**VectorCast单元测试工具使用说明**

1. **构建测试环境**

VectorCast测试工具会根据被测代码和开发环境自动构建测试架构，在此基础上用户可以根据需要自动地生成或添加测试用例，并灵活的配置以实现不同的测试需求，自动生成的架构中包括：被测对象、测试驱动、桩函数和依赖条件。

**1.1 配置工作目录**

在构建环境前，需要先配置工作目录（保存所有配置、项目组文件）。

默认工作目录为C:\VCAST\Environments

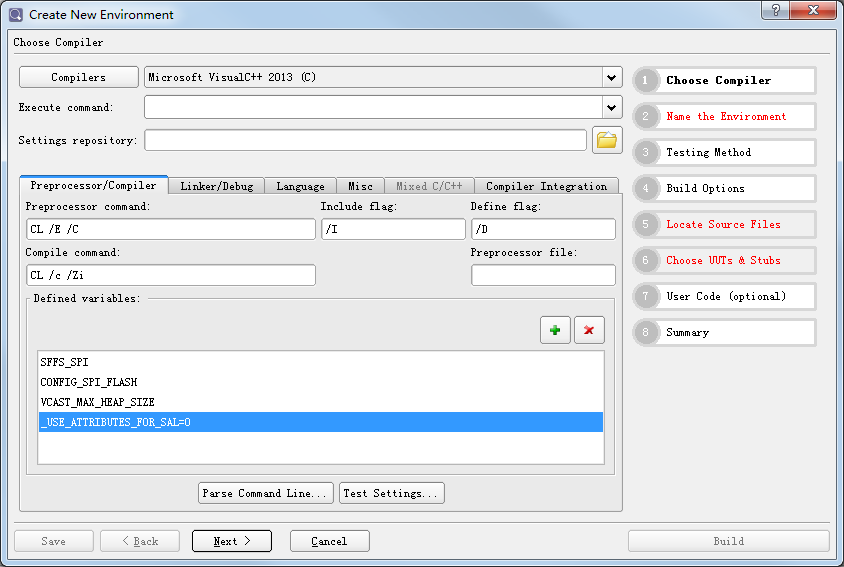
可通过File->Set Working directory，设置新的工作目录。

