

Cantata 8.0 有哪些新变化？

Cantata 8.0 于 2018 年 5 月发布，这一重大版本更新包括了重要的新功能和用户界面的改进，这个文档简要介绍了 8.0 中的主要变化。

介绍

Cantata 于 2018 年 5 月发布，这个重大版本升级包括了重要的新功能，包括自动测试维护、多个嵌入式目标环境下的测试重用以及更灵活的代码插装。8.0 也继续在用户界面上进行优化，在保留熟练用户的专家模式界面基础上更提供简单直观的新用户模式。

代码变更分析

软件开发生命周期尽早开展测试可以提高代码质量，减少代码返工的风险，即使计划非常好的项目依然需要在初始测试后对代码进行修改，单元测试和集成测试非常依赖于代码的结构，修改代码会带来很大的风险，修正其中的问题也很费时间，Cantata 代码变更分析这一新功能自动识别代码的改变对测试的影响，并帮助自动维护测试。针对代码的修改可以提供解决这些问题的指南和建议，并自动触发自动测试。

彻底的代码变更和测试依赖性分析可以发现：

- 哪些代码发生变化并影响现有的测试脚本
- 哪些测试脚本被代码修改影响
- 代码修改对之前的测试所实现的代码覆盖率产生什么影响

代码变更的结果在全新的代码变更管理器中显示，已经测试过的代码依然保留并与现在的代码进行对比。

源代码中所发生的修改并对测试脚本发生影响的情况包括：

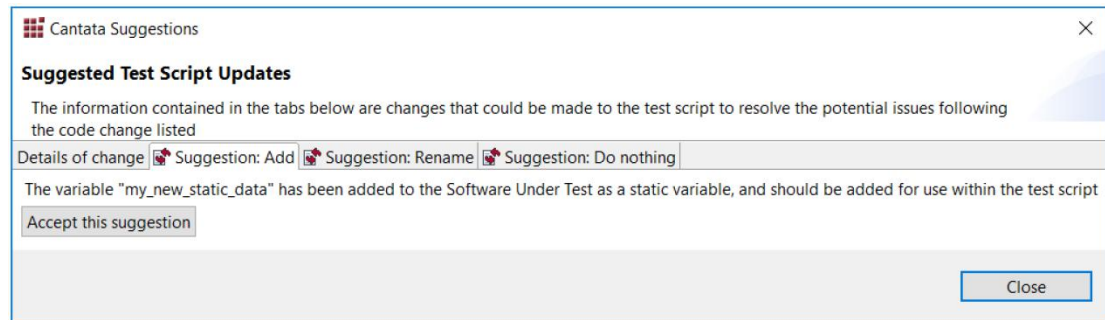
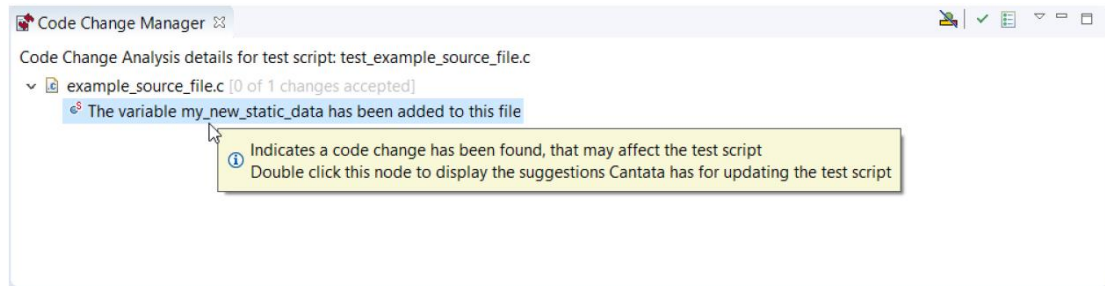
- 源文件 增加/删除/改名
- 头文件 增加/删除/改名
- 函数&方法
 - 增加/删除
 - 修改（范围，返回类型，名称或参数）
 - 函数体增加/删除
 - 函数体变更-影响代码覆盖率
- 变量

