

C/C++的单元/集成测试

VectorCAST/C++

版权声明：本文档版权归创提信息科技（上海）有限公司所有，并保留一切权利。

支持的主流嵌入式环境包含：

- Green Hills MULTI®
- WindRiver Tornado®
- LinuxWorks™
- TI Code Composer Studio™
- Diab SingleStep™
- Cosmic
- TASKING™
- Synopsys® ARC™
- CodeWarrior™
- Analog Devices Visual DSP++®
- ST Microelectronics®
- HighTec TriCore®
- Microchip®
- Paradigm
- Renesas™
- ARM®RVDS™
- IAR Systems®
- KEIL™
- NEC
- QNX®
- Borland®
- Mercury Computer Systems™

要点：

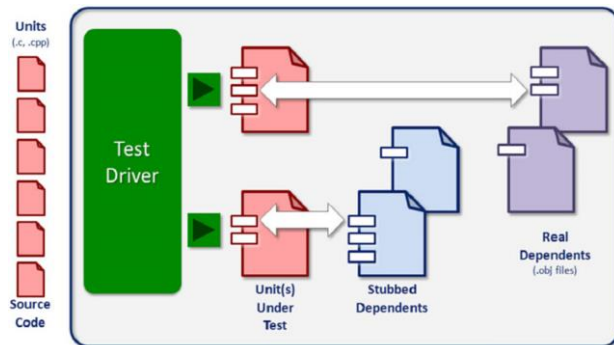
- 兼容 LINUX,UNIX,Windows 编译器
- 自动构建测试驱动和桩函数
- 集成包含 MC/DC 在内的代码覆盖率分析功能
- 支持主机，模拟器和嵌入式目标环境测试
- 自动化的回归测试
- 用户可配置编译器接口
- 支持 DO-178B,ISO26262, IEC61508, FDA, IEC62304 和 CENELEC 测试需求

什么是 VectorCAST/C++

VectorCAST/C++是一套集成的软件测试解决方案，能显著降低 C/C++测试过程中为达到安全性检测和嵌入式系统关键任务检测所必需的时间、工作量及成本。

自动化包括：

- 为单元测试和集成测试构建完整的测试环境
- 基于脚本命令或 GUI 图形界面执行测试
- 集成最好的需求管理系统和静态分析工具
- 根据基本路径来自动生成测试用例
- 根据测试需求自定义测试用例
- 回归测试
- 在调试阶段进行测试的回放
- 代码覆盖分析
- 支持敏捷开发和测试驱动开发 (TDD)



VectorCAST/C++自动构建测试组件

为什么使用 VectorCAST/C++

一般地，软件的单元测试要求为每一行被测代码生成至少一行的测试代码(以桩函数，驱动，测试数据的形式)。测试人员不仅必须编写代码，还要保证是按照预期的操作可调试的，这就是为什么编写这些用于测试的测试代码成了测试代码高成本和低效的主要原因。

通过 VectorCAST/C++，不需要编写任何一行代码就可以实现软件测试功能。

产品特点:

- 对任意复杂度的 C/C++代码自动生成完整的测试驱动和桩函数 (无需编写测试代码)
- 测试驱动支持复杂测试场景, 包括同一测试用例中连续调用不同函数
- 自动打桩能够获取输入, 控制任何预定义或者用户定义类型的输出
- 树状图形测试用例编辑器使用户方便创建和编辑测试用例
 1. 易于创建测试用例
 2. 测试静态, 保护和私有函数
 3. 构建任意复杂的类的对象
 4. 测试多态性和动态分配
 5. 抛出和捕获不同类型和值的异常
 6. 测试复杂的类的继承
 7. 单独的测试模板例程
- 捕捉和报告不期望的输出和信号
- 命令行接口允许通过脚本执行所有功能

易于使用的 GUI 界面:

- 测试用例构建---不需要编写测试代码, 参数和全局数据的值(被测单元和桩)都可以在 GUI 中定义
- 测试执行---不要求编译每一个测试用例
- 通过/失败---测试用例被执行后的结果以不同颜色显示在 GUI 中
- 代码覆盖率---对代码标注颜色进行显示, 覆盖级别包括语句、分支和 MC/DC 级别的覆盖
- 执行---能够在主机平台或者目标模拟器上运行, 也可以直接在嵌入式目标机上运行。执行平台可以通过 GUI 进行控制

VectorCAST/C++的功能

如何工作:

VectorCAST/C++首先分析您的代码, 然后调用代码生成器根据测试要求去自动构建一套完整并可执行的测试组件。一旦测试组件被成功构建, 用户可以使用 VectorCAST/C++构建和执行测试用例, 显示代码覆盖信息并生成测试报告。因为测试数据是独立于测试用例的, 可以进行自动的回归测试。

完整的代码覆盖分析

在测试过程中, 如果没有代码覆盖工具, 源代码的哪些部分被执行到是很难确定的。VectorCAST/C++提供集成的代码覆盖分析工具, 在单个或多个测试执行中, 提供关于源代码语句的报告, 为用户指明代码覆盖结果。代码覆盖度数据也可以被 VectorCAST/Cover 工具共享, 生成集单元、集成和系统测试覆盖率结果于一体的报告。

测试驱动开发

VectorCAST/C++支持敏捷开发和测试驱动开发 (TDD) 方法。设计一旦完成, 测试用例开发也就开始了。这使得用户可以在任何应用代码被开发前, 就可以构建所有的单元测试。开始阶段, 单元测试会由于缺少源代码而执行失败。但是, 随着各单元开发的推进, 源代码逐渐完善, 单元测试会逐步通过。这些单元测试用例还可以用于自动化的回归测试。

嵌入式目标环境测试

VectorCAST/C++结合 VectorCAST/RSP 使用可以支持直接在嵌入式目标系统上测试。VectorCAST/RSP 集成了交叉编译器和 RTOS, 成为测试实时应用测试的完美工具。测试用例可主机上设计, 然后在嵌入式目标环境上执行, 以验证目标机和交叉编译器的性能。

可重复测试

一旦测试用例被设计出来, 就可以使用 VectorCAST/C++自动运行测试用例对不同版本的软件进行测试。测试执行的管理和测试结果的记录都可由 VectorCAST/C++工具自动化完成。通过比较同样的测试用例在不同版本的源代码上执行的结果, 能在系统集成之前, 发现因为对代码“不经意的修改”导致的严重错误。

支持集成测试

可在一个 VectorCAST/C++测试环境中执行多个单元测试。这允许用户可以创建模拟跨单元和跨函数的复杂测试场景。

编译器集成

VectorCAST/C++支持主流的编译器，可以无缝的进行测试工作。所有的 VectorCAST/C++测试组件都是使用指定的编译器自动生成和链接的。同时也提供了和编译器的调试器的接口，以便能够在调试状态下运行测试用例。