**重要提示**：一个芯片一个工作目录，否则编译环境时会相互影响。

**1.2 创建环境**

选择File->New->C/C++ Environment，打开新环境创建配置框

（1）**Choose Compiler**选择编译器，以Microsoft VisualC++ 2013为例



（2）**Name the Environment**给新环境命名，如TESTA

（3）**Testing Method**选择测试方法

这里有四种方法：

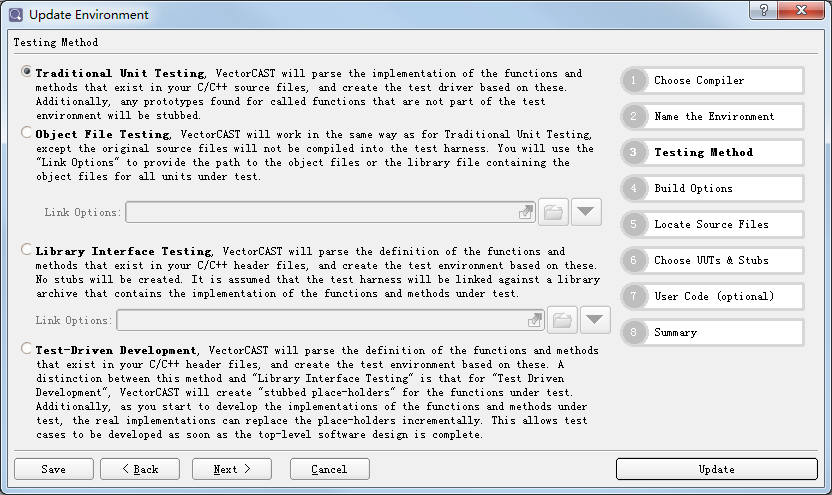
Traditional Unit Testing: 传统的单元测试（默认选择）

Object File Testing: 针对目标文件（Obj文件）测试

Library Interface Testing: 针对库接口测试

Test-Driven Development: 测试驱动开发

其中方法2和3不需要源码，但无法插桩，看不到覆盖率。

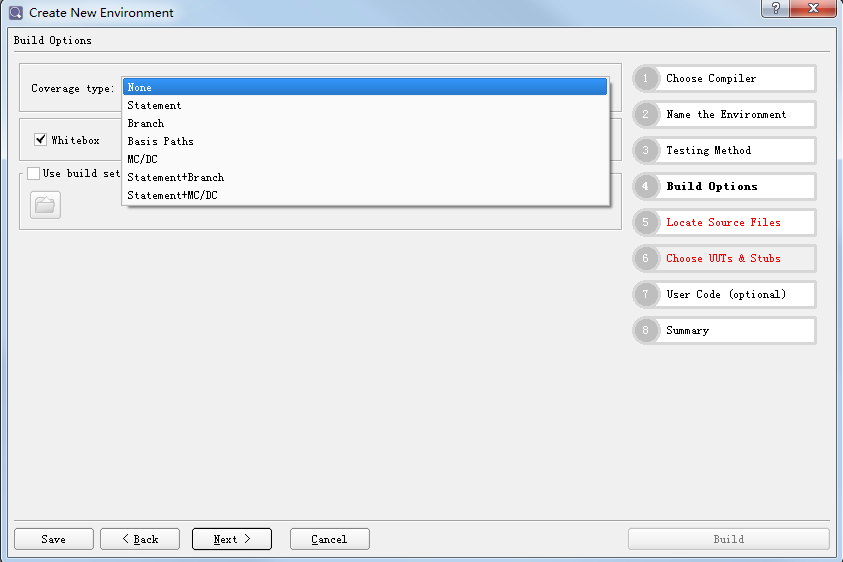


（4）**Build Options**

选择覆盖率类型：