Cantata 8.0 的新变化

- 增加/删除
- 变更（范围，类型和名称）

针对代码的变更需要自动同步测试用例，提供指导性的选择：

- 关于每一个源文件的代码修改对现有测试脚本的影响的详细信息
- 对更新测试提供建议，并对于接受这个建议的影响也提供详细指南
- 测试脚本自动重构以实现建议的测试更新

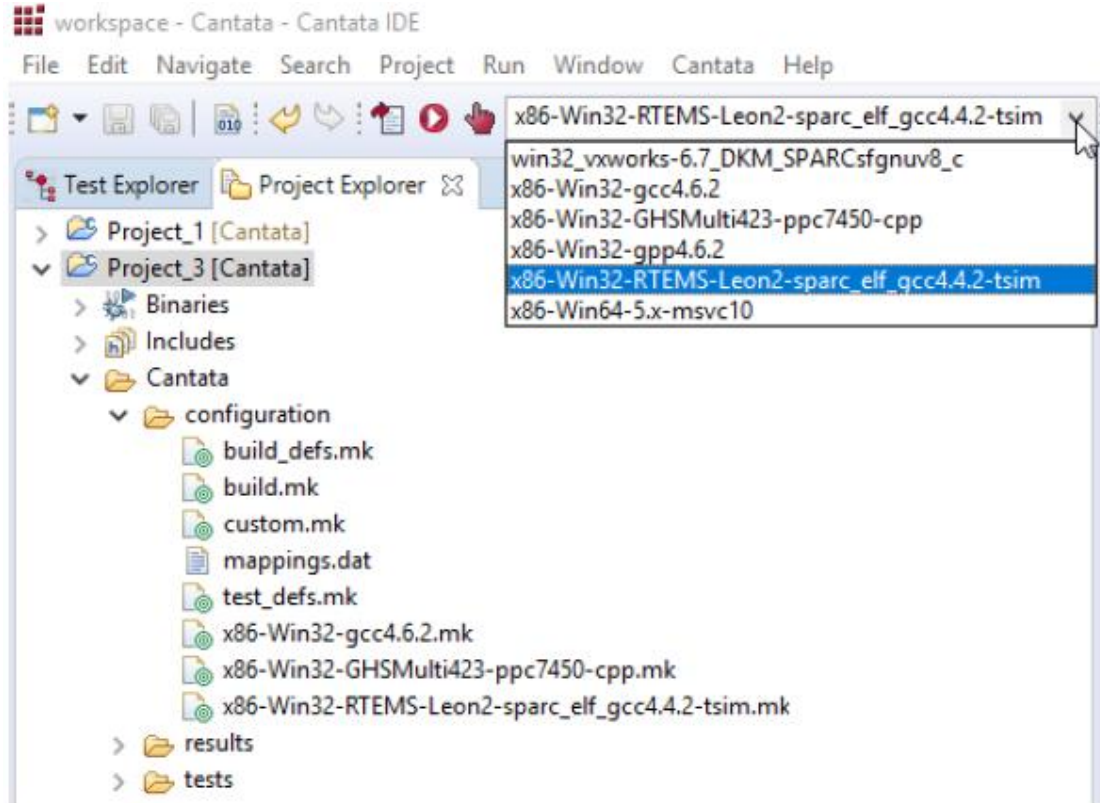


目标环境部署切换

8.0 中新的目标环境切换功能使得同样的测试在多个目标环境中重用更加轻松，特别是当代码在多个目标环境中重用时，这个功能具有很大帮助。

每一个目标环境都会产生独立的 `makefiles`，在其间进行切换很容易，就像在工具条下拉框中选择一个目标部署配置一样简单：

Cantata 8.0 的新变化



