USTCHCS 程序分析工具集

VS 版用户使用手册

（ver3.0）

安徽中科国创高可信软件有限公司
苏州研发中心
2019-11

表 1：文件修改历史

版本	修改描述	日期
2.5	首次发行	2019.05.13
2.8	1、第一章主要是工具的基本介绍以及对各个规则的展开介绍；1.1 工具介绍添加了工具集统计一览表，用于对各个规则集做简洁的介绍，使用户明确用哪个规则集比较合适； 2、第二章是工具的操作，2.1 工具安装步骤中，删除语言包的安装；2.2.1 中 2 通用选项配置的表格、规则集通用配置的表格有待确认；2.2.1 按照使用流程编写，原有的 misra2012 规则集操作步骤删掉了，只留 Bugfinder 规则集的操作步骤作为例子；2.2.4 检测结果说明也是以 bugfinder 为例； 3、完善了 3.1 使用限制中规则禁用的说明； 4、把原来的文档中所有的字体换成了宋体;删除了没有内容的章节，如：3.3 注意事项，3.4 网络使用说明，4.3 备份和恢复。	2019.08.20
3.0	1、底层分析工具增加对 MISRA C2012 Dir 规则组的支持(原规则前加 R)、新增 MISRA C:2012、MISRA C++:2008 若干标准规则支持和若干规则，支持编译诊断信息存入分析数据库、底层编译框架升级为 LLVM 9.0； 2、将 2.2 工具运行流程的 bugfinder 规则集操作演示实例替换为 bestpractice 规则集； 3、第二章 2.2.1 配置规则选项步骤中，新增“编译后运行选项”；2.2.3 编译配置中新增“编译语	2019.11.7

	言标准与规则集语言标准冲突时”选项；2.2.6 中修改了 Toolbar 的添加或移除按钮的下拉选项； 4、补充了 2.2.10 自动修复功能的说明、补充 2.2.11 清除功能中的选项中的清除方式； 5、更新了文档中与版本不符的软件界面截图，并将文档中的目录字体改为宋体，表格中的字体改为微软雅黑； 6、补充了插件和分析工具的更新日志； 7、在 3.3 工具异常处理中增加了 Toolbar 未自动显示、编译语言标准与规则集语言标准冲突时的解决方案； 8、在 2.1.5 中补充了跳转告警信息的方式；在 2.1.6 中增加了分析完成后提示音功能；增加了 2.1.3.1 过滤功能的具体使用方式，2.1.13 分析结果导出功能。	

目录

1. 工具概述

1.1 工具介绍

USTCHCS 程序分析工具集是一个包含了规则检测、代码度量、潜在缺陷分析、最佳实践分析及符号执行等功能模块的一组工具集，通过运行本程序分析工具集，可以检测软件代码是否违反预定的编写规范，是否存在运行时错误及潜在的风险，可以帮助用户评估和提高软件代码质量。工具集统计一览表：

表 2：工具集统计一览表

工具集名称	简介	可用规则数
代码度量检测	C/C++程序的一组软件度量值检测，用于衡量代码质量，可读性以及可维护性。	—
Bestpractice 最佳实践分析	C/C++程序的最佳实践规范符合性检测。	90
Bugfinder 缺陷分析	C/C++程序的通用缺陷代码检测。	96
GJB 5369-2005 对标检测	航天型号软件 C 语言安全子集，主要用于航天行业。	136
MISRA C:2012 对标检测	MISRA C 一开始主要是针对汽车产业，现也适于其他产业。	115
MISRA C++:2008 对标检测	MISRA C++是由汽车产业软件可靠性协会（MISRA）提出的 C++语言开发标准。其目的是在增进重要系统中的软件安全性定义了表达式以及运算符相关的规范。	192
SJ/T 11682-2017 对标检测	在 C/C++语言的基础上，结合 C/C++语言源代码的开发和测试实践以及 C/C++语言源代码中常见的缺陷而制定。	76

⚠*注：*

极个别规则由于实现技术上存在局限未完成或存在相应误报漏报，上表中工具集可用规则数为实际可用规则数，存在一些被禁用的规则详见第三部分第二小节《规则集禁用规则说明》；

工具集中各个工具单独授权，只有获取对应的授权 key 并添加注册后才能正常使用。

1.2 标准符合检测

标准符合检测（也称为规则检测）模块是基于静态代码分析（部分应用符号执行技术），从语法和语义层面检测代码是否符合预期规则的编写规范，旨在从代码编写层面杜绝危险的编程风格、语言上为定义的程序行为等等。

目前，市场上成熟的规则包括：MISRA C/C++规则、CERT C/C++，AUTOSRAR C++14，CWE C/C++， GJB 航空航天 C 语言编程规范，GBT 嵌入式 C 程序编程规范等。

本规则检测模块已实现代码度量检测、Bestpractice 最佳实践分析、Bugfinder 缺陷分析、GJB 5369-2005 对标检测、MISRA C:2012 对标检测、MISRA C++:2008 对标检测、SJ/T 11682-2017 对标检测等规则检测。

1.3 Bestpractice 最佳实践分析

Bestpractice 最佳实践，是中科国创高可信软件有限公司根据 C/C++标准，国际标准以及编程规范等，由分析团队开发的 C/C++程序的最佳实践规范符合性分析工具。

表 3：Bestpractice 最佳实践分析规则统计

类别	简介	规则数量	实现数量
头文件	C/C++头文件中存在的不良代码。	4	4
可读性	会造成代码可读性问题的不良代码，包括：函数体过大，嵌套层次过深等规则。	13	13
注释	注释中存在的不良代码，包括注释风格和注释代码的情况。	3	3
命名空间	命名空间可能引入的不良代码分析	2	2
命名问题	定义了在对对象（函数、变量等）的命名需要遵守的最佳实践。	12	12
函数(含方法)	定义了函数级成员方法应遵守的最佳实践。	6	5
表达式与语句	定义了表达式和语句中，关于副作用，控制流等相关的最佳实践。	15	14
switch 语句	定义了 switch 语句应遵守的最佳实践。	5	5
类型相关	定义了语言中类型相关的代码应遵循的最佳实践。	12	12
类	定义了 C++中类相关的代码应遵循的最佳实践	9	9
包含文件	定义了头文件包含 include 需要遵循的最佳实践。	4	1
库函数	定义了调用库函数时应遵循的最佳实践。	2	2
资源相关	定义了代码使用内存等资源时应遵循的最佳实践。	3	3
不完整代码	定义了代码函数完整性应遵循的最佳实践。	5	5

⚠*注：*

由于 Bestpractice 最佳实践中有一小部分编程规则是是 Undecidable 的或者基于语义的（如除零错等），这部分规则通过语法扫描是无法支持或仅能部分支持（存在漏报或者误报情况），因此本规则检测模块没有完全实现所有编程规范，其中部分未实现规范通过其他分析模块补充实现，以下规则集均存在相同的情况。

1.4 Bugfinder 缺陷分析

Bugfinder 潜在缺陷分析工具，是中科国创高可信软件有限公司根据 C/C++现有已被广泛认可的国际标准(如 MISRA C/C++)，编程规范(如 Google coding style,Linux kernel coding style)，编程指导原则(如 Clean Code)，常见缺陷等，由分析团队开发的 C、C++程序的常见通用缺陷分析工具。Bugfinder 通过基于静态代码分析，结合自动定理证明器，在确保高效的代码分析效率的同时，更精准的找到代码中潜在的缺陷及高风险代码。

表 4：Bugfinder 缺陷分析规则统计

类别	简介	规则数量	实现数量
一般	C++语言通用的缺陷情况。	2	1
拷贝粘贴代码	定了了用户在通过复制、拷贝代码时容易出现的缺陷情况。	12	12
变量与作用域	定义了变量及变量作用域相关的潜在缺陷。	2	2
指针与数组	定义了用户在使用指针和数组时容易出现的缺陷情况。	11	11
表达式	定义了表达式以及运算符相关的缺陷情况。	16	15
常量	定义了常量相关的缺陷情况。	3	3
一般语句	定义了一般语句容易引入的缺陷情况。	5	5
If 语句	定义了 if 语句容易引入的缺陷情况。	4	4
Swtich 语句	定义了 switch 语句容易引入的缺陷情况。	6	6
循环	定义了循环语句容易引入的缺陷情况。	7	6
函数	定义了函数中容易引入的缺陷情况。	6	5
类与异常	定义了类和异常处理容易引入的缺陷情况。	4	4
库函数	定义了库函数中容易引入的缺陷情况。	19	19
类型与宏定义	定义了宏定义中容易引入的缺陷情况。	2	2
注释	定义了注释中容易引入的缺陷情况。	1	1

1.5 GJB5369-2005 介绍

GJB5369-2005 航天型号软件 C 语言安全子集是参照了 MISRA（Motor Industry Software Reliability Association）1998 年的《Guidelines For The Use Of The C Language In Vehicle Based Software》和 LDRA（Liverpool Data Research Association）2000 年的《MISRA C Checking》，并结合航天型号软件的特点经过裁剪和补充而形成的。

GJB5369-2005 的规则包含 15 大类共计 138 个规则，其中推荐规则为 40 个，强制规则为 98 个，具体的规则实现情况参见下表（表 5）。

表 5: GJB5369-2005 规则统计

类别	简介	规则数量	实现数量
声明定义类	规定了 C 程序在声明和定义中需要遵守的编程准则	31	29
版面书写类	规定了 C 程序的代码格式，需要遵守的编程准则	12	12
分支控制类	规定了编写 if 和 swtich 语句需要遵守的编程准则	8	8
指针使用类	规定了使用指针时需要遵守的编程准则	5	5
跳转控制类	规定了程序中应避免使用 longjmp/goto 等跳转语句。	3	3
运算处理类	规定了在处理运算时需要遵守的编程准则	22	22
过程调用类	规定了编写函数时需要遵守的编程准则。	13	13
语句使用类	规定了编写语句时，需要遵守的编程准则	11	11
调用返回类	规定了函数的返回语句需要遵守的编程准则	5	5
程序注释类	规定了 C 程序中的注释需要遵守的编程准则	3	3
循环控制类	规定了在使用循环时需要遵守的编程准则	5	5
类型转换类	规定了在使用类型转换时需要遵守的编程准则	4	4
初始化类	规定了结构体，枚举和变量的初始化编程准则	4	4
比较判断类	规定了逻辑表达式，判等表达式应遵守的编程准则	4	4
名称、符号与变量使用类	规定了对象的名称应遵守的编程准则	8	8

1.6 MISRA C:2012 介绍

MISRA 全称为汽车工业软件可靠性协会， MISCRA C 是该协会定义的用于规避风险，提高 C 语言软件质量的一套 C 编程语言的子集，主要应用于嵌入式系统设计。如果在嵌入式系统的 C 语言应用开发中使用了该子集，那么许多 C 编程语言中存在的缺陷就能被规避了。

MISRA C 从最初发行的 1998 版，到 2004 版及最新的 2012 版，经过了 3 个版本的更新，规则也从最初的 127 个规则更新为 2012 版的 22 大类 146 个规则，具体的类别简介及规则实现情况见下表（表 6）。

表 6：MISRA C：2012 规则统计

类别	简介	规则数量	实现数量
开发环境	标准的 C 语言环境	3	0
未使用代码	程序中不应该存在没有用到的代码，包括声明，宏定义，函数参数等，也不应该包括死代码（即无效的代码）等需要遵守的规范。	7	3
注释	注释不应该存在//和、/*混用的情况，也不应该在//的行尾出现行拼接符\等需要遵守的规范。	2	2
字符集和词法管惯例	定义了八进制，十六进制的使用规范，三元符不能使用等需要遵守的规范。	5	5
标识符	定义了标识符相关规范，包括变量名、宏名称、函数名等的唯一性等需要遵守的规范。	9	5
类型	定义了位域使用时需要遵守的规范。	2	2
字面值与常量	定义了八进制常量不应被使用，无符号整数常量需要带 u 后缀等需要遵守的规范。	4	4
声明与定义	定义了变量类型、函数类型、变量和函数的声明、数组、枚举及指针类型等使用时需要遵守的规范	14	9
初始化	定义了变量、联合体，数组类型在初始化时需要遵守的规范。	5	5
基本类型模型	定义了在运算、赋值、比较及类型转换时，对于操作数的基础类型需要遵守的规范。	8	8
指针类型转换	定义了在使用指针类型进行类型转换时，需要遵守的规范。	9	9
表达式	定义了表达式的运算符优先规则、移位运算、无符号数运算时需要遵守的规范。	4	4

副作用	定义了在初始化器列表、表达式、赋值操作、逻辑运算、sizeof 运算中关于副作用需要遵守的规范。	6	5
控制语句表达式	定义了关于控制语句（包括 while，do while，for 语句）在使用时需要遵守的规范。	4	4
控制流	定义了关于 goto、label、break、return 以及 if 语句等在使用时需要遵守的规范。	7	7
Switch 语句	定义了 Switch 语句在使用时需要遵守的规范。	7	7
函数	定义了函数在使用时需要遵守的规范。	8	7
指针和数组	定义了指针在数学运算、减法以及比较时需要遵守的规范，以及数组在定义时需要明确数组长度等需要遵守的规范。	8	4
重叠存储	定义了交叉存储对象在使用时需要遵守的规范。	2	2
预处理指令	定义了预处理指令在使用时需要遵守的规范。	14	11
标准库	定义了在使用标准库时需要遵守的规范。	12	12
资源	定义了程序在使用资源时需要遵守的规范。	6	0