None，Statement, Branch, Basis Paths, MC/DC, Statement + Branch,

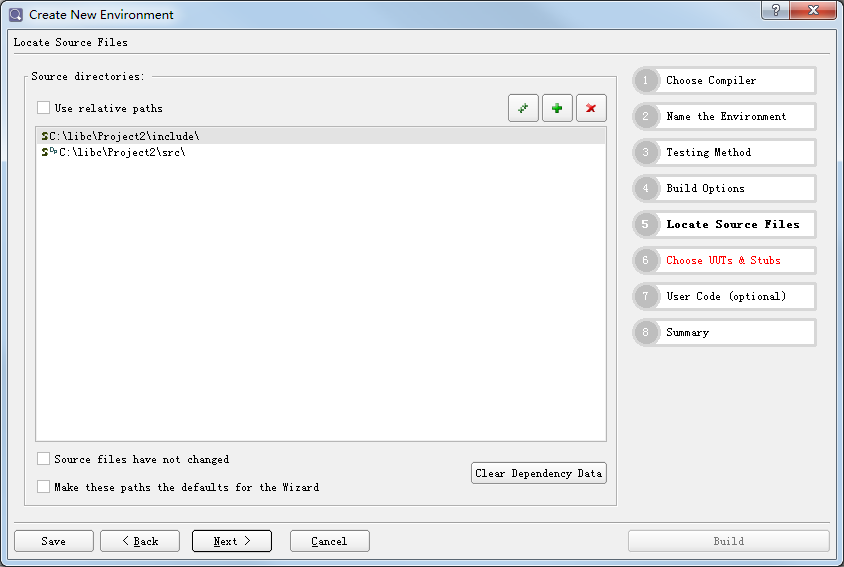
Statement + MC/DC



勾选WhiteBox，可看到调用的静态、保护和私有对象。

（5）**Locate Source Files** 定位源文件

需要包括被测对象的所有头文件和源代码

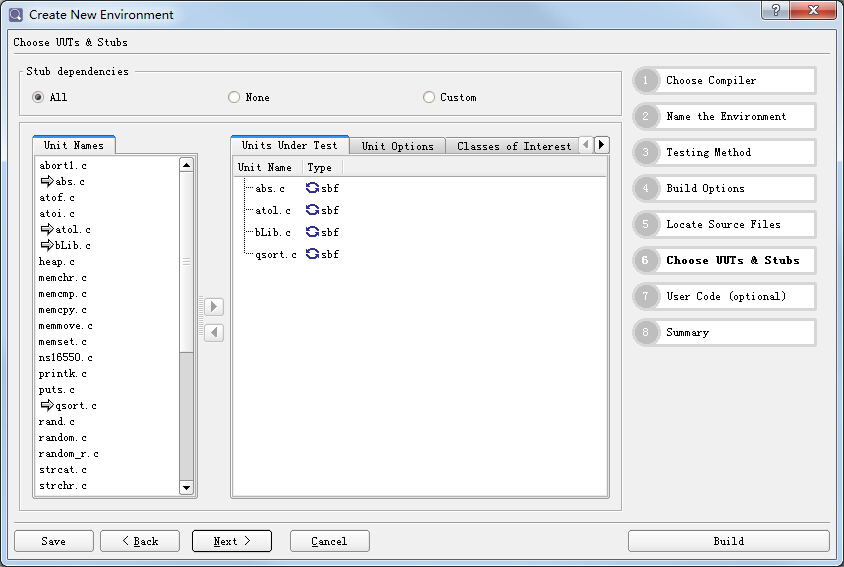


上例中C:\libc\Project2\include包括了被测对象的所有头文件

C:\libc\Project2\src 包括了被测对象的所有源码

（6）**Choose UUTS & Stubs** 选择被测对象和桩函数

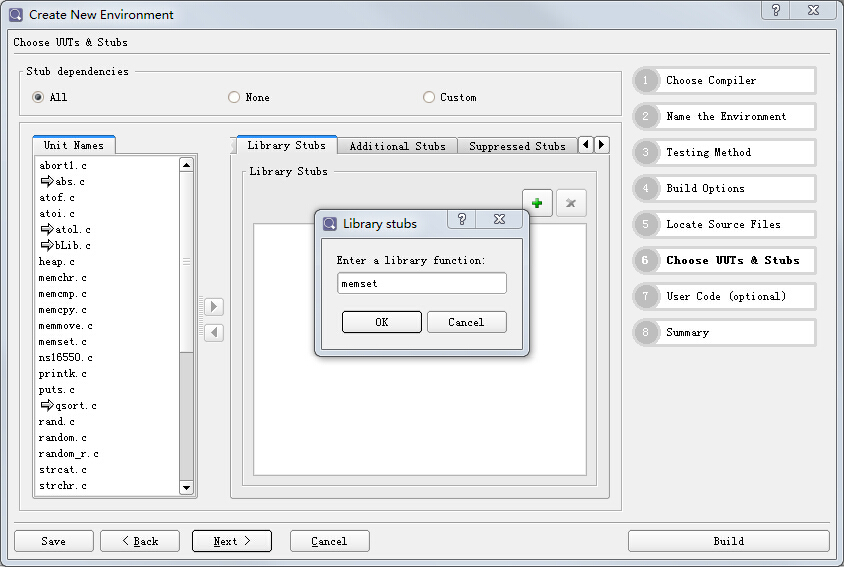
可以选择一个.C文件，也可以同时选择多个.C文件进行测试



除了Units Under Test，通过

右箭头可切换其它选项，常用的有两项：

1. **Library Stubs：**可对系统函数、库函数进行打桩



如上例中，可增加MEMSET系统库函数为桩函数.