测试重用可以加快在多个嵌入式目标环境配置中的测试进度，避免在多个项目中复制，降低了测试维护的工作量，消除了维护多个项目和配置的风险。

Cantata 测试结果文件记录了每一个目标配置中执行的测试，所以对每一个环境都提供单独的测试认证相关的证据。

除了内置的目标部署配置环境之外，每当新的配置包开发完成，都可以从 QA Systems 公司下载。

测试浏览器

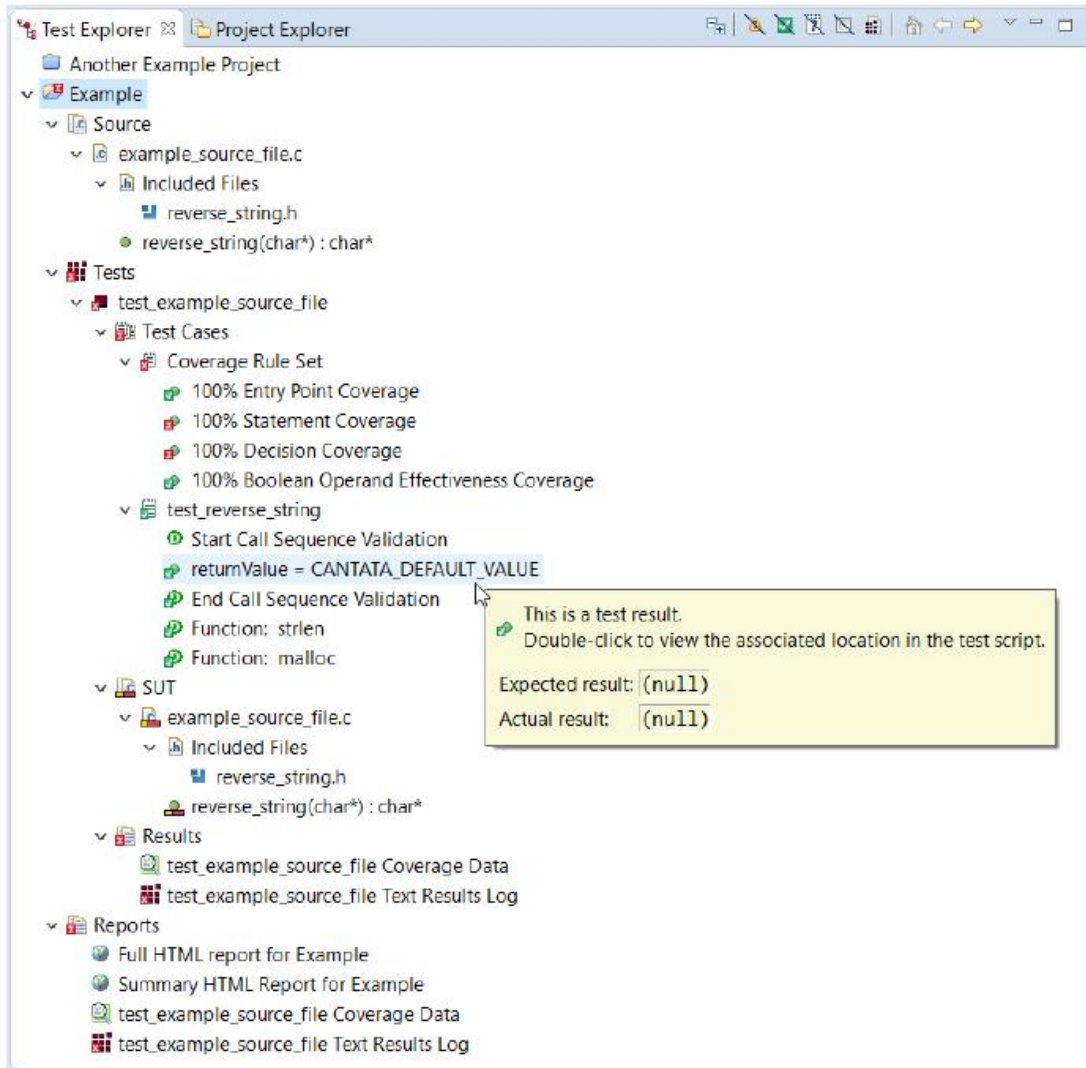
Cantata 8.0 引入了一个新的测试浏览器，更加容易查看代码、测试用例、测试结果和报告。这是一个测试的中心视图，更加清晰和容易进行导览，只显示和 Cantata 测试相关的信息。

测试浏览器提供了一个场景相关的控制集中器，对 Cantata 的功能和视图进行管理，对项目浏览的控制功能进行了结合和提升，这些诊断信息以前在覆盖率结果和测试结果浏览器中显示。