1.7 MISRA C++:2008 介绍

MISRA C++是 MISRA 协会定义的用于规避风险，提高 C++语言软件质量的一套 C++编程语言的子集，主要针对 C++语言，应用于苛刻性系统设计。MISRA C++：2008 子集集成了现有的 C++指导标准，并且显著的提升为最好的标准。 MISRA C++的规则包含 20 大类，70 个子类，共 228 个规则，具体的类别简介及规则实现情况见下表（表 7）。

表 7：MISRA C++：2008 规则统计

类别	简介	规则数量	实现数量
语言无关问题	列举了与语言无关的问题，如没有用到的代码、变量、声明，死代码等。	18	11
一般问题	C++语言通用规范。	3	1
词法惯例	定义了八进制，十六进制的使用规范，三元符不能使用，标识符定义等需要遵守的规范。	17	14
基本概念	定义了声明，定义等规范，以及一个定义原则；名字查找，类型相关的规范。	13	8
标准转换	定义了整型提升和指针转换相关的规范。	5	5
表达式	定义了表达式以及运算符相关的规范。	42	40
语句	定义了表达式语句、复合语句、选择语句、迭代语句以及跳转语句相关的规范。	23	23
声明	定义了限定符、枚举、命名空间、asm 声明和连接规范相关的规范。	16	14
声明子	定义了声明子、函数定义和初始化相关的规范。	9	9
类	定义了成员函数、联合体、位域相关的规范。	8	8
派生类	定义了多基类、成员名称查找和虚函数相关的规范。	7	5
成员访问控制	定义了成员访问控制相关的规范。	1	1
特殊成员函数	定义了构造函数和拷贝类对象相关的规范。	5	5
模板	定义了模板声明、名称解析、模板实例化和函数模板定制相关的规范。	10	3
异常处理	定义了条件包含、源代码包含、宏替换和编译指令相关的规范	17	14
预处理指令	定义了 Switch 语句在使用时需要遵守的规范。	19	16
标准库	定义了标准库使用时需要遵守的规范。	5	5
语言支持库	定义了语言支持库使用时需要遵守的规范。	8	8
诊断库	定义了使用诊断库时需要遵守的规范。	1	1
输入/输出库	定义了使用输入输出库时需要遵守的规范。	1	1

1.8 SJ/T 11682-2017:C/C++语言源代码缺陷控制与测试规范

SJ/T 11682-2017 标准是在理解 C/C++语言的基础上，结合 C/C++语言源代码的开发和测试实践以及 C/C++语言源代码中常见的缺陷而制定的，C 语言语法遵循 ISO/IEC 9899:2011 标准，C++语言语法遵循 ISO/IEC 14882:2011 标准。

表 8：SJ/T11682-2017 规则统计

类别	简介	规则数量	实现数量
行为问题	在类型转换、赋值、运算等相关行为问题时需要遵循的规范。	17	17
数据处理	在处理数据时需要遵循的规范。	23	13
错误的 API 协议实现	在调用 API 函数时需要遵循的规范。	5	5
劣质代码	在编码中为避免劣质代码而需要遵循的规范。	33	33
错误的初始化与清除	内存的初始化和清理时需要遵循的规范。	5	5
指针问题	指针操作时需要遵循的规范。	3	3
时间与状态	在多线程及递归时需要遵循的规范。	2	0

2. 操作步骤

2.1 软硬件安装

2.1.1 硬件环境

安装环境硬件基本要求：

- Intel/AMD 双核 1.8G 以上 CPU，4G 以上内存，10G 以上磁盘空间

推荐硬件配置：

- Intel/AMD 四核 2.5G 以上 CPU，16G 及以上内存，64G 以上磁盘空间（固态硬盘）

2.1.2 软件环境

安装环境软件要求：

- Windows 7 以上 64 位操作系统

依赖软件包：

表 9：依赖软件包

软件名称	版本
Git	12.16 及以上
MS VC++ x86 运行库	2015 及以上
Visual Studio	2013、2015、2017

⚠ *注：*

如果安装的 Visual Studio 版本为 2015 或 2017，则不需要单独安装 MS VC++ x86 运行库了。

2.1.3 Visual Studio IDE 安装

安装步骤（截屏以 Windows 7，Visual Studio 2013 为例（以下简称 VS2013））：

1）运行 VS2013，选择【工具】菜单中的【选项】；

图 1

2）在选项对话框中的左侧列表中，选择【环境】下的【扩展和更新】，单击【添加】；

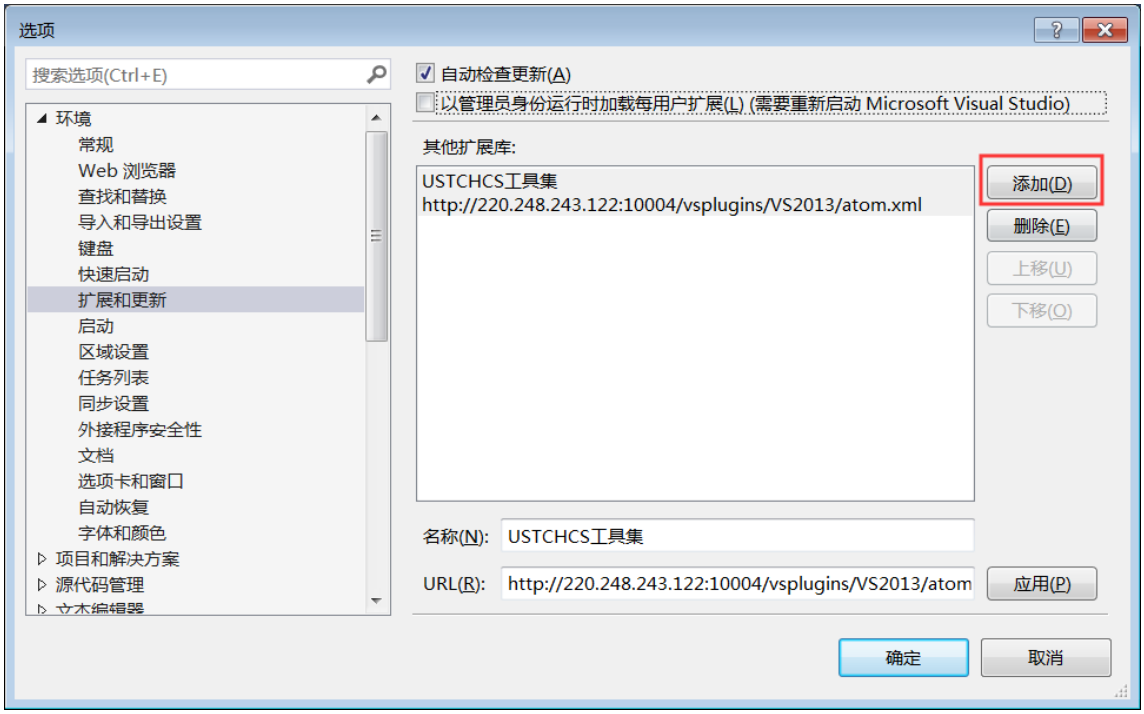

图 2

3）输入服务器地址 http://220.248.243.122:10004/vsplugins/VS2013/atom.xml 和名称 USTCHCS，点击【确定】；

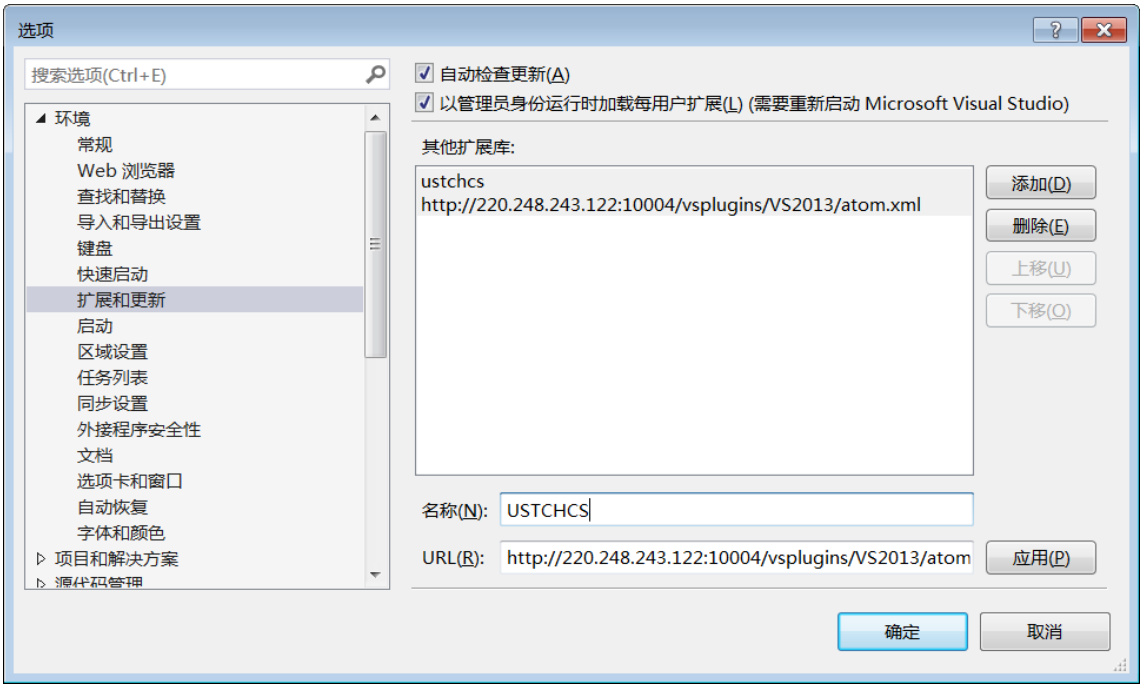

图 3

⚠*注：*

若需安装其他版本，将地址中的 2013 换成 2015 或 2017 即可。

4）选择【工具】菜单中的【扩展和更新】子菜单；

图 4

5）进入菜单后，在左侧选择【联机】中步骤（3）输入的名称项目，右侧出现相应的插件版本，选择【下载】；

图 5

6）下载完成后选择【安装】，完成后重启 VS2013 即完成插件的安装。

2.2 工具运行

打开 VS 并创建或打开 C/C++项目，在菜单栏中可以看到【USTCHCS 分析工具集】；

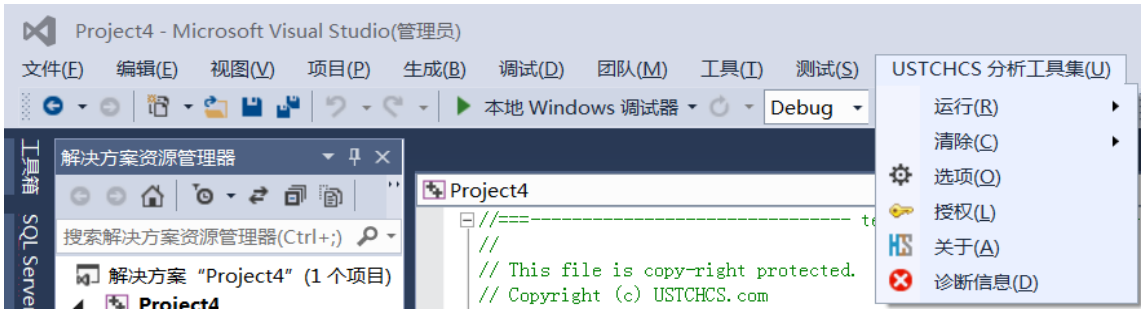

图 6

其中快捷按钮的功能如下：

运行：【运行】菜单后可以选择需要运行的规则集；

清除：【清除】菜单后可以清除规则集运行后的告警标记；

选项：对 USTCHCS 分析工具集做相应的配置；

授权：对 USTCHCS 分析工具集进行授权；

关于：对 USTCHCS 分析工具集的版本、公司、背景等相应的简介；

诊断信息：展示告警信息。

2.1.1 配置规则选项

1）首先打开【USTCHCS 分析工具集】菜单，选择【选项】，弹出选项窗口；

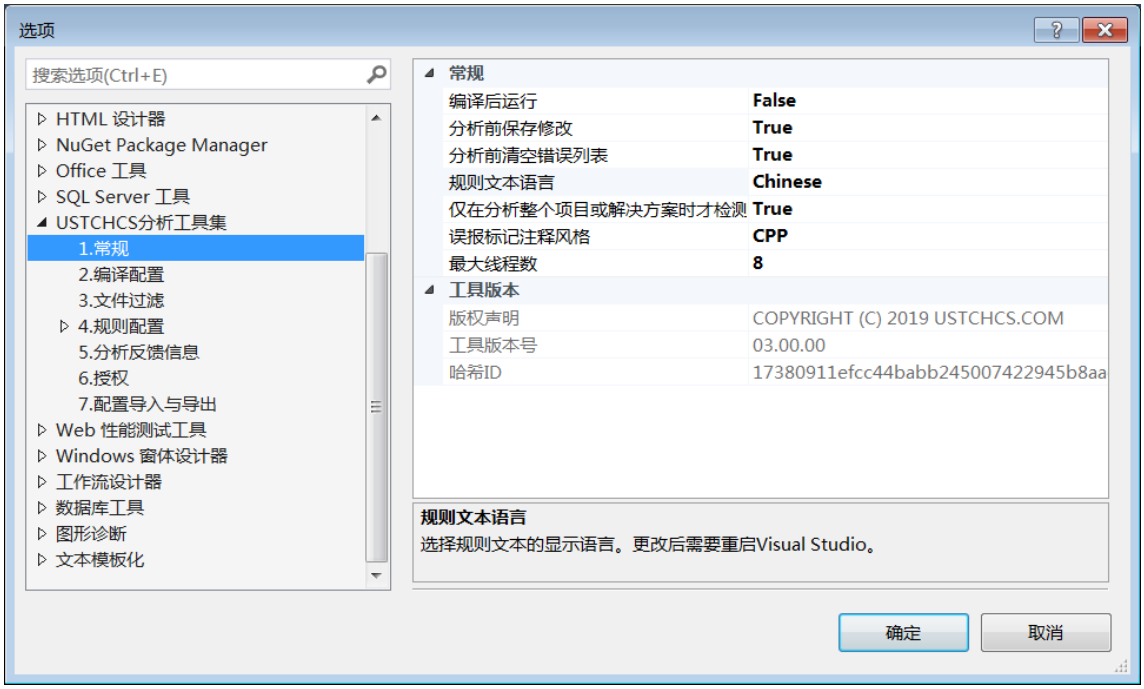

图 7

右侧的选项设置中提供了基础的通用配置，包括：

常规：

- 编译后运行：编译后对实际编译的文件执行测试，检测规则集为工具栏所勾选的规则集；
- 分析前保存修改：在使用工具分析待测文件前保存用户的全部修改；
- 分析前清空错误列表：进行下一次分析时清空上一次运行时生成的错误列表；
- 规则文本语言：设置规则文本语言；
- 仅在分析整个项目或解决方案时才检测系统规则：设置系统规则的检测方式；
- 误报标记注释风格：设置误报标记的注释风格；
- 最大线程数：设置最大的线程数目；