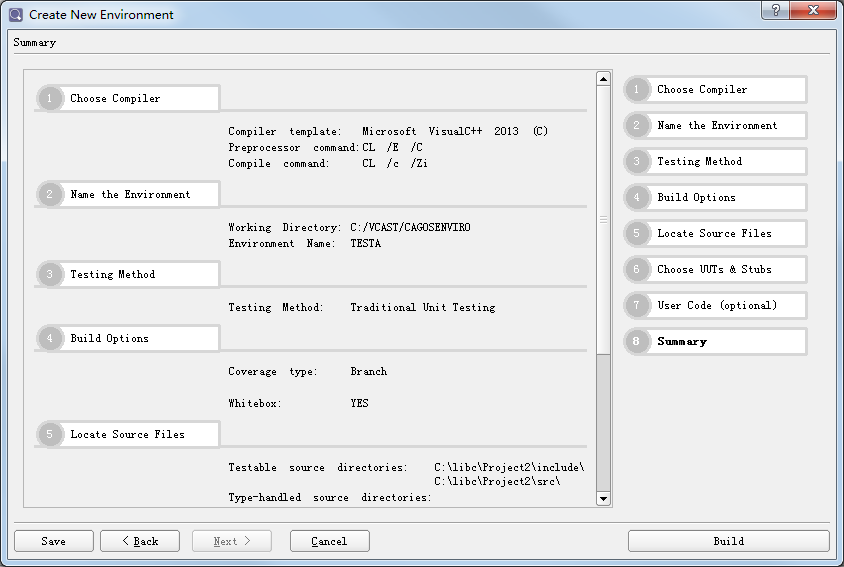
1. **Additional stubs:**

添加额外的桩（不在所测环境的桩、所添加路径里不包含的桩）

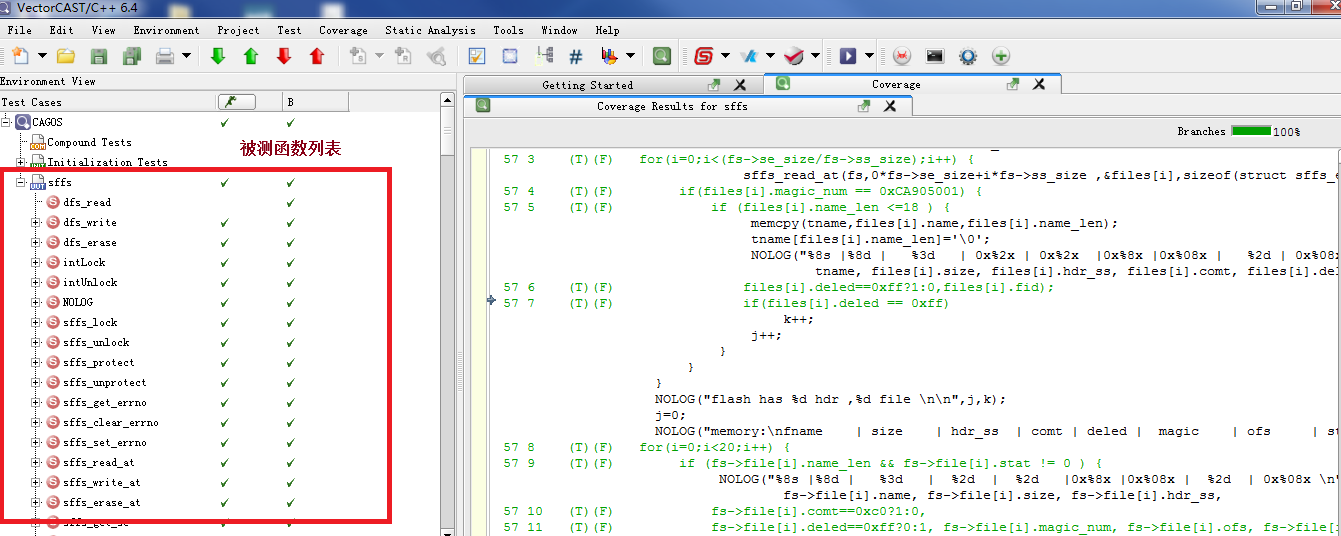
（7）**User Code（Optional）**

用户可自己定义全局变量、参数等，供打桩时使用

（8）**Summary：**总览环境的配置信息



构建好测试环境后, 测试工程的界面如下图:



VectorCast会将被测对象的所有函数列出来，可对每个函数单独测试。

1. **设计测试用例**
2. **生成测试用例**：

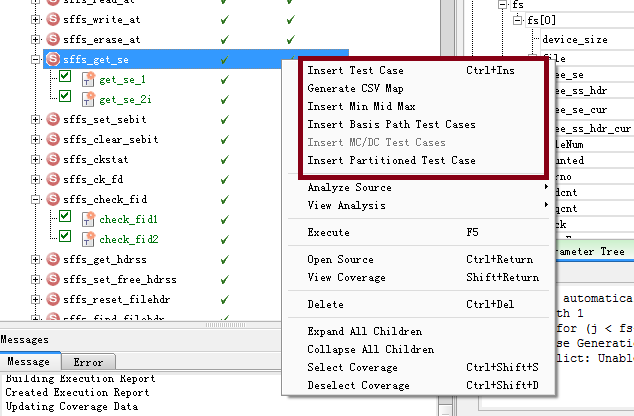
手动插入：右键->Insert Test Case

自动生成：右键->Insert Min and Max (边界值)

