

QAC 代码静态分析软件

确保软件应用层面的质量和安全

企业级自动化源代码分析

Perforce QAC 能够快速高效地检测并报告数据流问题、软件缺陷、语言执行错误、不一致性、危险使用和编码标准违反情况。通过遵循“尽早且频繁”的理念，在创建时就检测软件缺陷，从而简化开发生命周期，降低成本和周期。QAC 提供高效、稳健和全自动化环境来引入并执行编码标准。可以创建并自定义多个缺陷检测和度量分析报告。QAC 提供监测复杂度和突出显示超过定义阈值的代码的功能，实现可测试和可维护代码的开发。

内建的C/C++/Rust分析器是QAC 源代码分析平台的核心组成部分，具备分析和报告功能，可以直接突出显示代码质量缺陷和规范的违反情况，将错误检测和最佳安全实践结合起来，充分集成到QAC 软件平台中。

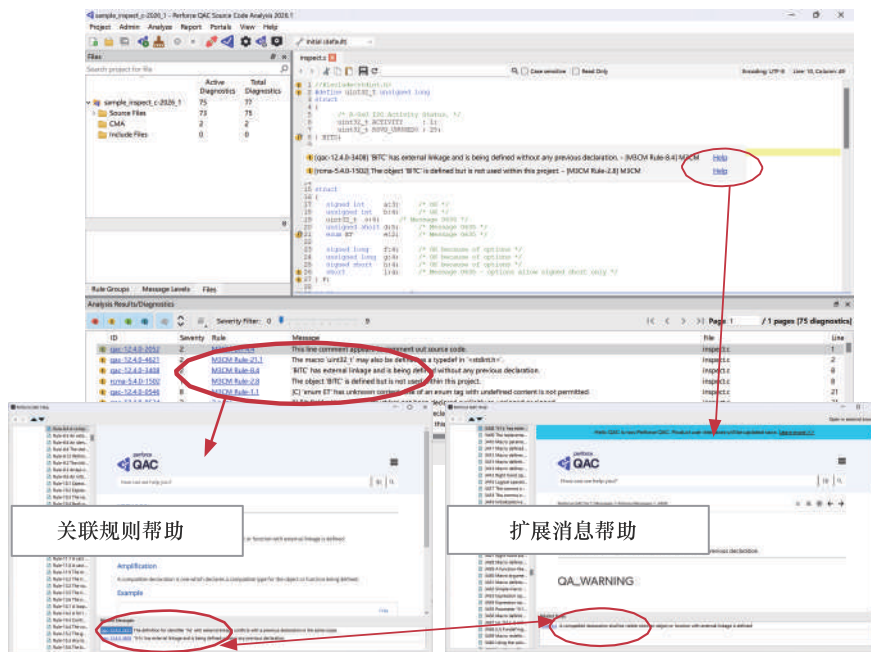
QAC检测问题所在之处，解释问题产生的原因并展示问题的修复方式

QAC 代码静态分析软件自动对您的源代码进行深入分析，而无需执行程序；可以检查您的软件的可靠性、安全性和通用的最佳编码规范的合规性；可以通过配置，在桌面或者服务器上实现自动化运行。QAC 检测编译器和大多数开发人员遗漏的问题，这些问题包括合规标准明确规定的鲜为人知的问题及代码结构，尽管不一定是一个代码错误，但可能导致无法预测的行为。

与故障捕获工具和不成熟的静态分析工具不同，QAC 可以找出更多问题，同时产生极少的误报和漏报。

优势：

- 扩展至数百万行代码
- 提高任何软件应用程序的整体质量和安全性
- 提高代码可移植性和可重用性
- 持续检测源代码对您选择的编码标准的合规性
- 向您的开发工程师提供上下文反馈，帮助他们纠正错误并从中汲取经验
- 减少手动代码审查和低效的工具及方法造成的瓶颈
- 静态分析源代码，而无需执行程序



手动代码审查不再被认为是可行的解决方案了，我们需要工具来实现该过程的自动化。我们发现QAC准确度非常高而且能够检测更多真正的问题，最终提高代码质量。”

AndrásLénárd, MűszerAutomatika Group 的资深软件工程师

我们发现25%的缺陷（.....）在早期就可以被QAC检测出来（在编码阶段）。我们的分析总结出：在开发过程后期发现的缺陷，平均需要多花2人天的时间来修复。QAC的回报期少于18个月。”

Robin Sayce-Jones, Trailer Systems at Haldex的资深软件工程师

主要特征

领先的缺陷预防功能

使用专有的高性能C/C++/Rust语言解析器以及深度数据流(Deep Flow Dataflow)分析引擎，QAC能够在运行时创建精确的软件行为模型，跟踪代码中变量的值。这个精密的分析方法能够使代码覆盖率实现最大化，同时尽量减少误报和漏报，使QAC可以检测编译器或者其它工具没有报告的致命缺陷，并识别由于使用危险、过度复杂和不可移植的语言而导致的软件问题。