工具版本：显示工具的版本信息。

左侧是对工具集的进一步配置，包括：

- 1.常规
- 2.编译配置
- 3.文件过滤
- 4.规则配置
- 5.分析反馈信息
- 6.授权
- 7.配置导入与导出

后续会进行详细的介绍。

2.1.1 工具授权界面

首先打开【USTCHCS 分析工具集】菜单，选择【选项】，弹出选项窗口，在选项列表的左侧选择【6.授权】；也可以直接在【USTCHCS 分析工具集】中选择【授权】进入授权页面；

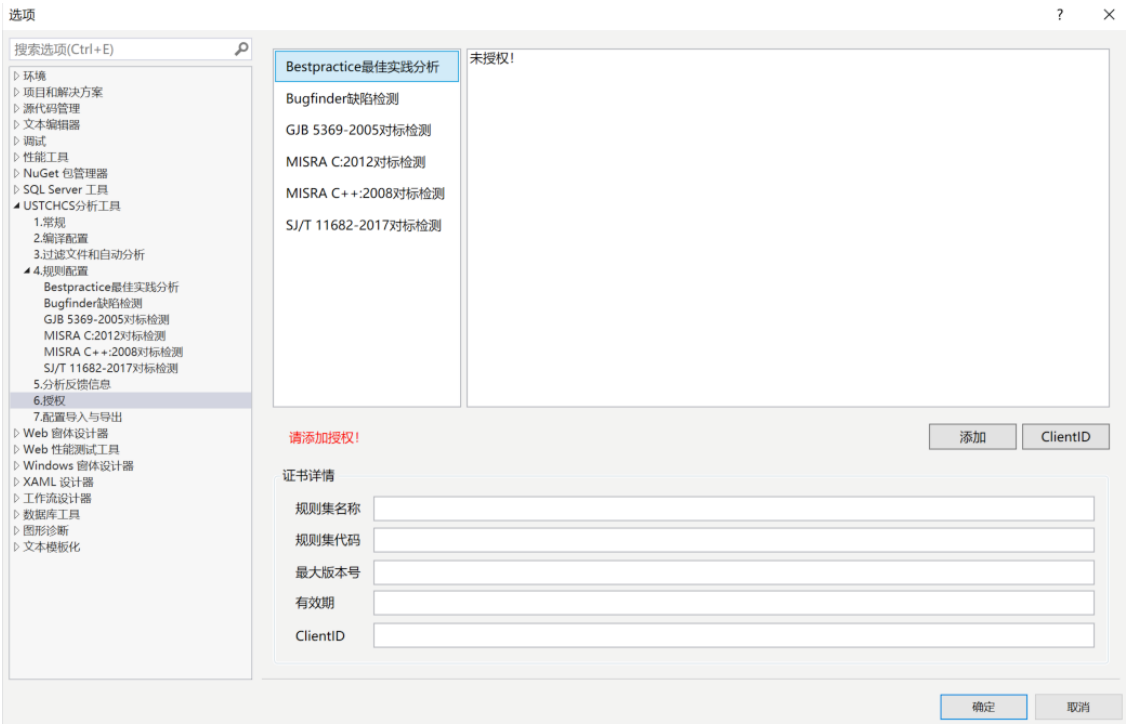

图 8

右侧的选项中提供了规则集的授权配置，包括：

- 列出全部规则集，可对其一一授权；
- ClientID：用于将客户 ID 复制到系统的粘贴板；
- 证书详情：包括规则集名称、规则集代码、最大版本号、有效期和 ClientID 等信息。