测试浏览器的新特性包括：

- 以上下文菜单直接创建和管理 Cantata 测试
- 过滤器聚焦相关源文件、测试状态和结果
- 对每一个树节点工具提示附加信息和相关的结果

Cantata 8.0 的新变化



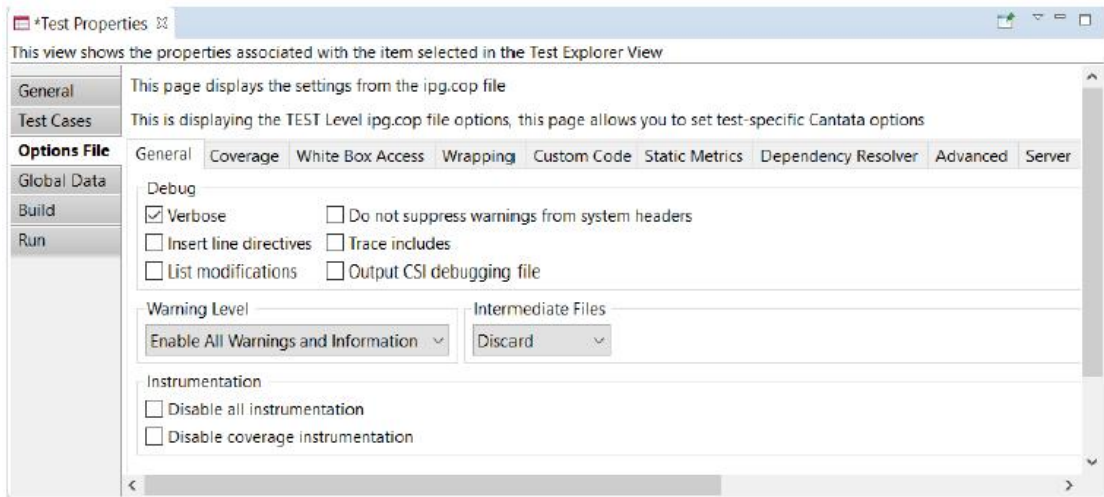
测试属性

新的测试属性视图提供了测试环境控制的单一入口，可以对测试浏览器中选中的每一个测试项设置其属性，这个视图改进了之前测试脚本管理器中的控制功能。

测试属性可以让用户管理如下内容：

- 项目和测试脚本级别选项
- 全局数据设置
- 测试用例函数名和注释描述
- 控制测试构建的设置（make file，头文件路径，代码覆盖率和静态内存）
- 测试运行配置的控制（测试用例执行的启用/关闭）