->Insert Basis Path Test Cases（基本路径）

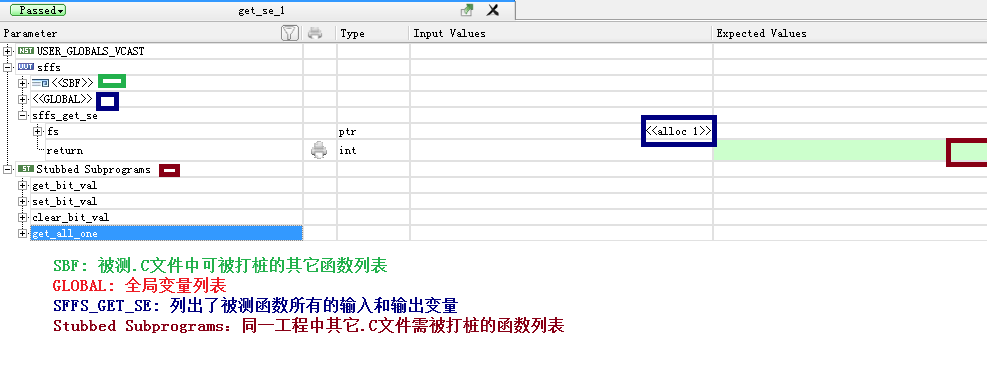
->Insert MC/DC Test Cases（MC/DC方式）

->Insert Partitioned Test Case （区域划分）

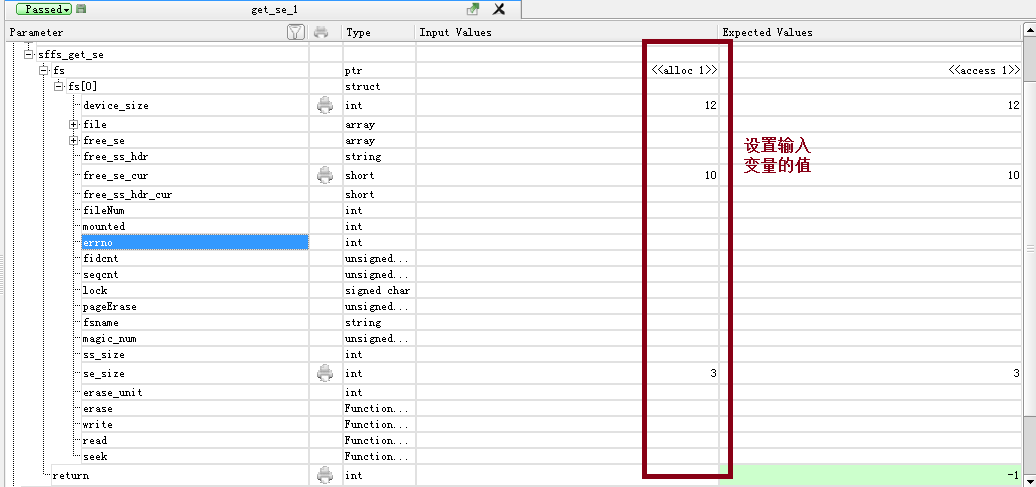


**（2） 设置测试用例的输入值和输出值**

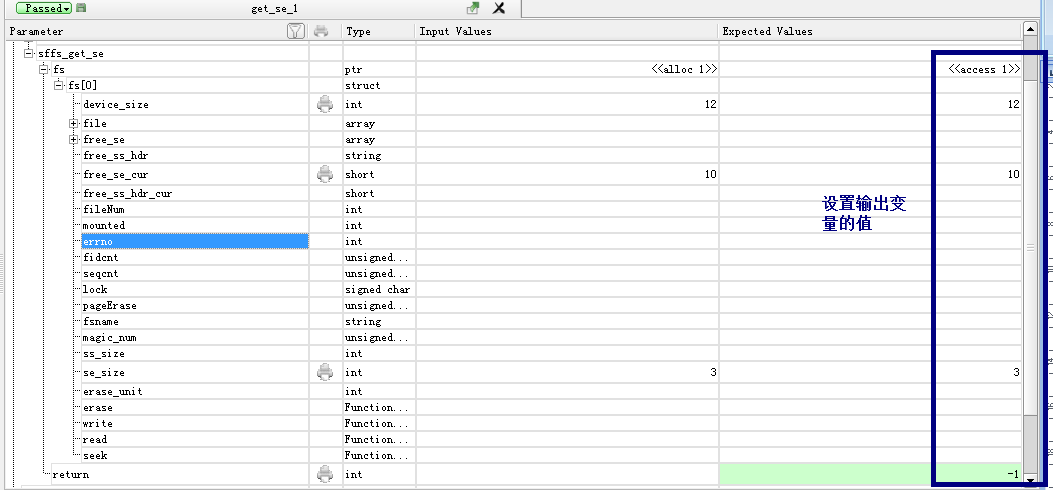
双击生成的测试用例，包含项如下：



自动生成的测试用例已对输入赋值，但也可根据自己的思路，对已生成的值修改：只需输入变量值即可。



根据输入变量的值，判断输出的值，将值填入

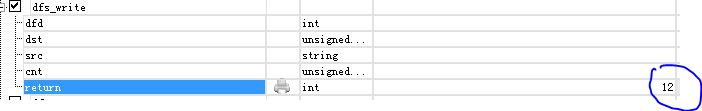


**（3）打桩**

桩函数的设置有两种方式：

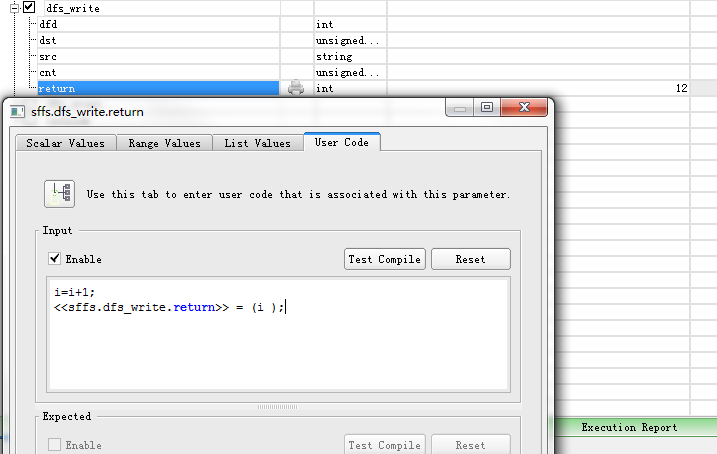
1. 直接指定桩函数的返回值

