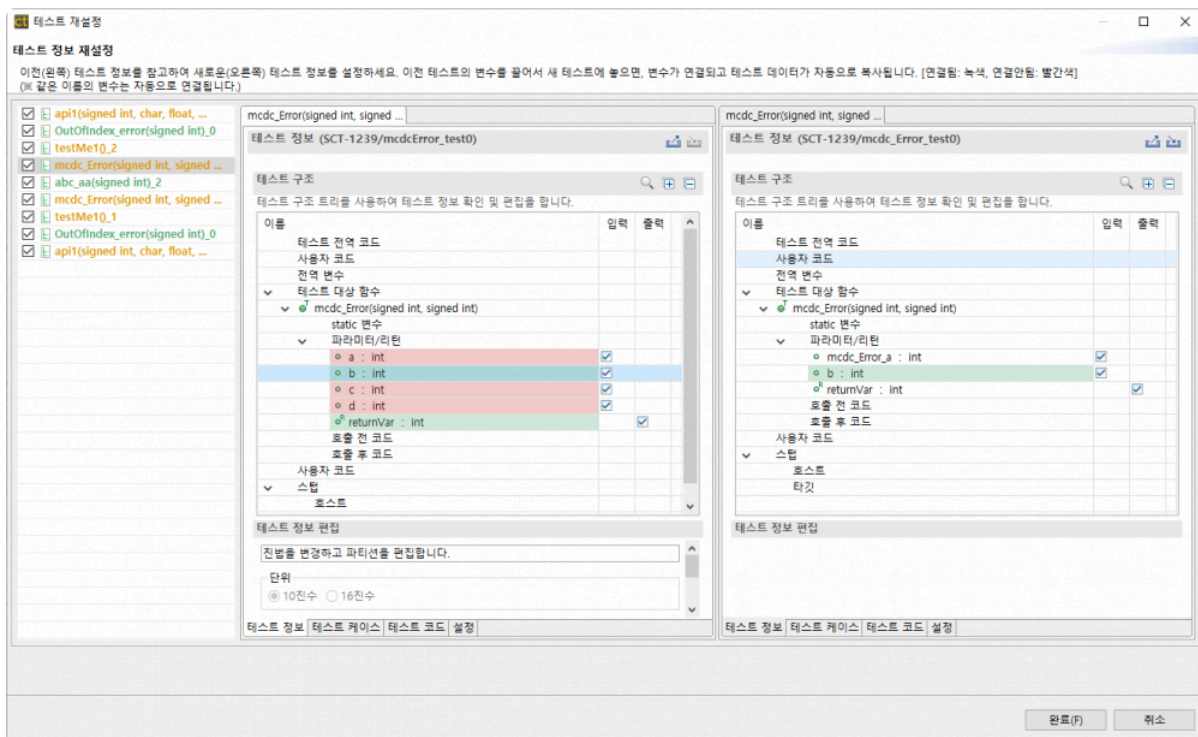
전역변수 재설정

소스 코드에서 테스트에 사용하는 전역 변수의 이름을 변경하는 경우, [전역 변수 재설정] 기능을 사용할 수 있습니다.



테스트 정보 재설정

소스 코드에서 테스트 대상 함수의 매개변수 개수나 이름이 변경된 경우, [테스트 정보 재설정] 기능을 사용할 수 있습니다.



이전 테스트 정보를 참고하여 새로운 테스트 정보를 설정할 수 있습니다. 이전 테스트의 변수를 끌어서 새 테스트에 놓으면, 변수가 연결되고 테스트 데이터가 자동으로 복사됩니다.

* 테스트를 우클릭하여 수동으로 테스트 재설정을 할 수 있습니다.

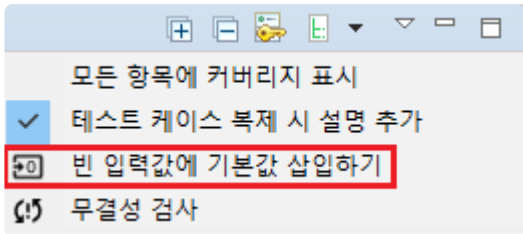
무결성 검사

소스 코드를 변경하고 재설정을 취소하는 경우, 유닛 테스트 뷰의 보기 메뉴에 [무결성 검사]를 통해 수동으로 할 수 있습니다.