工具集的授权步骤如下：

1）将复制的客户标识发送给本公司，并告知需要授权的分析工具集和授权到期时间；

2）选择【添加】后，将接收到的授权码复制到证书输入框中。

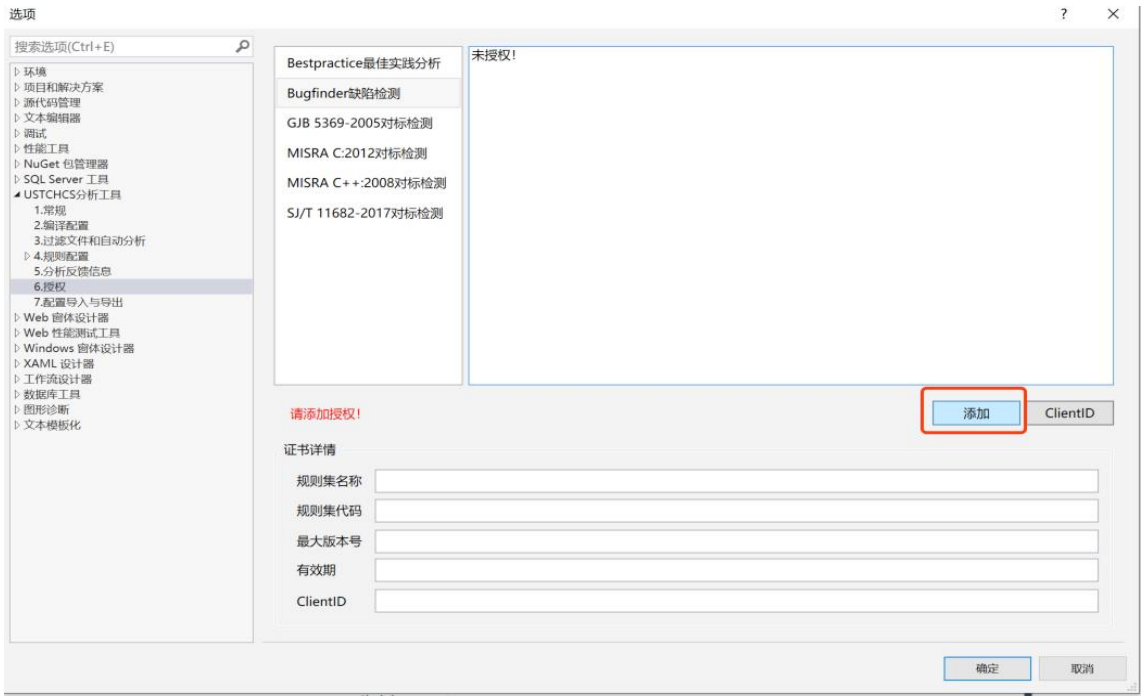

图 9

2）输入完成后自动验证，并更新授权信息。

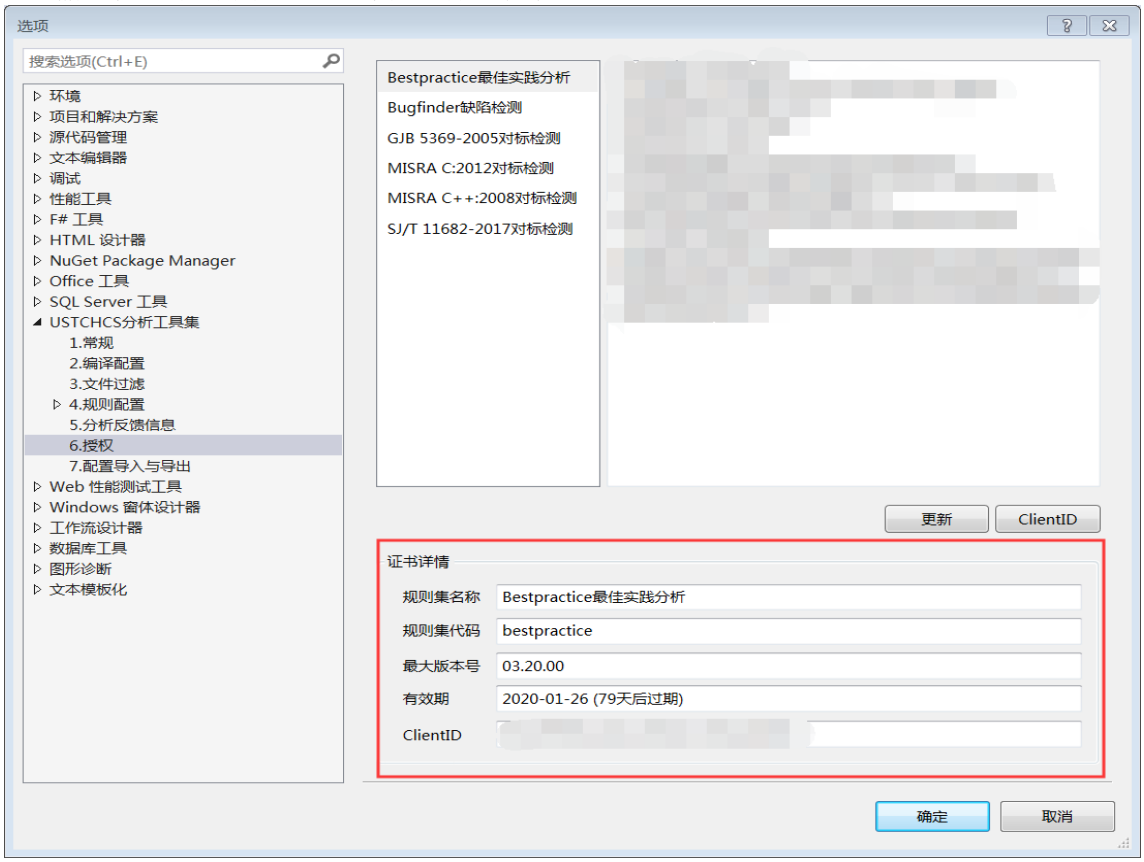

图 10

⚠*注：*

本工具由中科国创高可信软件有限公司进行授权，授权码由中科国创高可信软件有限公司统一进行管理。

2.1.2 编译配置

首先打开【USTCHCS 分析工具集】菜单，选择【选项】，弹出选项窗口，在选项列表的左侧选择【2.编译配置】；

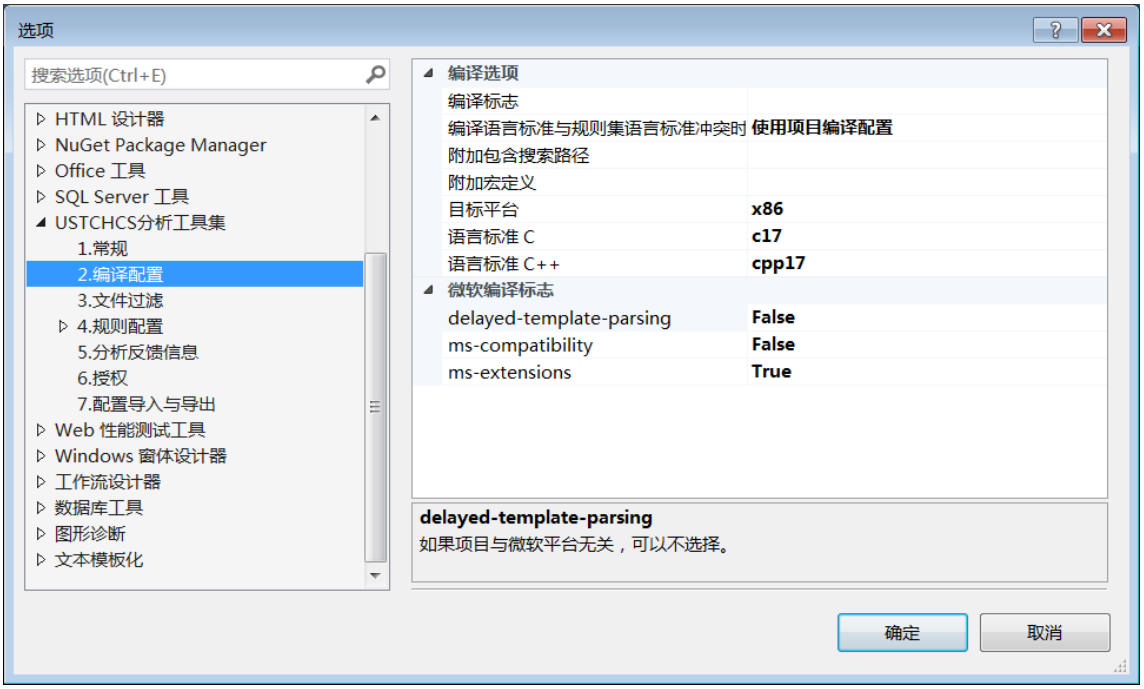

图 11

右侧的选项中提供了规则集的编译配置，包括：

编译选项

编译标志：添加编译标志；

编译语言标准与规则集语言标准冲突时：编译语言标准与规则集语言标准冲突时，设定需要使用的语言标准；

附加包含搜索路径：添加附加的头文件搜索路径（绝对路径）；

附加宏定义：添加额外的宏定义；

目标平台：设置待检测项目使用的平台；

语言标准 C：设置待检测项目使用的 C 语言标准；

语言标准 C++：设置待检测项目使用的 C++语言标准；

微软编译标志

如果项目与微软平台无关，可以不选择；

配置完成后点击【确定】即可。

2.1.3 文件过滤

首先打开【USTCHCS 分析工具集】菜单，选择【选项】，弹出选项窗口，在选项列表的左侧选择【3.文件过滤】；

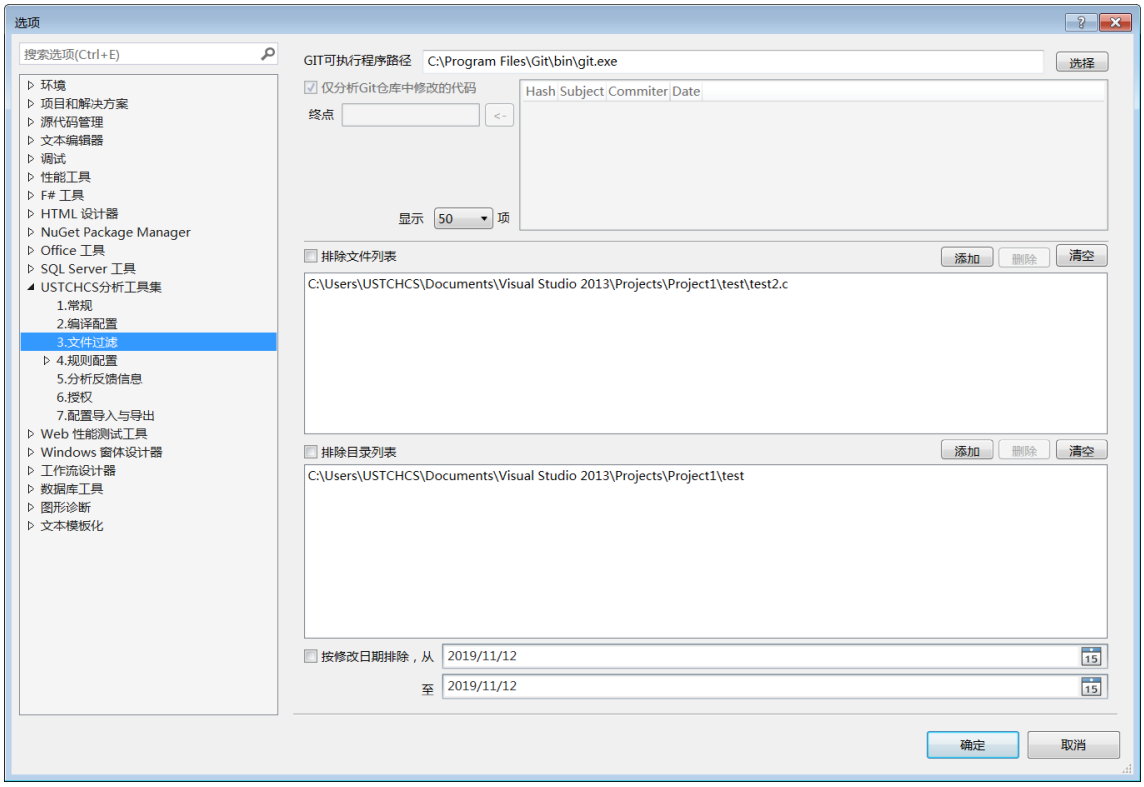

图 12

右侧的选项中提供了规则集的过滤配置，包括：

Git 相关配置

GIT 路径：git.exe 路径，用于调用 git；

仅分析 Git 仓库中修改的代码：勾选后可开启 Git 过滤功能；

终点：根据 Git 仓库的 commit 选定检测终点，起点为当前仓库的状态；

显示项：设置显示的 commit 数；

排除相关配置

排除文件列表：按照文件进行排除；

排除目录列表：按照目录进行排除；

按修改日期排除按照修改日期排除；

配置完成后点击【确定】即可。

2.1.3.1 过滤功能使用

1) 用户首先通过【Git 路径】导入 git 可执行文件路径，然后勾选【仅分析 Git 仓库中修改的代码】选项，即为开启过滤功能；在菜单右侧列表框中选择需要检测的 commit 终点，点击【<-】按钮，即完成 commit 终点的。其中，【显示】选项可以设置显示列表框的 commit 数；