Cantata 8.0 的新变化

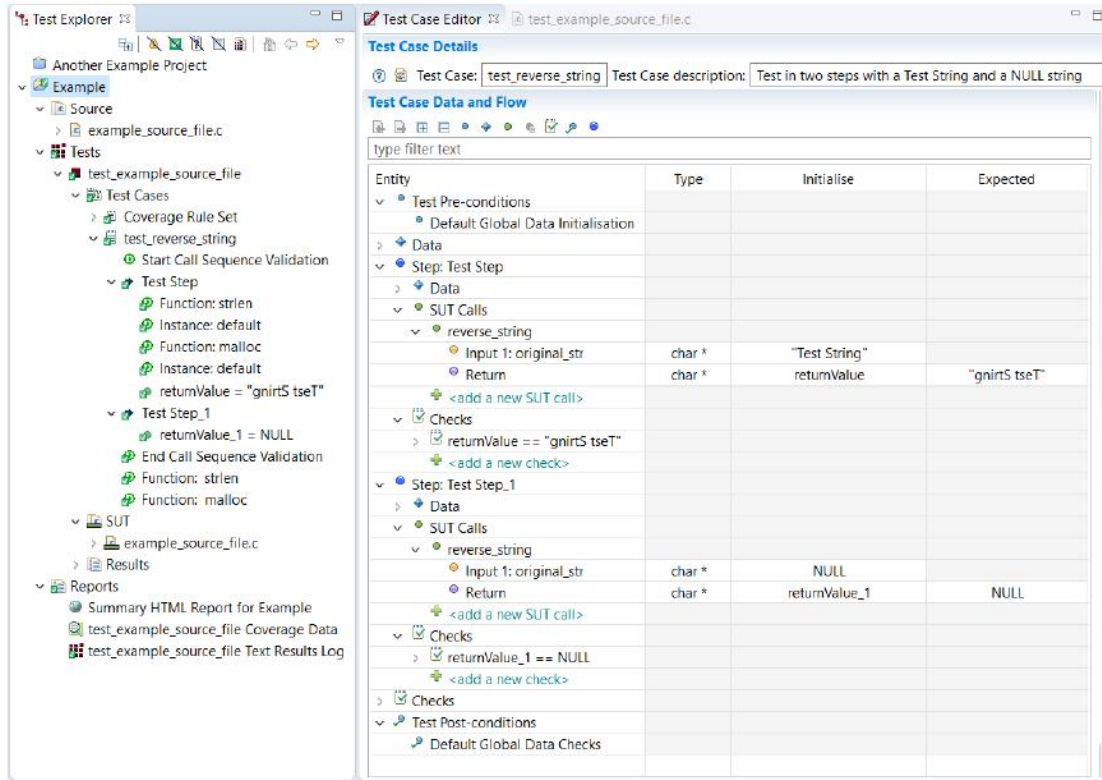


测试用例编辑器

测试用例编辑器也做了改进，新的测试步骤增加了测试用例中的更深层控制，测试用例分解为一个或多个测试步骤，每一个测试步骤具有自己的数据设置、调用被测软件和检查。缺省情况下，每一个新的测试用例产生时具有单一的测试步骤，可以进一步增加测试步骤，测试步骤的顺序可以拖拽调整。

测试用例编辑器的改进简化了测试场景的创建，因为多个测试步骤可以共享初始测试设置和最后的测试收尾。有些场景涉及到保留每一个测试步骤间的 SUT 状态，让面向对象类和内部独立函数的验证变得更加简单。