检测其他人为遗漏的无法预测的行为



清楚地识别错误，
而无需执行代码

切实有效的结果

QAC清楚地展示必须修复的缺陷，包括一个基于综合型知识库的帮助系统，提供详细的指南（附有示例），帮助开发人员修复源代码中发现的问题。由于开发人员可以实时获得其开发环境中的上下文反馈，因此他们能够在创建新代码或者审查现有代码时按要求做出变动（调整）。通过这种方式，开发人员可以形成最佳实践方法的意识，能够迅速养成符合企业预期的编码习惯。

基于AI的“质量左移”

QAC将深度静态分析与AI辅助代码修复（AI-Assisted Code Remediation）相结合，通过MCP（Model Context Protocol）向AI提供缺陷信息、规则文档、修复指导及过程间数据流分析等高质量上下文，帮助生成更加精准、符合编码规范的修复建议。开发人员可直接在IDE中完成建议审阅与确认，QAC随后自动重新分析验证修复效果，形成“静态分析—智能修复—验证确认”的质量闭环，在保障开发者主导和功能安全合规的同时，有效提升缺陷修复效率，推动软件质量前移，实现更高效、更智能的软件开发流程。



基于AI自动修复代码



实时掌握您的代码
质量

集成CI/CD，持续监测并改进代码质量

QAC可无缝集成Jenkins、GitLab CI、Azure DevOps、GitHub Actions等主流CI/CD平台，通过QA-CLI实现静态分析、增量分析、质量门禁（Quality Gate）及结果上传的全流程自动化，将代码质量检查与编码规范合规验证持续融入软件开发流程。结合Validate平台，可实现缺陷跟踪、质量趋势分析和合规状态监控，帮助团队及时发现和解决问题，降低修复成本，持续提升代码质量，为功能安全及高可靠软件开发提供高效、可追溯的持续质量保障。

支持生成可配置的分析报告和合规报告，可按项目需求输出编码规范符合性、缺陷统计及质量指标等结果，为质量评审、过程审计和功能安全认证提供规范、可追溯的依据。

大规模代码的分析

使用QAC进行的自动代码静态分析可以在开发早期检测缺陷、漏洞和合规性问题，能够快速地进行修复，而且耗费的成本较低。QAC运行速度快、无干扰、使用方便，可扩展至任何大小的开发环境。因此，其产品需要在关键任务和高安全性环境中安全可靠运行的企业都相信QAC可以帮助降低软件故障风险，提高质量，缩短研发周期。

学习简单、使用方便

QAC 强大的图形化用户界面 (GUI) 提供一个链接到深层知识库的上下文环境，以此解释了其发现的问题需要得到纠正的原因，并提供修复问题的帮助指南。

先进的数据流检测

QAC通过链接到内植的、先进的跨函数跨解析单元的数据流的可满足性模理论 (SMT) 算法引擎，执行深层次的语言分析，检测缓存溢出、除数为零、无效代码、不可达代码以及许多代码质量缺陷问题。广泛的检测范围包括：变量、指针别名、双向可疑变量使用分析和回路分析之间的相互依赖性（首次、末次和中间迭代分析）。

可调整以适应现有的开发环境

QAC可以轻易地集成到现有开发系统和持续集成环境中，用自动化代码分析强化“尽早且频繁”的测试，帮助避免需要在开发后期花费巨额进行修复的错误。这可以加速目前的代码审查流程并重新确定该流程的重点，从而帮助全面提高效率，同时还提高软件的质量和安全性。此外，QAC还可以用于增量分析，确保只分析新的变化并迅速提供反馈。

稳健而灵活地实施编码标准

QAC可以通过可选模块得到补充，自动检查对主要编码标准的合规性，生成合规性认证所需的报告和审核文档。通过配置检查通用语言和项目或者具体的域问题所适用的规则，还能够强制执行自定义编码标准。QAC也可以抑制目标源代码位置的消息，对这些抑制行为进行审核，报告合规标准执行的偏离情况。

符合主要C语言标准的可用模块包括MISRA C:2004, MISRA C:2012, MISRA C:2023, Barr-C, CERT C和CWE C.

符合主要C++语言标准的可用模块包括MISRA C++:2008, MISRA C++:2023, HIC++, JSF AV C++, CERT C++, AUTOSAR C++14和CWE C++.

现代C++语言特征

对于C++语言，QAC能够解析和分析大多数现代C++语言特征，包括对语义转移(Move Semantics)、原始字符串(Raw String Literal)、数字说明符(Digit Specifier)和特殊标识符(Special Identifier) 'override' 与 'final' 的细致分析。QAC还查找使用例外(Exception)、模板(Template)、过载(Overloading)以及许多其它运用C++语言特征的区域中的设计问题和故障。

对Rust语言的全新支持

自QAC v2026起，产品新增对Rust语言的原生支持，可集成Clippy诊断，并提供跨函数、跨文件的数据流分析，以及Rust与C/C++混合项目的统一分析能力，重点覆盖关键运行时缺陷、unsafe代码与FFI边界风险，帮助企业逐步将Rust纳入现有代码质量与安全治理体系。