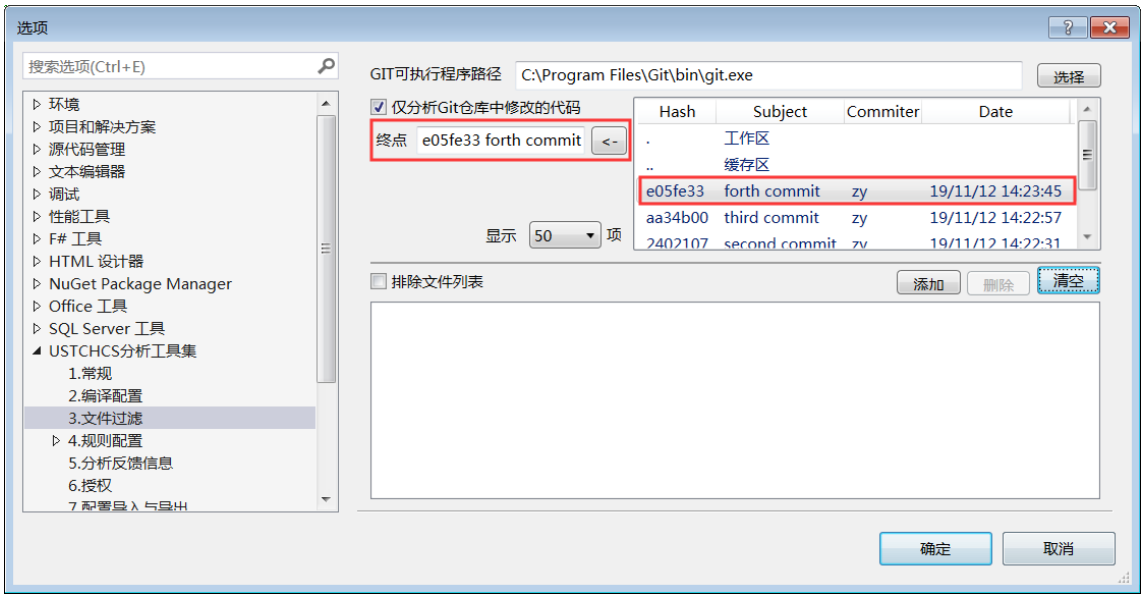

图 13

点击【确定】，即使用本工具将终点与当前仓库的状态做对比，并且仅检测有改动的部分，若无改动，则不进行检测；

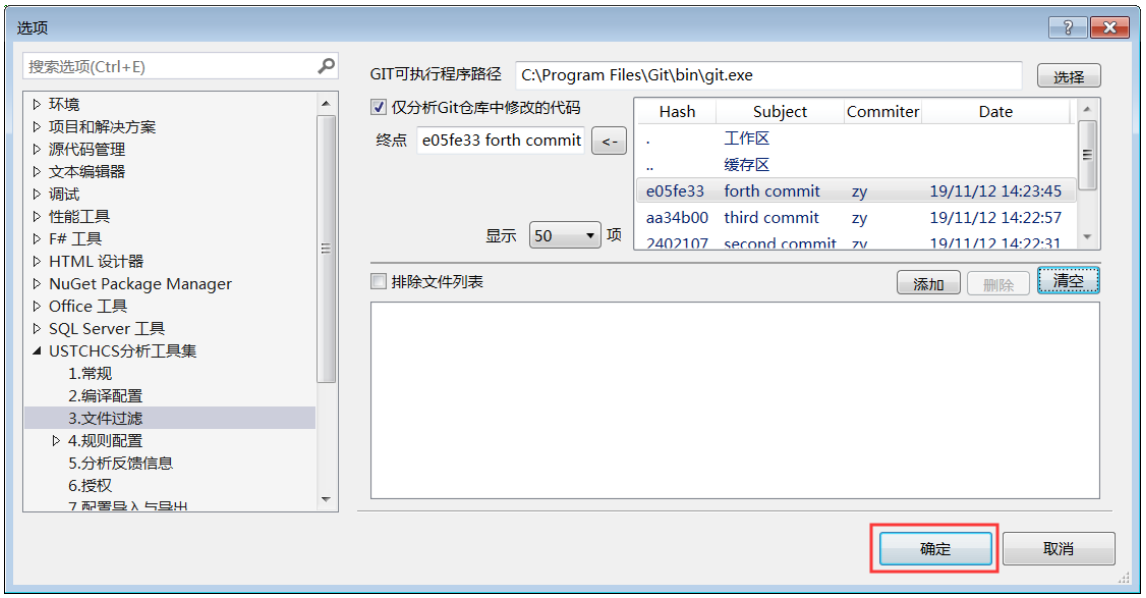

图 14

2) 用户可在勾选【排除文件列表】选项后，点击【添加】按钮，选择需要排除的文件，点击【确定】，即可在使用本工具分析待测文件时过滤掉添加的文件；

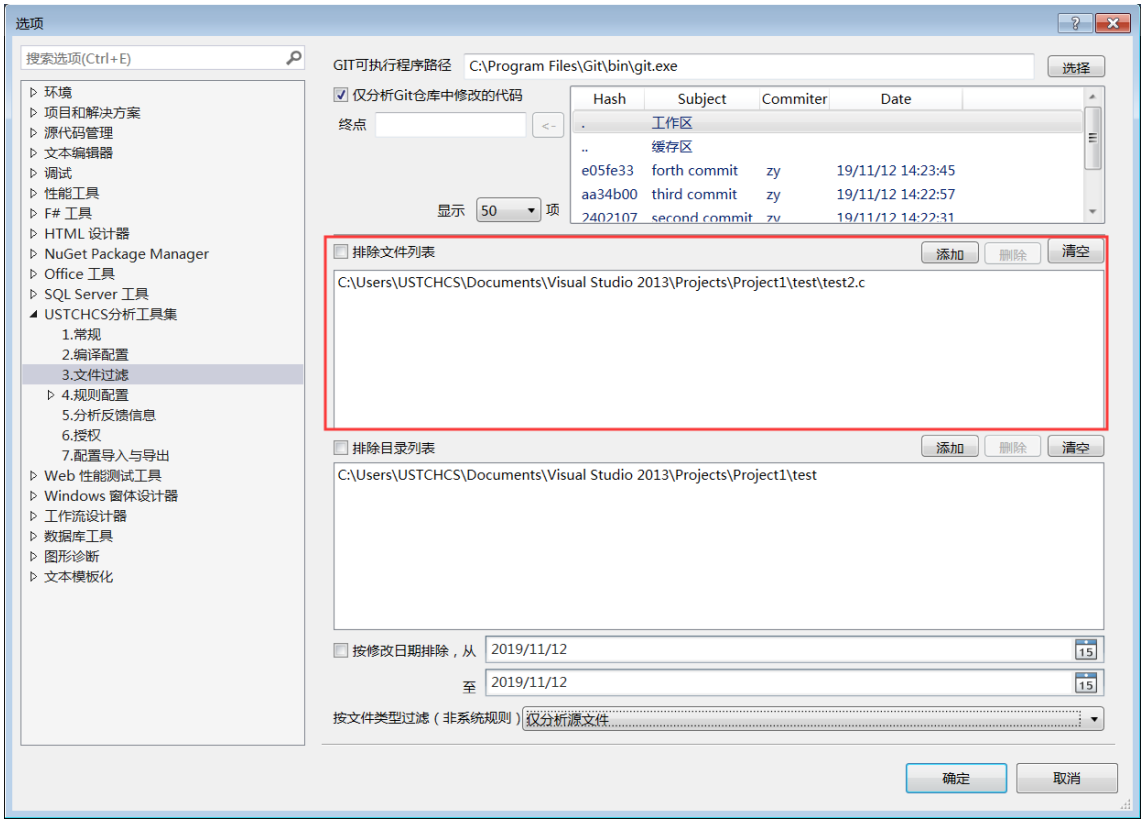

图 15

3) 用户可在勾选【排除目录列表】选项后，选择【添加】按钮，选择需要排除的目录，点击【确定】，即可在使用本工具分析待测文件时过滤掉添加的目录。

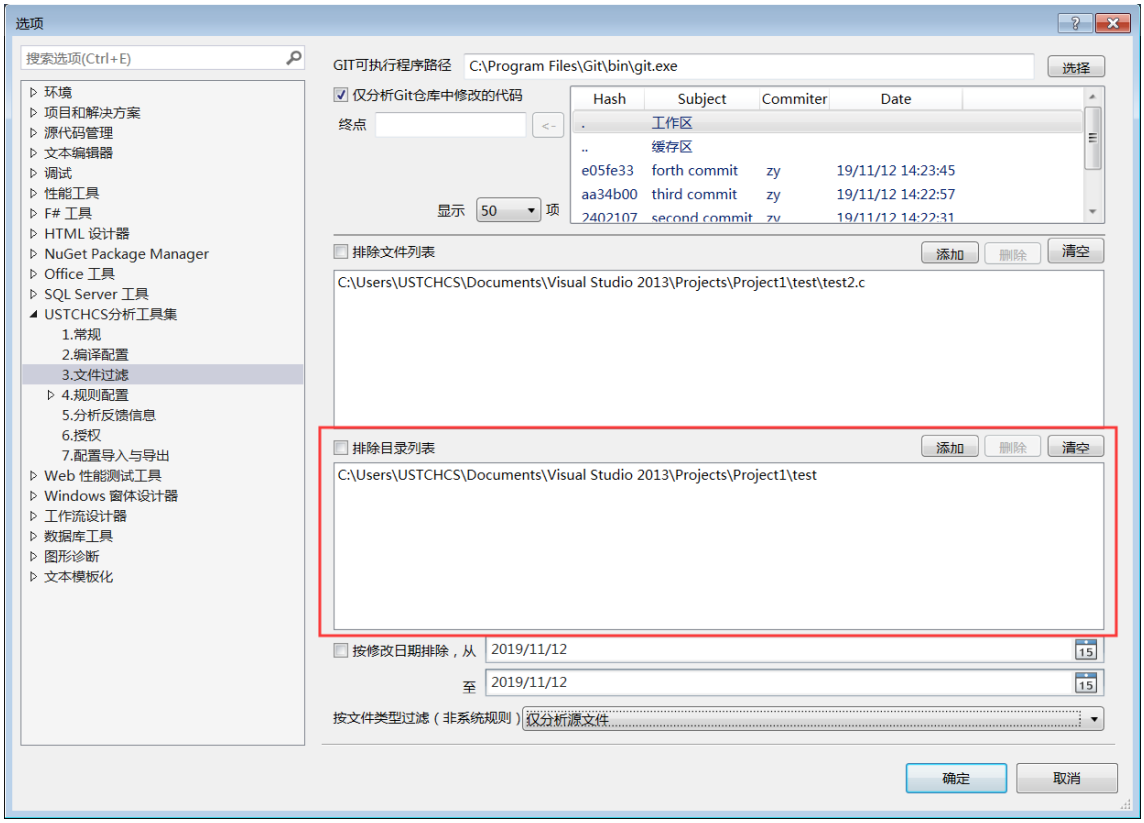

图 16

4) 用户可在勾选【按修改日期排除】选项后，选择需要过滤的时间范围，点击 【确定】，即可在使用本工具分析待测文件时过滤掉该时间段内修改的文件，其中，终止日期不得早于起始日期，否则无法被选中。

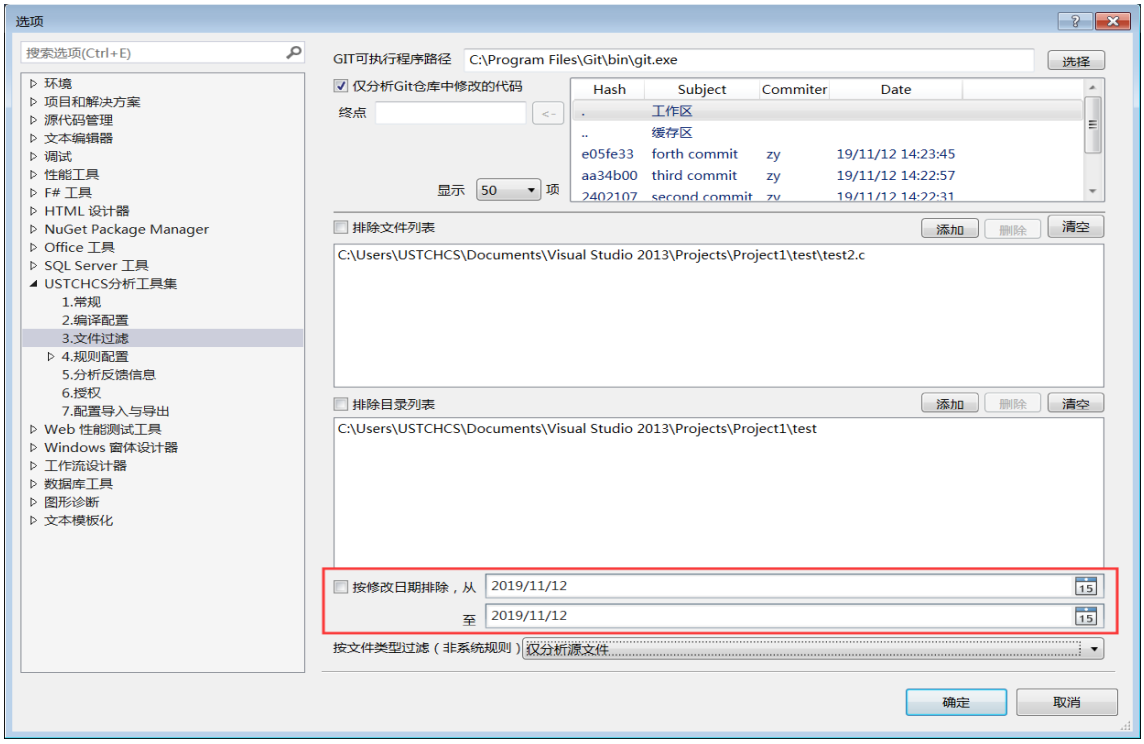

图 17

5) 用户可以在【按文件类型过滤（非系统规则）】中选择需要过滤的文件类型，即可在使用本工具分析待测文件时仅分析用户选择的文件类型。

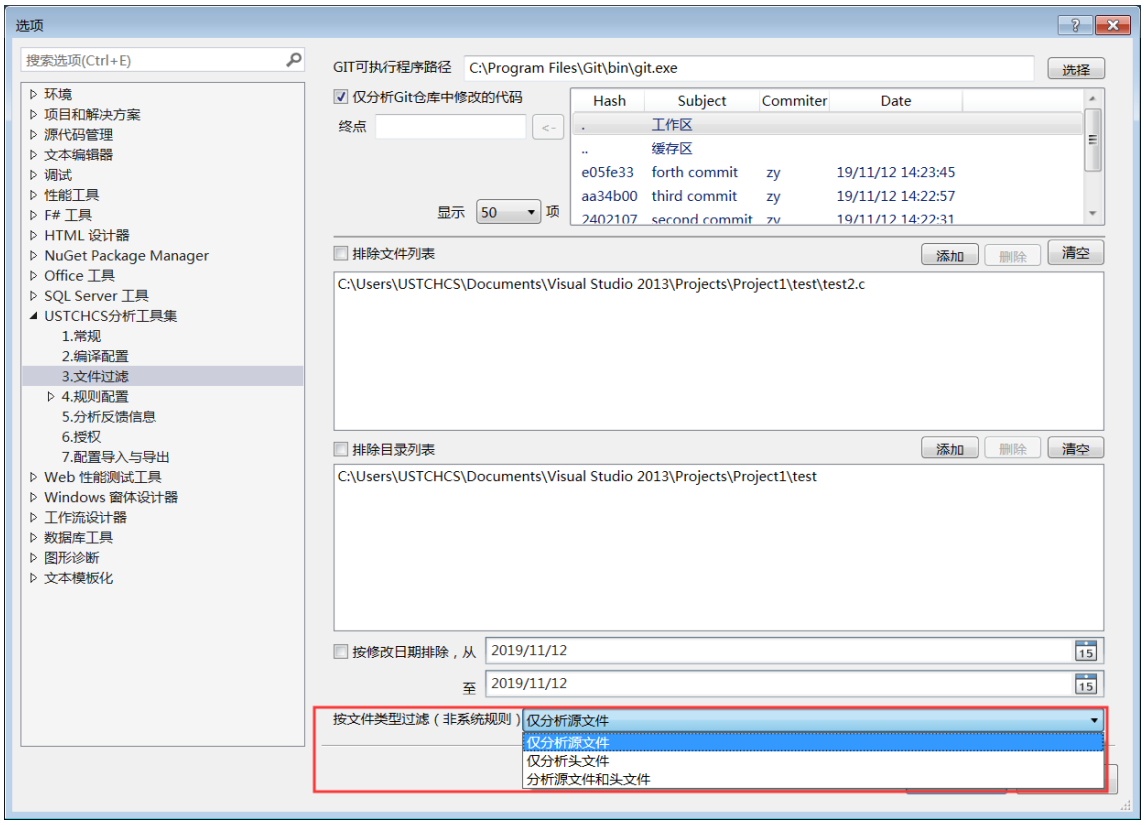

图 18

⚠*注：以上过滤功能可同时使用。*

2.1.4 规则配置（以 Bestpractice 最佳实践分析为例）

首先打开【USTCHCS 分析工具集】菜单，选择【选项】，弹出选项窗口，在选项列表的左侧选择【4.规则集配置】；

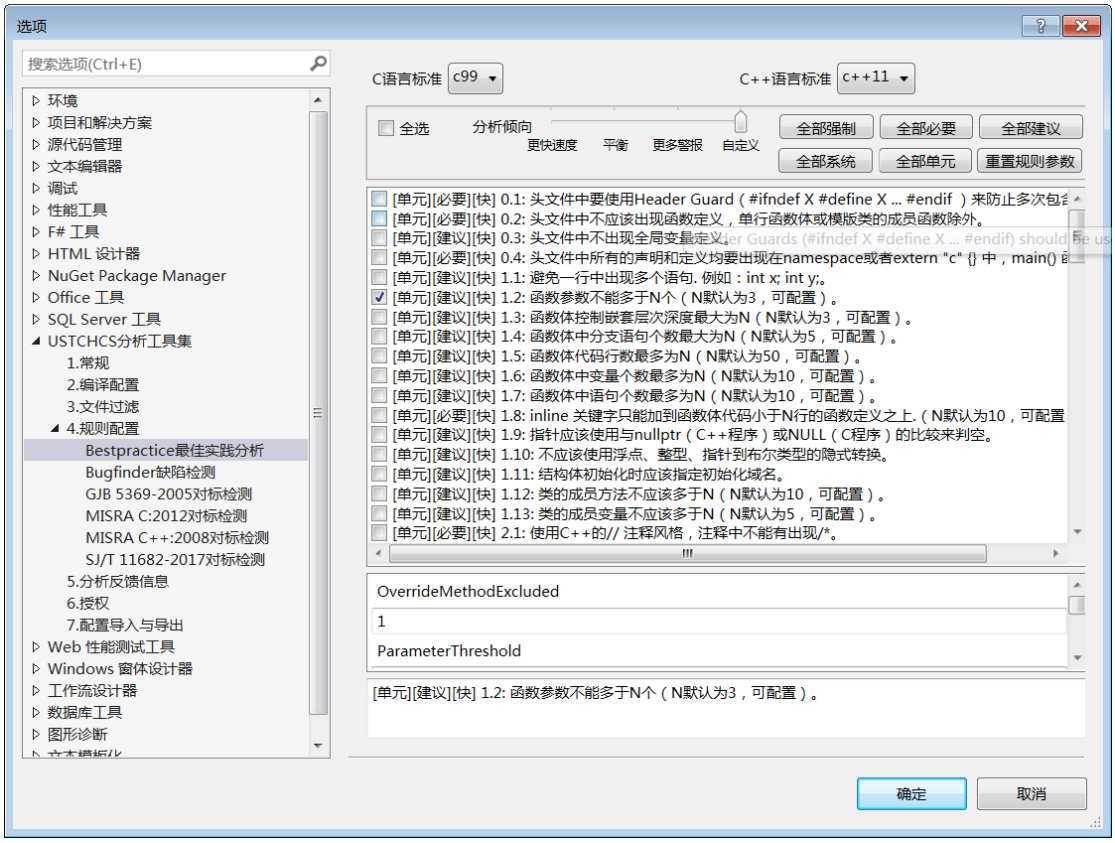

图 19

右侧的选项包括了该规则集中的的相关选择配置以及规则可选项：

规则集配置

- 全选：选择该规则集中的全部规则；
- 分析倾向：通过滑动滑动条筛选规则；
- 全部强制：选择全部强制类规则（若该规则集中没有此类规则则选择规则数为 0，以下选择相同）；
- 全部必要：选择全部必要类规则；
- 全部建议：选择全部建议类规则；
- 全部系统：选择全部系统类规则；
- 全部单元：选择全部单元类规则；
- 重置全部规则参数：部分规则支持额外参数，重置后可将其还原成默认值；