Cantata 8.0 的新变化



定制化代码插装

为了测试需要，在被测软件的任意位置插入额外代码现在可以实现了。这避免了用条件编辑的方式修改已经产品化的代码，可能会导致：

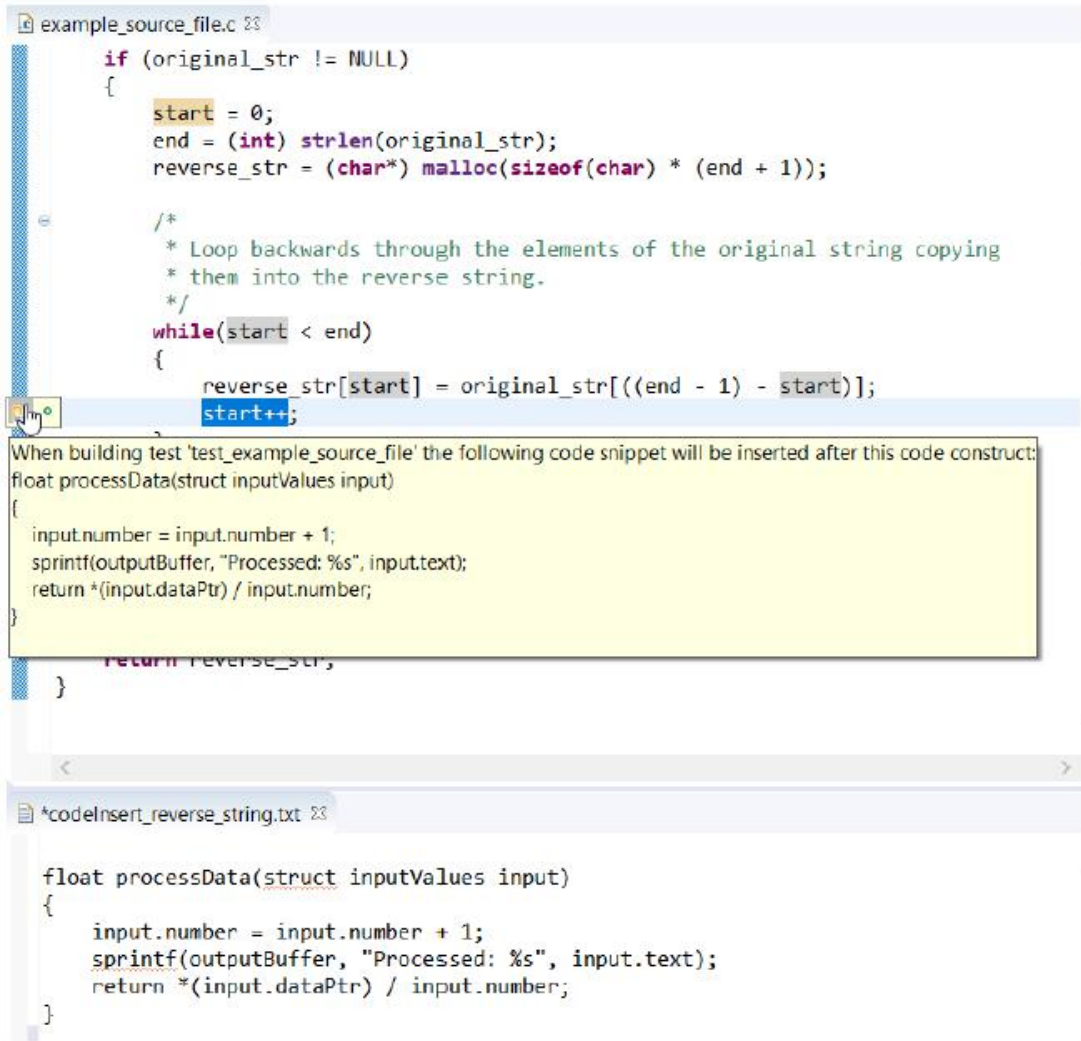
- 潜在生产环境中的意外行为
- 遗留测试代码的膨胀
- 代码的可维护性和清晰度降低

Cantata 的封装（Wrapping）提供了在函数调用时的代码修改的能力，然而定制化的代码插装可以在任何地方使用：

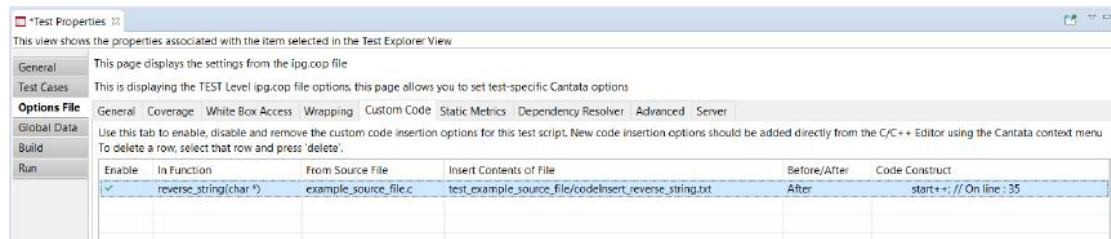
- 注入错误
- 避免使用调试跳转
- 改变局部变量的值
- 改变变量和寄存器的值
- 从无限循环中跳出
- 触发中断
- 在不修改生产代码的情况下验证和存储潜在代码修复

定制代码可以通过 C/C++ 编辑器指定具体插入的位置，代码片段可以加到任何语句之前和之后。

Cantata 8.0 的新变化



插入的代码存在单独的文件中，可以由一个或者多个测试脚本使用，通过新的测试属性工具栏和 C/C++ 编辑器来控制。



新的平台支持

Cantata 与当前基于 Eclipse 的各种集成开发环境紧密集成，包括 Eclipse Ready 插件的工具链。Cantata 基于 Neon (4.6.3) Eclipse 版本，也可以安装为 Galileo (3.5) 到 Oxygen (4.7) 的 Eclipse-Ready Plug-in。这使得支持的工具链和生态非常广泛。支持 GNU GCC 和 g++ 编译器版本升级到 7.1 (Windows) 和 7.3 (Linux)。

Cantata 8.0 的新变化

C++11&14 的支持也得到增强，支持以前在 7.2 中不支持的语言特性。