勾选需打桩的函数，直接在return处输入返回值。



1. 通过User Code写逻辑给桩函数返回值

若桩函数需在不同的情形下输出不同的值，则可对桩函数编码实现

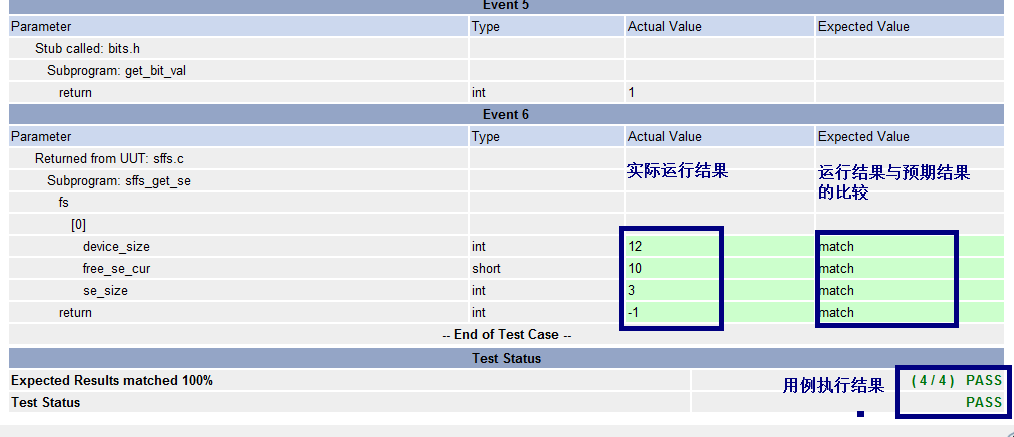


1. **测试用例的执行**

选中对象，右键Execute，测试用例就开始执行。单次执行的对象可为单条用例，或单个函数，或所有测试用例。

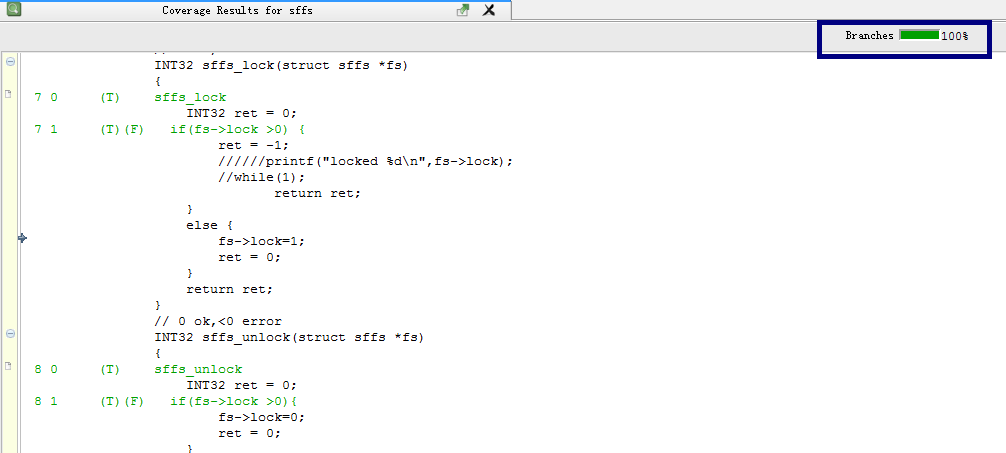
（1）查看测试结果

运行完成后测试结果如图所示：



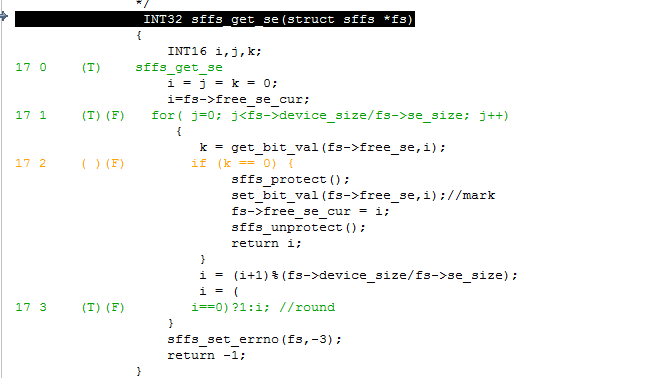
（2）查看覆盖率情况

选中被测函数，右键->View Coverage，则可以查看覆盖率情况：



（3）查看分支的执行情况

全部执行到的为绿色，部分执行到的为黄色，没有执行到的以红色显示。



以分支覆盖率为例：

每条测试用例执行后，会分析此用例对每条分支的覆盖情况，有助于测试人员对分支的分析情况。