规则可选项

- 选择需要使用的规则（可多选）；

配置完成后点击【确定】即可。

2.1.5 运行检测（以 Bestpractice 最佳实践分析为例）

通过菜单检测：

完成 2.2.5 中的规则集选择后，打开待测 C/C++文件，然后打开【USTCHCS 分析工具集】菜单，选择【运行】，在弹出的右侧窗口处选择要运行的规则集即可；

图 20

通过 ToolBar 检测

1）USTCHCS 工具栏如下：

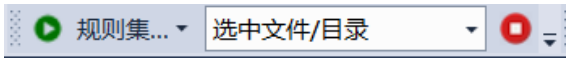

图 21

2）其中，绿色按钮为开始运行，后边的下拉按钮可以选定需要运行的规则集，可多选，未授权的规则不会显示在此界面；

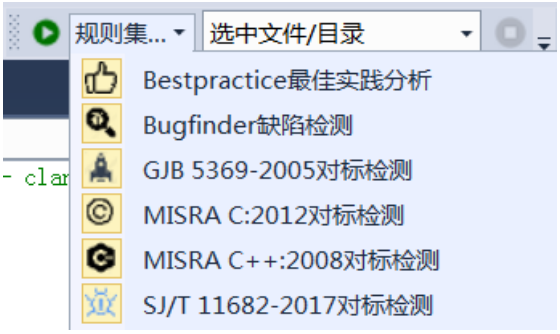

图 22

3）中间的选择列表根据不同的待测代码选择范围提供了如下功能：

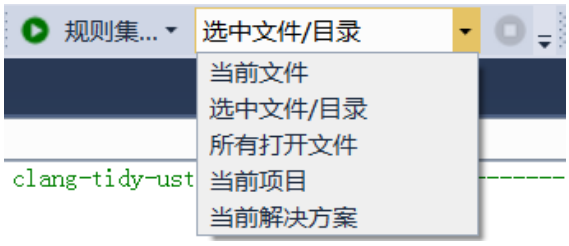

图 23

- 选中文件/目录：对选定的文件/目录进行检测；
- 所有打开文件：仅对所有打开的文件进行检测；
- 当前项目：对当前项目的所有 C/C++文件进行检测；
- 当前解决方案：对当前解决方案的所有 C/C++文件进行检测；

4）红色按钮为强制停止，后边的下拉列表用于添加或移除按钮：

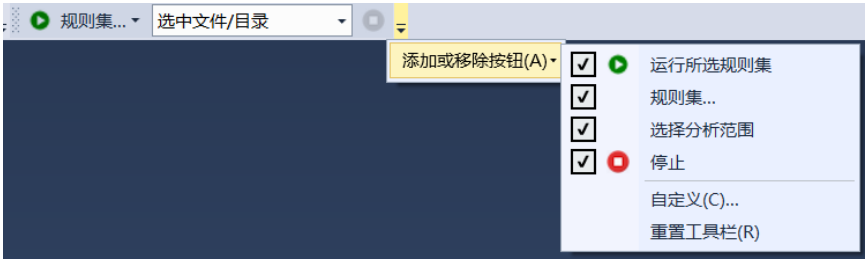

图 24

5）执行完毕后，在诊断信息窗口中查看告警项：

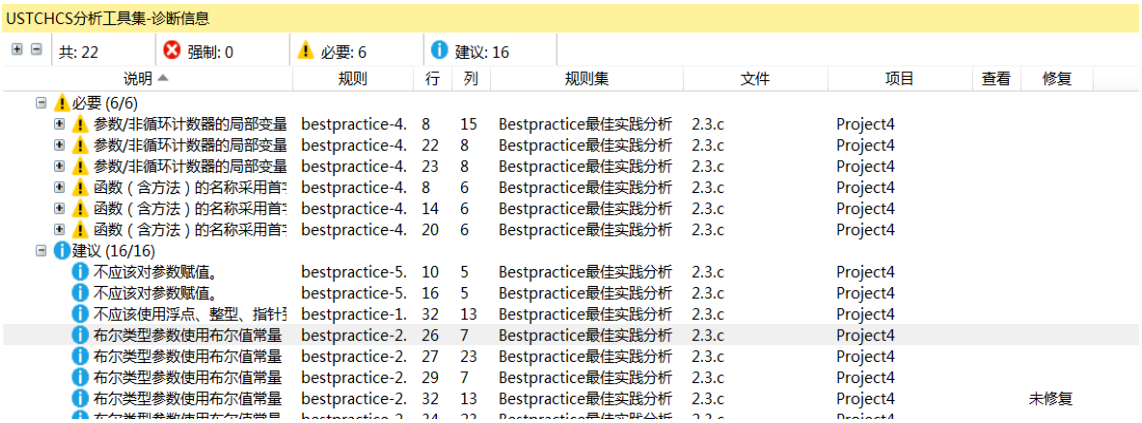

图 25

6）可以通过双击或右键→【跳转到】来查看相应的告警信息，其中，下图中（221/221）指的是显示的告警个数和查到的告警个数，可通过限制配置显示的告警个数。

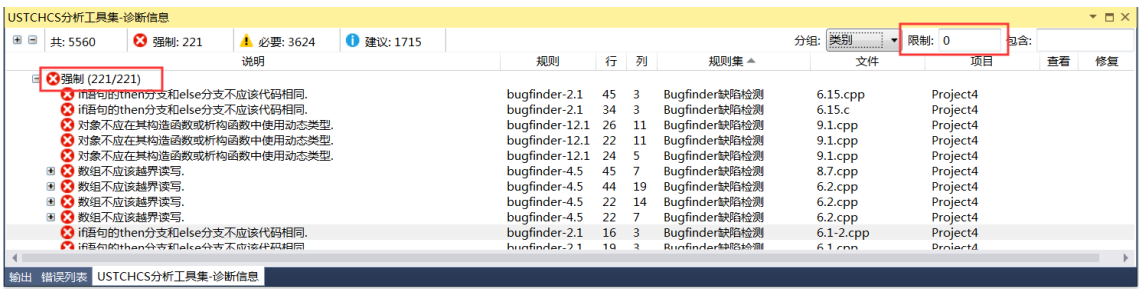

图 26

下图中右上角选项：

- 分组：定义告警显示的分组方式，提供了按照类别、项目、文档等分组方式，未分组则按照默认方式显示；
- 限制：配置显示告警个数的数量，当值为 0 时，可显示全部告警；
- 包含：用于对诊断信息列表中的告警按照关键字进行查找；

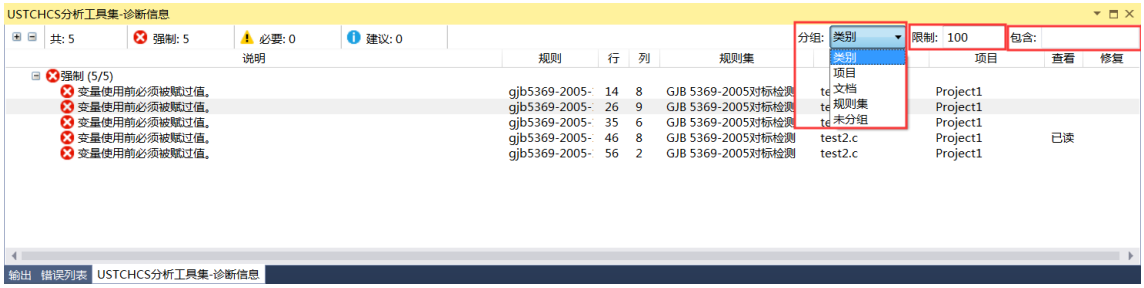

图 27

⚠*注：*

在使用本工具前，可能需要在【USTCHCS 分析工具集】→【选项】→【2. 编译配置】项中根据待测项目使用的语言标准对【编译选项】进行配置，若存在多个语言标准冲突时，需在 2.1.3 小节的【编译语言标准与规则集语言标准冲突时】选项中进行配置。