2.3. 빈 입력값에 기본값 삽입 기능 추가

테스트 케이스의 입력값이 비어있는 경우, 테스트 뷰 영역의 풀다운 메뉴에서 [빈 입력값에 기본값 삽입하기]를 선택하여 기본값을 입력할 수 있습니다. 입력되는 기본값은 다음과 같습니다

문자열	(빈 문자열)
문자열이 아닌 경우	0



3. 버그 수정 및 기능 변경

- [버그 수정 목록](#)
- [기능 변경 목록](#)

3.1. 버그 수정 목록

- wchar 테스트를 생성할 때 발생하는 문제와 함수 호출 정보를 출력할 때 생기는 오류 수정
- CLI로 RTV 프로젝트 생성 시 link.mk 파일이 자동으로 생성되지 않는 버그 수정
- [테스트 데이터 내보내기] 할 때, 호스트와 타겟 스텝의 심볼을 중복 심볼로 인식하는 문제 수정
- CCS의 tes 옵션에서 발생하는 대소문자 구별, description, breakpoint 문제 수정
- RTV 프로젝트를 생성할 때, '다음' 버튼이 동작하지 않는 버그 수정
- 동일한 함수가 서로 다른 TU에 있을 때 테스트 생성이 되지 않는 문제 수정
- 테스트 실행 시 빌드 스텝이 disable 되는 문제 조치
- 스텝 심볼의 테스트 데이터에 대한 전처리 로직 오류 수정
- asm pragma에 관련된 코드를 변환할 때 생기는 오류 수정
- 테스트 케이스 진단 메시지가 너무 큰 경우 테스트별 보고서의 엑셀 파일이 비정상적으로 생성되는 문제 수정
- 테스트 데이터의 description에 큰따옴표가 포함되어 있으면 Controller Tester에서 [프로젝트 내보내기] 시 파일이 깨져 [프로젝트 가져오기]가 되지 않는 문제 수정
- 구조체 포인터 멤버에 대해 CS_OUTPUT 매크로가 비정상적으로 생성되는 문제 수정
- CCS 스크립트를 사용할 때, entry point가 main이 아닌 경우 테스트 실행이 되지 않는 문제 수정
- [디버그 정보 확인] 수행시 발생하는 오류 수정
- 테스트 데이터를 자동으로 생성하는 매크로 수정

3.2. 기능 변경 목록

커버리지 표시 기준 변경

커버리지 표시 기준을 분기 커버리지에서 문장 커버리지로 변경하였습니다.

Controller Tester 3.4	Controller Tester 3.3
<pre> 326 void Bend() 327 { 328 int i,j,diff; 329 for(i=bend_no; i>=0&&len<length; i--) 330 { 331 if(bend[i].x==bend[i-1].x) 332 { 333 diff=bend[i].y-bend[i-1].y; 334 if(diff<0) 335 for(j=1; j<=(-diff); j++) 336 { 337 body[len].x=bend[i].x; 338 body[len].y=bend[i].y+j; 339 GotoXY(body[len].x,body[len].y); 340 printf("%*"); 341 len++; 342 } 343 if(len==length) 344 break; </pre>	<pre> 326 void Bend() 327 { 328 int i,j,diff; 329 for(i=bend_no; i>=0&&len<length; i--) 330 { 331 if(bend[i].x==bend[i-1].x) 332 { 333 diff=bend[i].y-bend[i-1].y; 334 if(diff<0) 335 for(j=1; j<=(-diff); j++) 336 { 337 body[len].x=bend[i].x; 338 body[len].y=bend[i].y+j; 339 GotoXY(body[len].x,body[len].y); 340 printf("%*"); 341 len++; 342 } 343 if(len==length) 344 break; </pre>