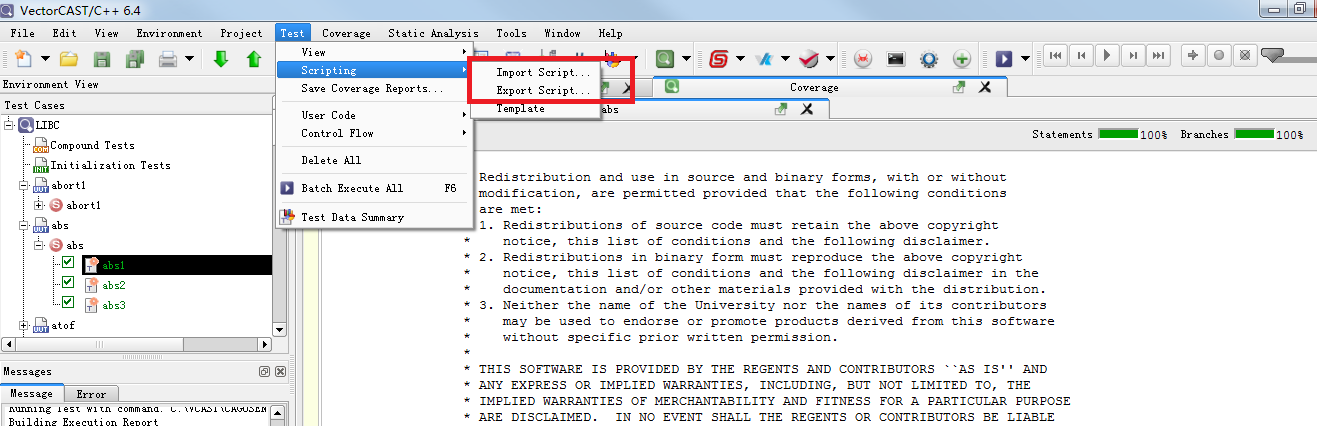
1. **测试用例结果导出：**

（1）**测试脚本的导入导出**

选中需导出的被测对象，选择Test->Scripting->Import Script 导入用例

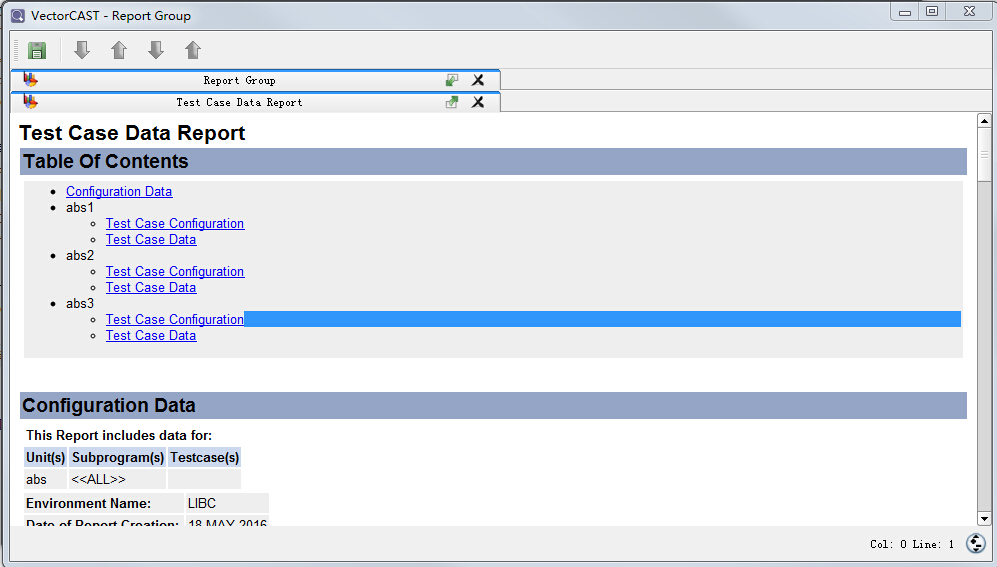
Export Script 导出用例

导入导出用例的格式为tst



（2）**测试用例数据的导出**，支持HTML和TXT格式

选中需导出用例的被测对象，选择TEST->VIEW->Test Case DATA



**（3）测试报告导出：支持HTML、TXT格式。**

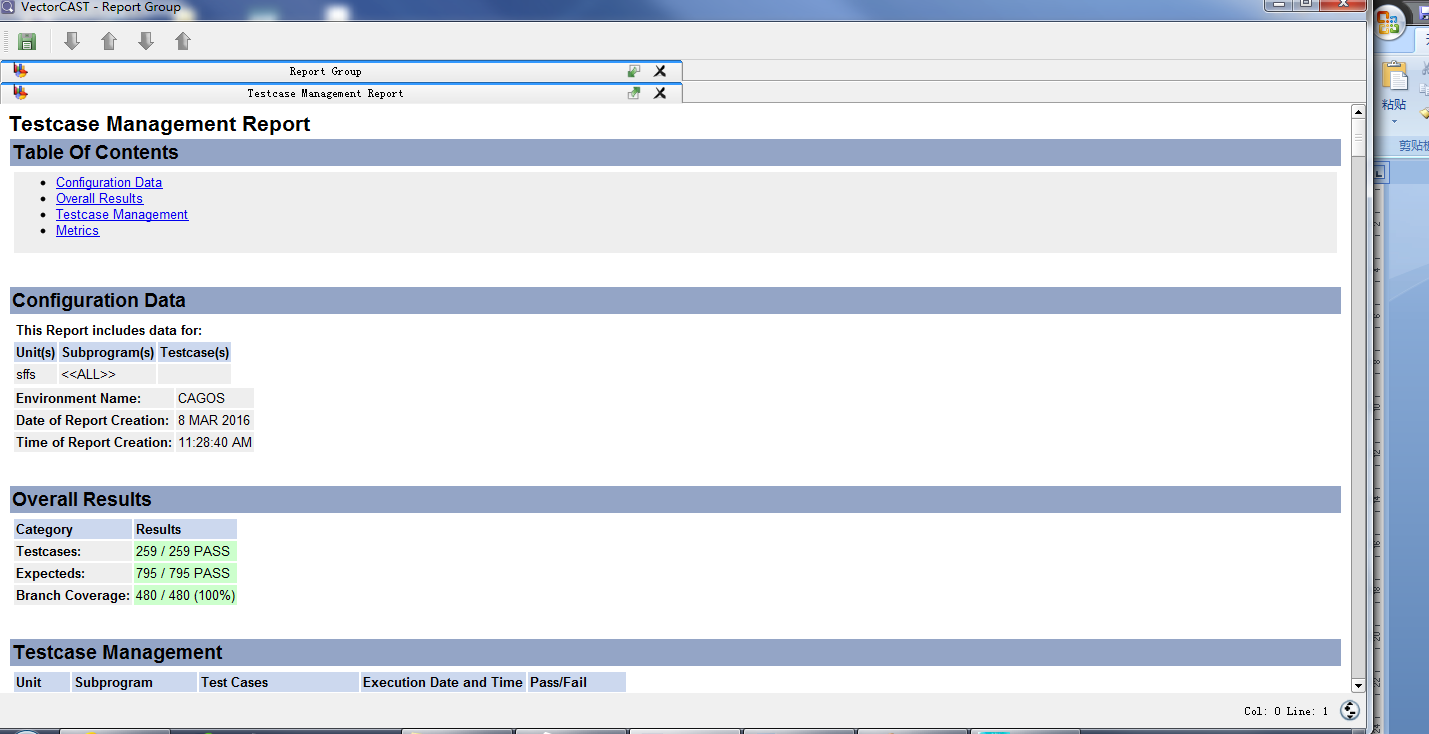
测试报告支持以下几类：

a. 测试用例总的执行结果

b. 测试用例总的覆盖率结果

c. 每个函数测试用例的执行结果

d. 每个函数覆盖率的结果



测试报告清晰明确。

1. **通过启动脚本启动VCAST**

有些编译器没有嵌套在VCAST工具里，需要通过启动脚本启动VCAST，以CCS编译器举例，在目录Workspaces/CCS4/vcast.bat下编写启动脚本如下：

set VCAST\_CCS\_INSTALL\_DIR=C:\progra~1\TexasI~1\ccsv4（安装目录路径）

set path=% VCAST\_CCS\_INSTALL\_DIR%\scripting\examples\loadti;%path%

set path=% VCAST\_CCS\_INSTALL\_DIR%\tools\compiler\tms470\bin;%windir%\system32;%path%（2、3行告诉工具要找的程序、编译器、库、头文件位置，%path%必须有否则许多程序读不到，%windir%\system32;可省略）

set DEBUGSERVER\_ROOT= C:\progra~1\TexasI~1\ccsv4\Debugserver（指定Debugserver路径）

start %VECTORCAST\_DIR%\vcastqt –lc（启动工具且设为C语言）

cmd

查看短路径命令：dir /x