2.1.6 分析反馈信息

首先打开【USTCHCS 分析工具集】菜单，选择【选项】，弹出选项窗口，在选项列表的左侧选择【5.分析反馈信息】；

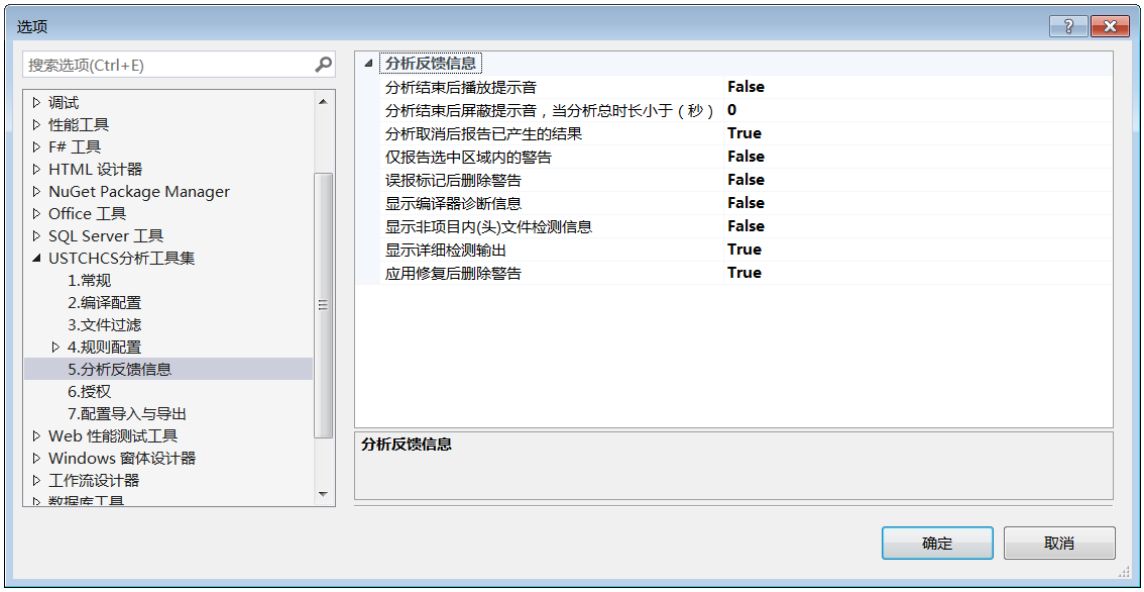

图 28

右侧的选项中提供了规则集的反馈信息配置，包括：

分析结束后播放提示音：分析结束后播放提示音，取消分析后不播放；

分析结束后屏蔽提示音，当分析总时长小于（秒）：若任务时长小于配置时长限制（默认 0 秒），不播放提示音；

分析取消后报告已产生的结果：分析取消后仍会将已产生的结果报告在诊断信息列表中；

仅报告选中区域内的警报：仅报告当前文件选定区域内的警报（在选定区域非空的情况下）；

误报标记后删除警告：将警报标记为误报后在诊断信息窗口删除该警报；

显示编译器诊断信息：显示 clang 编译器诊断信息；

显示非项目内（头）文件检测信息：显示非项目内的头文件检测信息；

显示详细检测输出：显示详细检测信息在输出列表中；

应用修复后删除警告：将已修复的警报在诊断信息窗口中删除；

配置完成后点击【确定】即可；

2.1.7 配置导入与导出

首先打开【USTCHCS 分析工具集】菜单，选择【选项】，弹出选项窗口，在选项列表的左侧选择【7.配置导入与导出】；

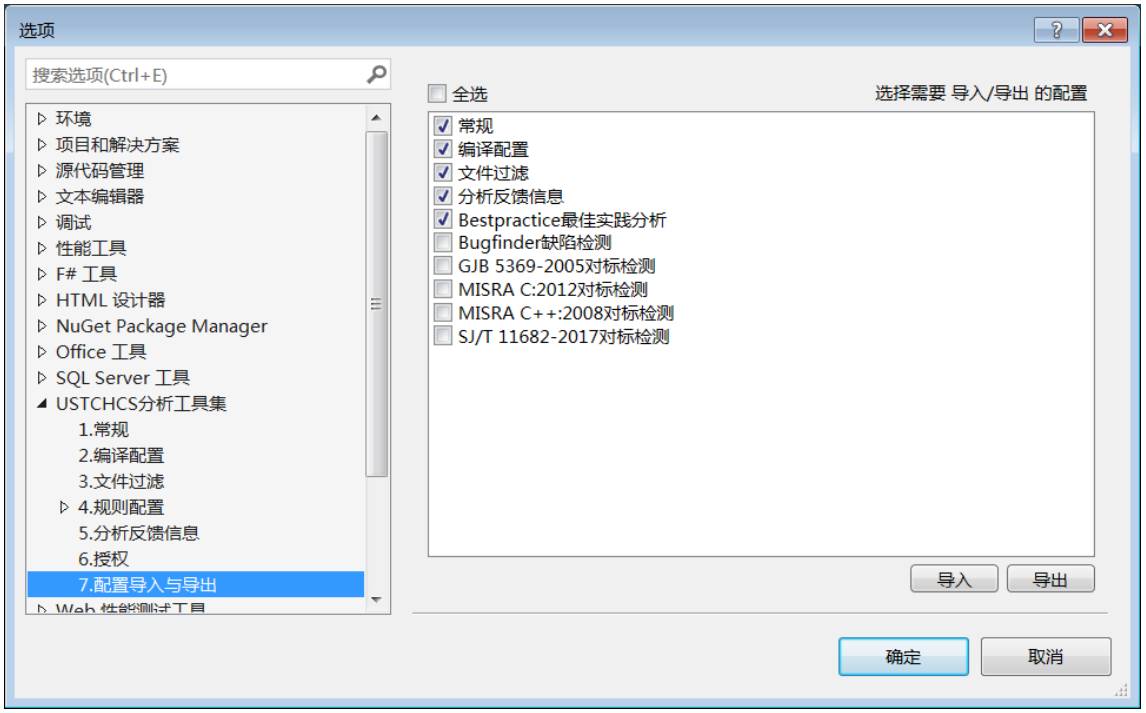

图 29

导出：

1）在右侧的选项中根据用户的需求选择需要导出的项；

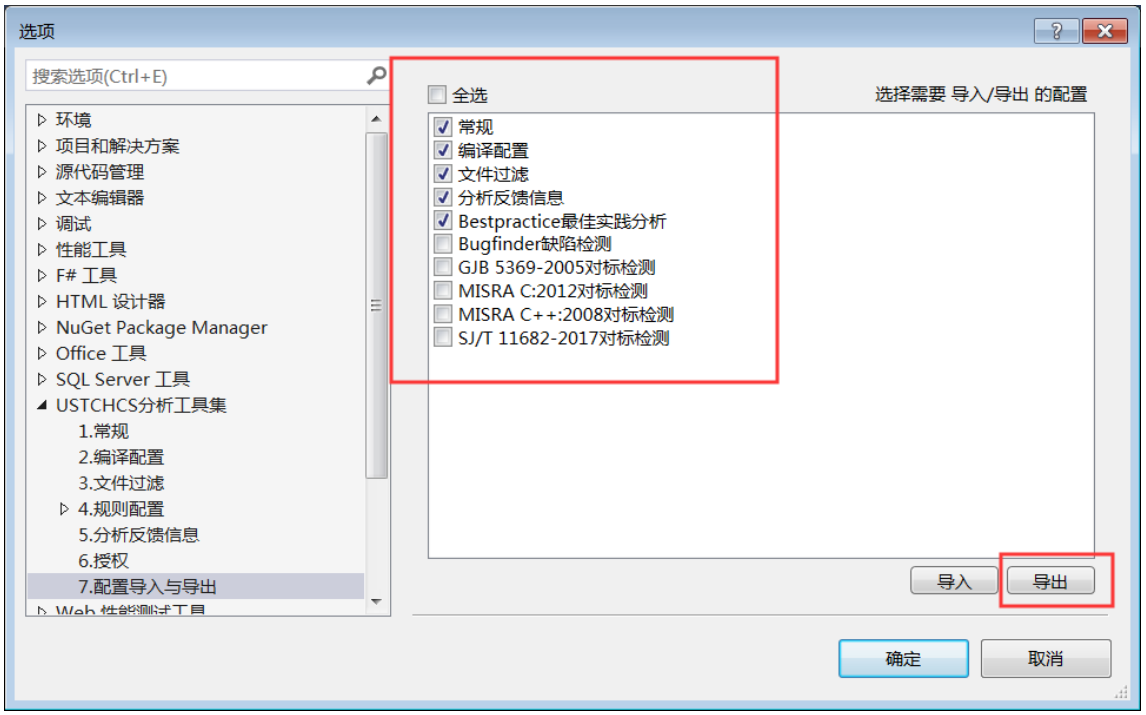

图 30

2）完成后选择右下角的【导出】选项，在弹出的菜单中选择相应的路径，而后【确认】；

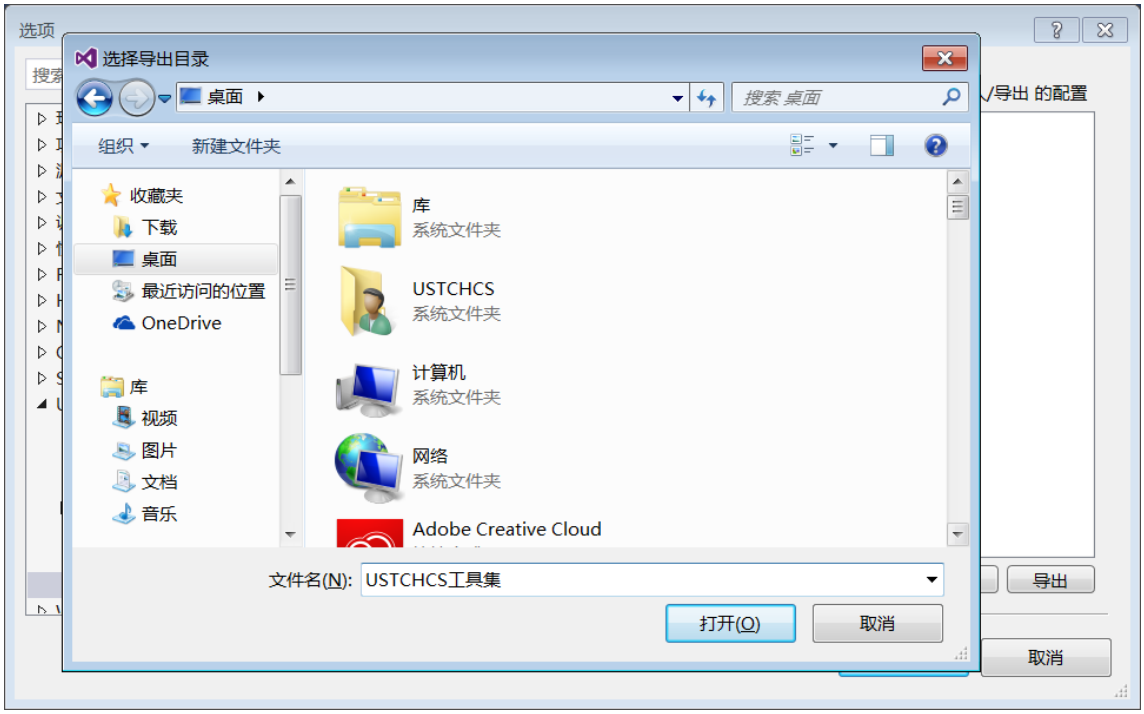

图 31

3）完成后提示信息如下，在相应的路径中查看后缀为.config 的配置文件即可；

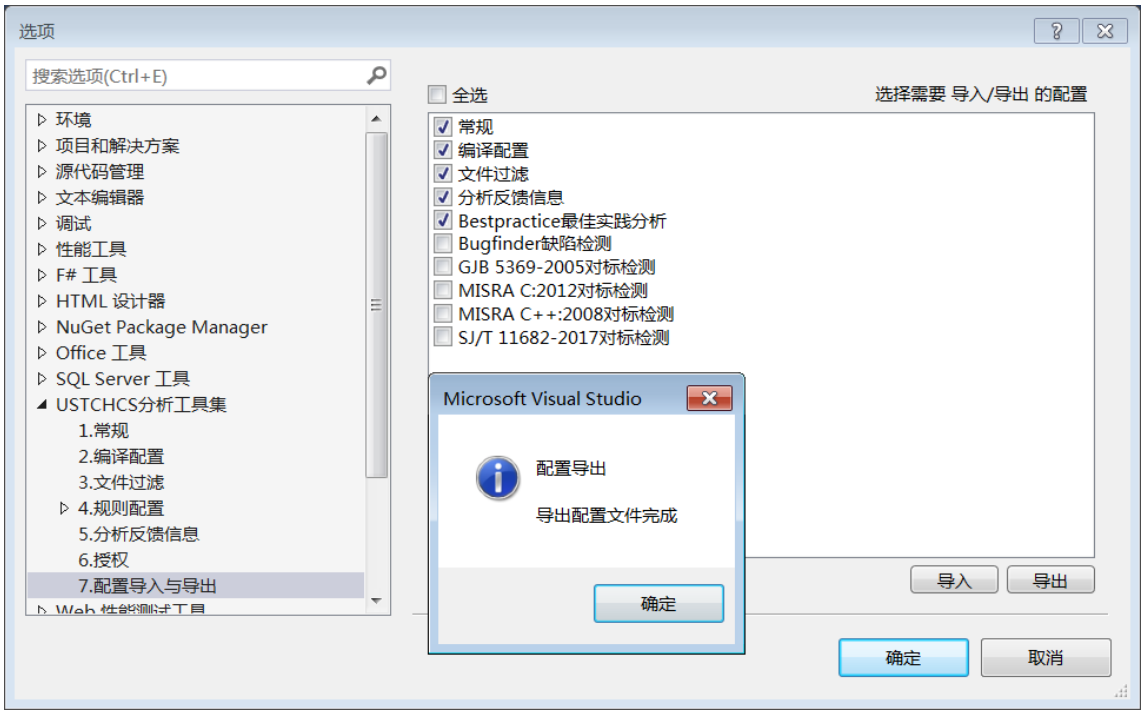

图 32

导入：

1）直接选择【导入】选项；

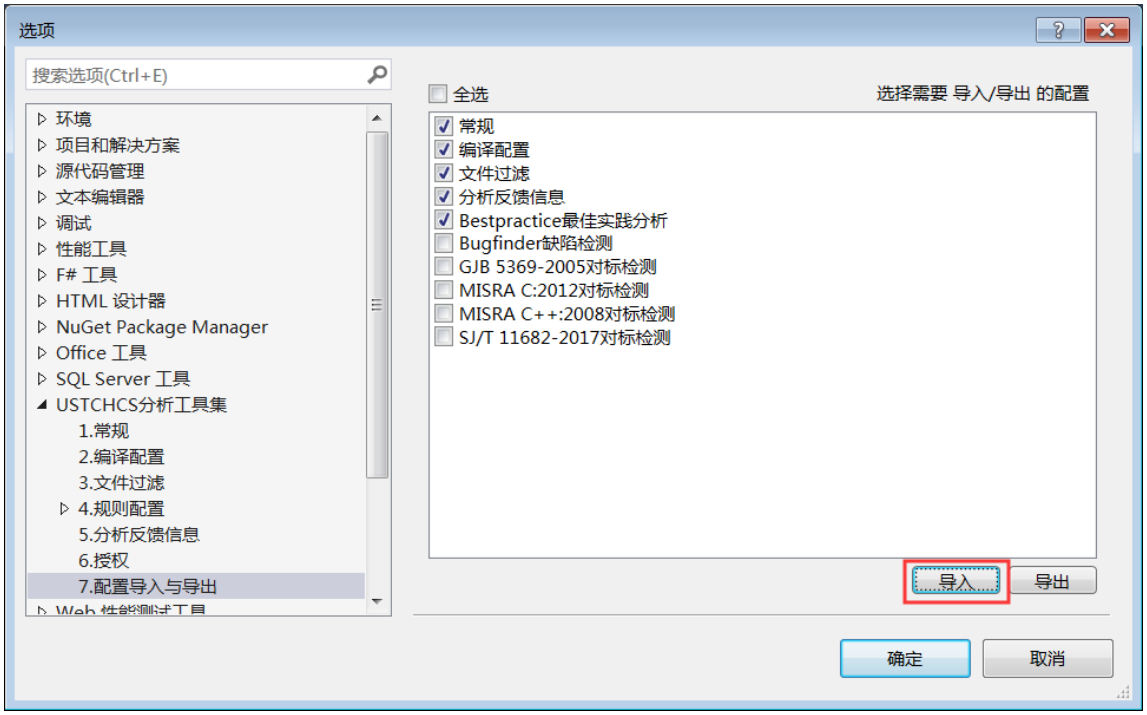

图 33

2）选择配置文件；

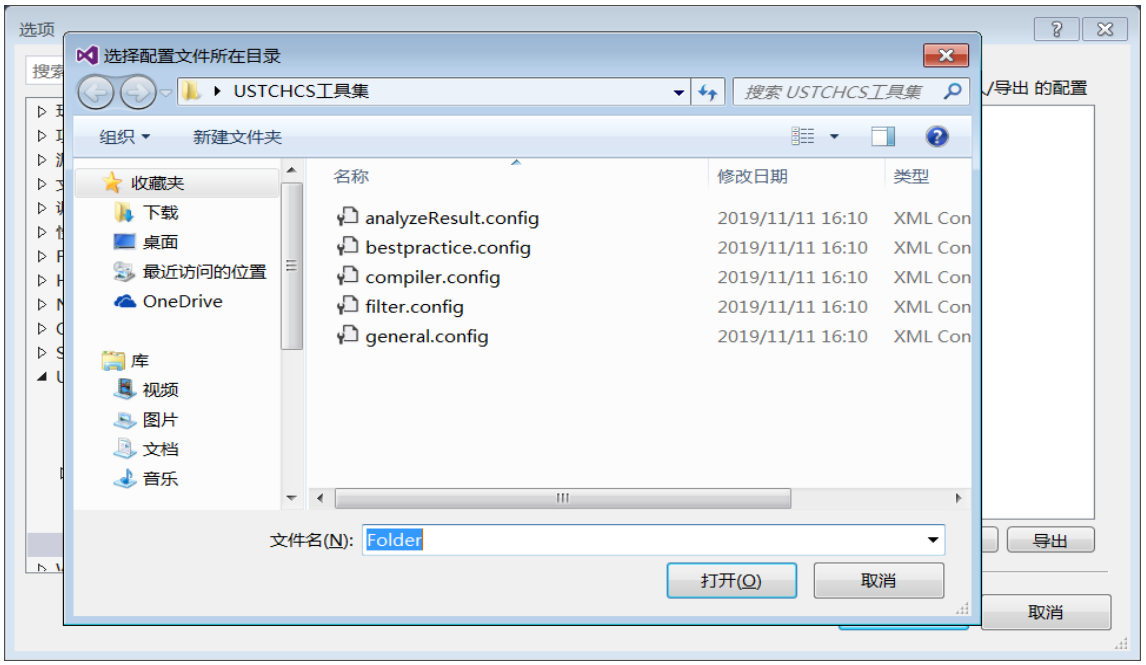

