

C++Test 介绍

一. 概述

C++Test 是针对 C/C++ 语言的完整的自动化测试工具，它能提高软件开发团队的生产力和软件质量，它可以执行编码规则、静态分析、代码走查，并且提供动态测试和组件测试的功能，给团队提供了一个最佳实践的方式，来确保 C/C++ 代码的品质。

C++Test 自动测试任何 C/C++ 类、函数或部件，自动生成测试用例、测试驱动程序或桩调用，无需手工编写。**C++Test** 能够自动测试代码构造（白盒测试）、测试代码的功能性（黑盒测试）和维护代码的完整性（回归测试）。

同时 **C++Test** 还是一个 C/C++ 编程规范自动检查工具，它内置了 800 多条业界规则，并且可以图形化地定制自己/团队的规则。

二. 主要功能

一) 静态分析

1. 代码编程规范的自动检查：**C++Test** 内置集成了 800 多条由业界知名专家经验总结出的 C/C++ 代码编程规范；同时它还提供自动化的代码编程规范检查。

2. 代码编程规范的目录是按照以下问题的种类和严重级别（0-5 级）来分类的。包括：

- ⌘ 性能
- ⌘ 可靠性
- ⌘ 可移植性
- ⌘ 安全性
- ⌘ 可维护性

可根据以上的代码编程规范的目录组别，来启用或屏蔽规范，并使用选定的代码编程规范来分析代码。

3. **RuleWizard** 图形化定制规则：**RuleWizard** 是个图形化的代码编程规范的定制工具，同时可图形化的实现代码走查，具有修改并自定义代码规则的功能，使用户可以建立一整套自己的代码编程规范。

二) 动态测试

1. 自动执行白盒测试：确保代码中每一条独立的路径至少会执行一次，所有的循环及逻辑运算也会运行，强化代码的架构功能。

2. 自动执行黑盒测试：可快速自动建立测试用例，以检查程序运行结果是否为预期中的结果，并可自定义测试用例的值，更可有效检查错误是否存在。

3. 自动执行回归测试：可纪录测试结果，并以此建立测试用例，当重新测试的结果有所不同时，即通知使用者状况异常。

实现以上三种测试时，**C++Test** 自动建立测试环境，自动生成测试用例/测试驱动程序/桩函数，来自动分析代码。

4. 桩函数设置：**C++Test** 自动生成被测函数的桩函数。它可以按“自动生成桩函数”、“使用原函数作为桩函数”和“自定义桩函数”三种方式生成桩函数。

5. 兼容 CppUnit 测试用例：**C++Test** 可以生成 CppUnit 测试用例和图形化测试用例两种测试用例。可以把 CppUnit 测试用例导入到 **C++Test** 中使用，也可

以把 C++Test 自动生成的 CppUnit 格式测试用例输出到 CppUnit 环境下使用。

6. 导入导出测试用例：C++Test 可以导入和导出一个函数、一个文件和整个工程的测试用例。我们甚至可以保存整个 C++Test 测试工程作为整个产品的一部分（就像项目文档一样是产品的一部分）。
7. 提供覆盖率分析：
显示被测代码的各种覆盖率状况,分为:
 - ① Line Coverage：语句覆盖
 - ② Basic Block Coverage：没有包含分支，控制流转的一段代码覆盖
 - ③ Path Coverage：一条从函数入口到出口的路径覆盖
 - ④ Decision (Branch) Coverage：分支覆盖
 - ⑤ Modified Condition/Decision Coverage (MC/DC)：MC/DC 覆盖
 - ⑥ Condition Coverage：条件覆盖
8. 提供测试报告：C++Test 提供多达 20 多种测试报告的模板，可以生成汇总、统计和详细的测试报告。

三·高级特性

C++Test 的高级特性可以使用户快速地定制符合业务要求的测试用例，特别是满足特定要求的黑盒测试的测试用例。它可提供以下非常有效的功能，来大幅度提高软件测试的效率。简介如下：

1. 直接导入 VC++ 工程或者通过 makefile 文件导入一个工程。
2. Data Source：用 excel 文件，CSV 文件或数据库的表作为测试用例的输入，自动生成一系列的有规律的测试用例。
3. Test Objects：使用一些特殊或定义好的值构造一个类对象，在其他地方使用此类的时候，就可以重复使用此对象。
4. BugDetective：这是一种相对于通过代码分析和动态单元测试无法找到某些 Bug 的，全新的在执行代码路径和流路径时找到 Bug 的方法。这种技术使用程序的默认执行路径，它允许跨功能模块、跨包；而不象标准的代码走查，只把检测局限在一个函数和一个包内部。

四·版本分类

1. C++test Professional Edition：C/C++ 自动测试工具。可自动测试任何 C/C++ 类，函数或部件。内置 800 条以上业界著名的 C/C++ 代码规范，可实现自动代码编程规范检查。进行自动单元测试，包括：白盒测试/黑盒测试/回归测试。自动生成测试用例/测试驱动程序/桩函数。提供覆盖率分析和测试报告。
2. C++test Architect Edition：包括 Professional Edition 的所有功能特点，另增加 RuleWizard 功能——提供图形化的代码走查，能够让客户创建和修改代码规范标准，具有自定义代码规范功能。可建立一套用户自己的代码规范。
3. C++test Server Edition：包括 Architect Edition 的所有功能特点。同时提供命令行界面，自动构建过程中的编码标准和批处理过程，自动执行单元测试。

- 4 · 可选 **License----Security** : 静态分析源代码, 分析代码路径, 并模拟执行各路径, 找到 各种潜在的错误。

五 · 支持平台

- * Windows NT/2000/XP
- * Linux kernel 2.4 or 2.6 or higher with glibc 2.2 or higher and an x86-compatible processor
- * Linux kernel 2.6 or higher with glibc 2.3 or higher and an x86_64-compatible processor (32-bit compatibility package is required)
- * Solaris 7, 8, 9, 10 and an UltraSPARC processor

IDEs with Plug-in Support

- * Eclipse 3.1, 3.2 (32-bit)
- * Visual Studio .NET 2003 and 2005
- * Wind River Workbench 2.5+ (for embedded development)

Host Compilers

- * Windows: Microsoft Visual C++ 6.0, .NET, .NET 2003, or 2005, GCC 2.95.x, 3.2.x, 3.3.x, 3.4.x; Green Hills MULTI for Windows x86 Native v4.0.x
- * Linux (x86 processor): GCC 2.95.x, 3.2.x, 3.3.x, 3.4.x, 4.0.x, 4.1
- * Linux (x86_64 processor): GCC 3.4.x, 4.0.x, 4.1
- * Solaris: GCC 2.95.x, 3.2.x, 3.3.x, 3.4.x, 4.0.x, 4.1, Sun C++ 5.3 (Sun Forte C++ 6 Update2), Sun C++ 5.5 (Sun ONE Studio 8), Sun C++ 5.6 (Sun ONE Studio 9), Sun C++ 5.7 (Sun ONE Studio 10) Sun C++ 5.8 (Sun ONE Studio 11); Green Hills MULTI for SPARC Solaris Native v4.0.x

Target Compilers

- * Wind River GCC 3.4.x and DIAB 5.4+
- * GCC 2.95.x - 4.1 cross-compilers
- * Green Hills 4.0.x