主要检查

避免会降低代码可重用性的、导致产品失效和功能安全事故的和出现黑客可以利用的安全漏洞的C/C++/Rust语言结构。QAC帮您避免的风险包括：

- | | |
|------------------------|----------------|
| ✖ 未定义的行为 | ✖ 移位操作 |
| ✖ 违反ISO语言限制规定 | ✖ 不变的分支操作 |
| ✖ 溢出和回绕（包括除数为零） | ✖ 对象/功能声明和定义问题 |
| ✖ 未初始化的数据 | ✖ 危险的语言应用 |
| ✖ 存储器/指针操作问题（包括空指针解引用） | ✖ 不可移植的语言运用 |
| ✖ 控制流问题 | ✖ 标识符的命名惯例 |
| ✖ 类型转换 | ✖ 违反最佳实践 |
| ✖ 冗余代码 | ✖ 污染数据的使用 |

技术规格

一般特征

- 命令行接口 (CLI)
- 带消息浏览器的交互式 GUI
- 在线帮助&知识库
 - 使用&实现
 - 上下文信息
 - C/C++/Rust语言
 - 特定的编码标准
 - 汇总&详细报告
- IDE 集成
- 支持 C++03, C++11 到 C++23

代码分析功能

- 3000+ 可选消息
- C/C++/Rust语言特定的解析引擎
- 解析任何大小和复杂度的代码
- 处理常用语言扩展
- 跨模块分析
 - (链接时间检查)
- 语义错误检测
- 交互功能和跨 TU
- 数据流错误检测
- 近似名称分析

信息输出控制

- 基于注释的抑制
- 基线

度量指标

- 基于项目: 5
- 基于文件: 35
- 基于功能: 36
- 基于C++类: 10
- 度量指标阈值警告

结果输出

- 可配置 HTML 报告
- 标准报告类型
 - 合规性
 - 代码审查
 - 抑制
 - 度量指标数据

编码规范实施

- 用户可配置的编码标准
- 补充模块
 - MISRAC: 2004 & 2012 & 2023
 - MISRAC++: 2008 & 2023
 - CERT C/C++
 - CWE C/C++
 - HIC++
 - JSF AV C++
 - AUTOSAR C++14
 - ISO TS 17961 (C Secure)
 - Barr-C
- 传统代码的规则子集
- 最佳实践问题
- 命名惯例检查程序
- 布局检查程序
- 防御式编程 - 避免缺陷
- 可扩展规则库
- 可定制的消息文本
- 偏离支持

ISO C/C++标准支持

- 全面检查 ISO C/C++限制因素
 - 未定义的行为
 - 未确定的行为
 - 由实现定义的行为

支持的主机平台

- Windows 10/11/64
- Linux RHEL7 及以上 (64位)
- CentOS 7, Ubuntu 22

IDE 集成

- Microsoft Visual Studio™ 2015, 2017, 2019 和 2022
- Visual Studio Code
- Eclipse V4.4.2以上版本和相关IDE

持续集成环境

- Jenkins, Gitlab CI/CD, Azure DevOps等
- 其它持续集成环境可以通过命令行接口集成

支持的编译器

- GNU gcc、g++
- MinGW gcc、g++
- Microsoft Visual C++
- Analog Devices Visual DSP++
- Altera Nios II gcc
- GCCARM Embedded
- ARM RVCT
- COSMIC
- Freescale Code Warrior
- Cypress Image Craft C
- eCosCentric
- Green Hills C/C++
- IAR C/C++
- Keil
- Melexis
- Panasonic
- Microchip MPLAB
- National Instruments Lab Windows
- QNX qcc
- Renesas
- SUN CC
- TASKING VX Toolset
- Texas Instruments
- Wind River Diab
- XILINX C/C++
- 也可适配其它编译器

支持的版本控制系统

- AccuRev
- Clearcase
- CVS
- Git
- Mercurial (Hg)
- MKS
- Perforce
- PVCS
- Subversion
- Synergy
- Team Foundation Server

TÜV SÜD认证

TÜV SÜD已经证实QAC“可用于安全相关软件的开发”，符合主要安全评价标准、IEC 61508, ISO 26262, EN 50128, IEC 60880和IEC 62304, 使我们的客户能够更容易地实现这些标准的产品认证，且耗时较少。



欲知更多 QAC 产品和技术相关信息，欢迎通过电话：+86-021-68580866 或 e-mail:info@trinitytec.net联系我们，我们将提供详尽的技术支持和服务。

公司简介:

创提信息科技（上海）有限公司 Trinity Technologies

成立于2012年，专注于高可靠性和高安全性嵌入式软件的开发过程管理，自动化验证和产品网络安全的最佳实践方案。凭借全球领先的软硬件工具链产品和专业的技术能力，长期服务于国内汽车、航空航天、国防军工、轨道交通、工业自动化和医疗器械等关键行业众多客户，助力研发团队完善研发流程并改进效率，尤其是快速地满足功能安全、网络安全和适航等研发标准的合规性要求。