图 34

⚠ *注：*

导入配置文件需勾选对应导入配置的项；

允许在 VS 中同时导入多个项目进行分析。

代码的路径中不要包含中文！

3）完成后提示信息如下，重启 VS 后即可。

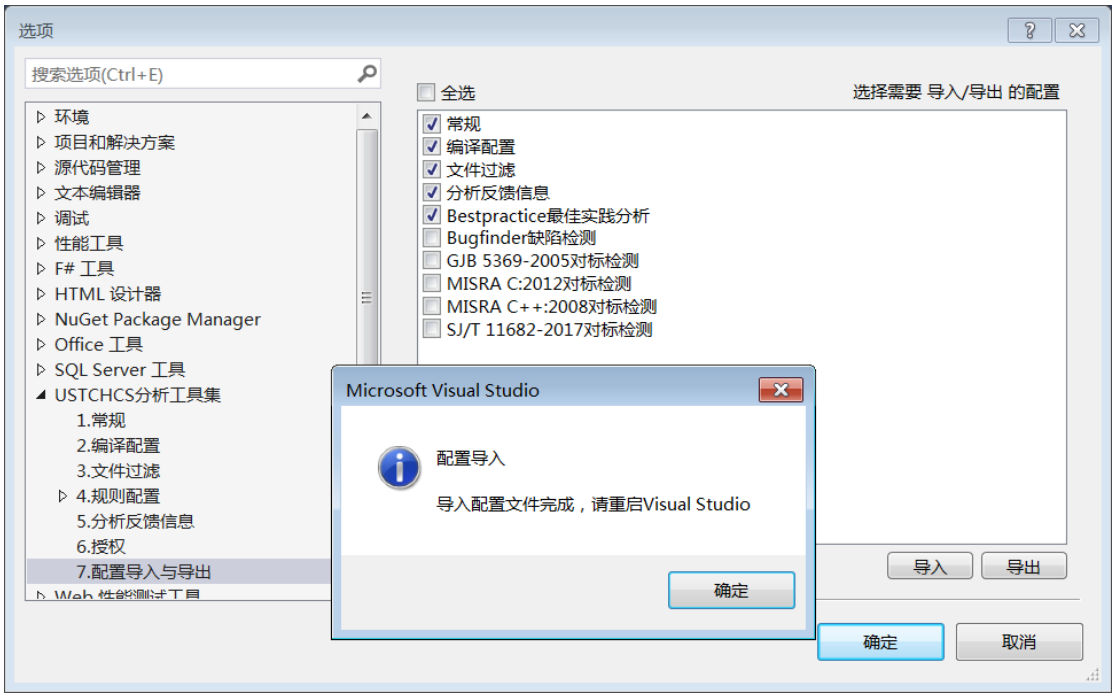

图 35

2.1.8 误报标记功能

针对每个警告，都提供了误报标记功能，用户可以通过在诊断信息列表中选中想要标记为误报的项，右键→【标记误报】，将该警报标记为误报。

图 36

若在 2.2.7 中将【误报标记后删除警告】置为 true，则标记为误报的警报将不再显示在诊断信息列表中；反之，告警仍在诊断信息列表中，但源代码中被标记为误报的行会被打上一个 tag，在执行下一次的分析时，不会再有相应的告警信息。

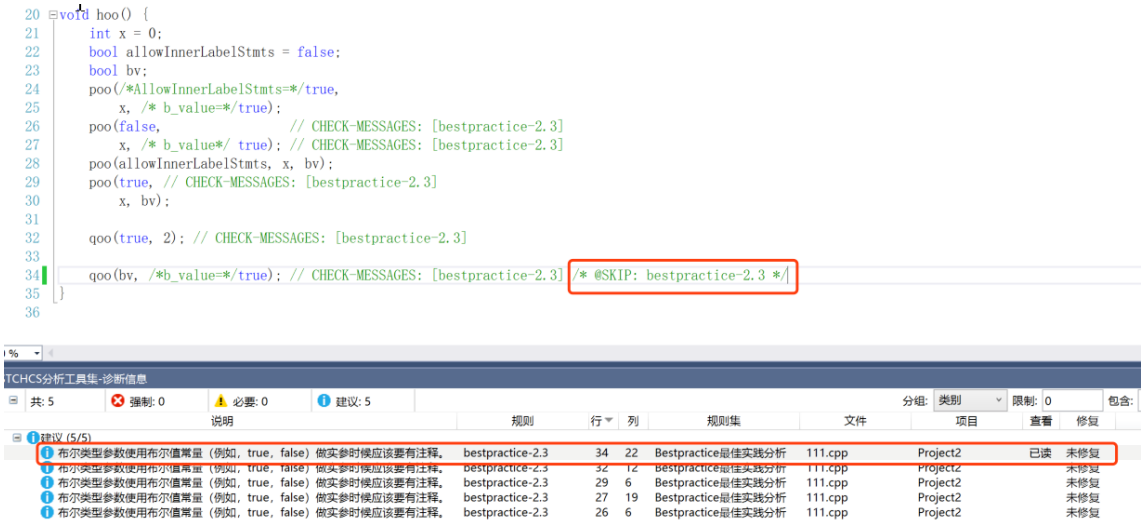

图 37

2.1.9 自动修复功能

针对部分规则，分析工具提供了自动修复功能，以 Bestpractice2.3 为例;

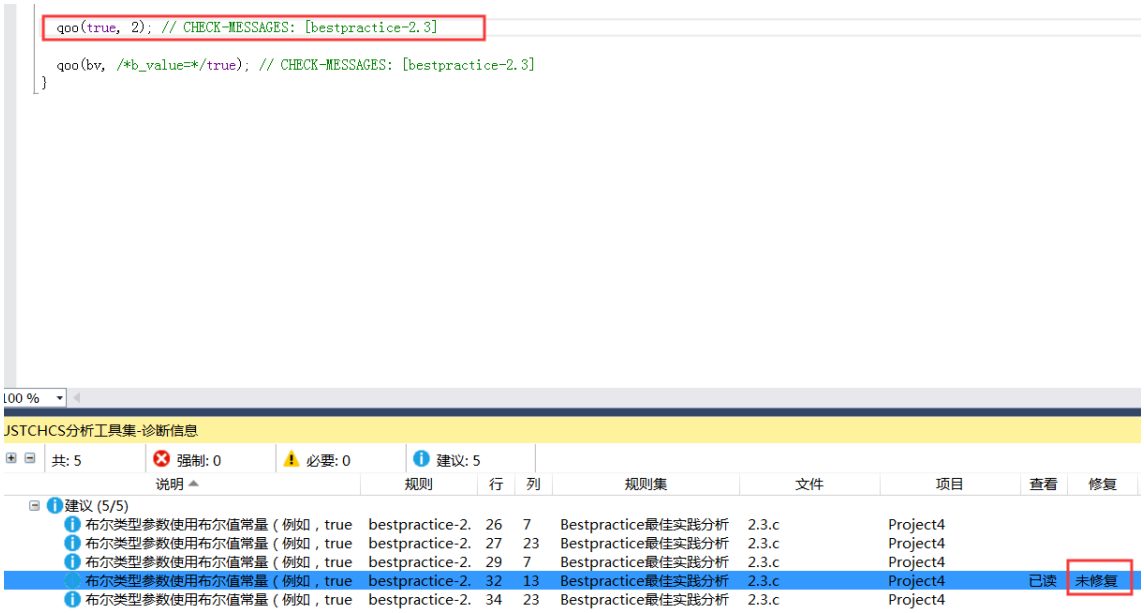

图 38

用户可以通过右键诊断信息窗口中的告警信息，弹出操作菜单，选择【应用修复】来应用自动修复功能。

图 39

用户可以在 USTCHCS 分析工具集【选项】中选择【分析反馈信息】中的【应用修复删除警告】选择是否在修复完成后删除告警。选择【True】修复后，在诊断信息列表中的【修复】状态栏会由未修复变为已修复状态，如下：

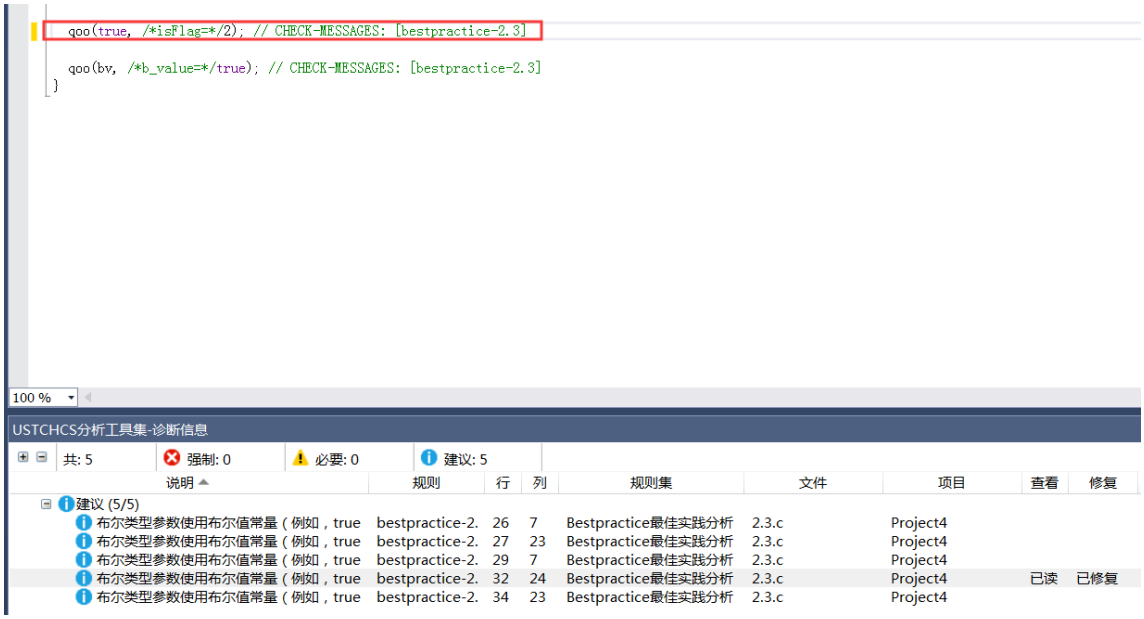

图 40

选择【False】修复后，在诊断信息列表中该警告会自动删除，如下：

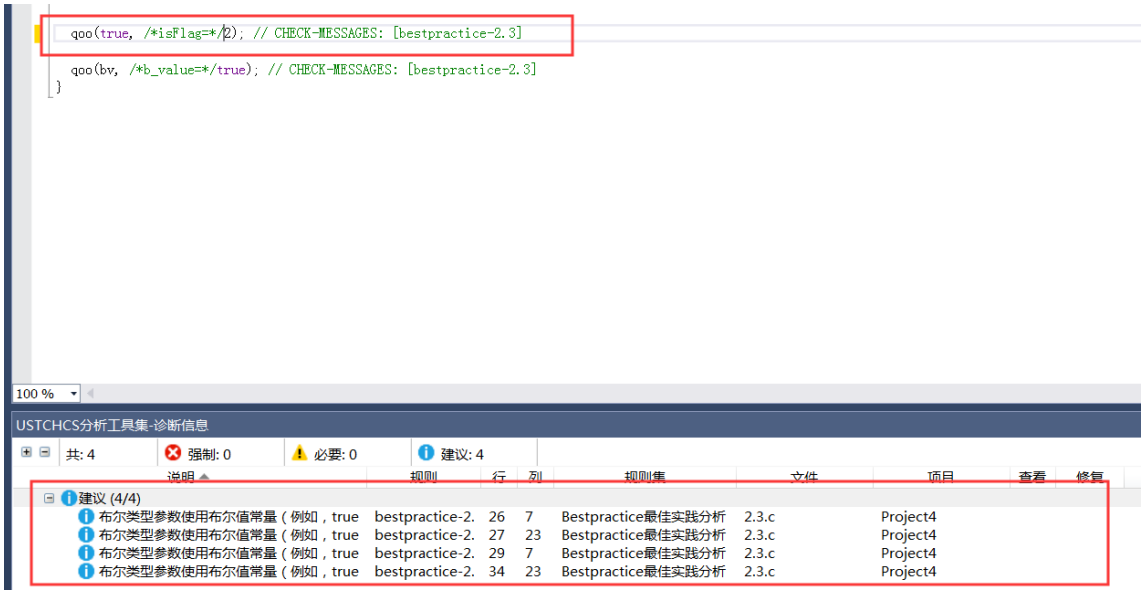

图 41

⚠ *注：*

若代码文件中存在“自动修复”标记，当用户修改了该代码文件，应保存修改后再应用“自动修复”功能，以确保“自动修复”功能的正常应用。

2.1.10 清除功能

本工具还提供将诊断信息窗口中的标记（警报）清除的功能，用户可以直接在诊断信息窗口中选中警报（支持多选）,然后在右键菜单中选择【删除】菜单进行删除；

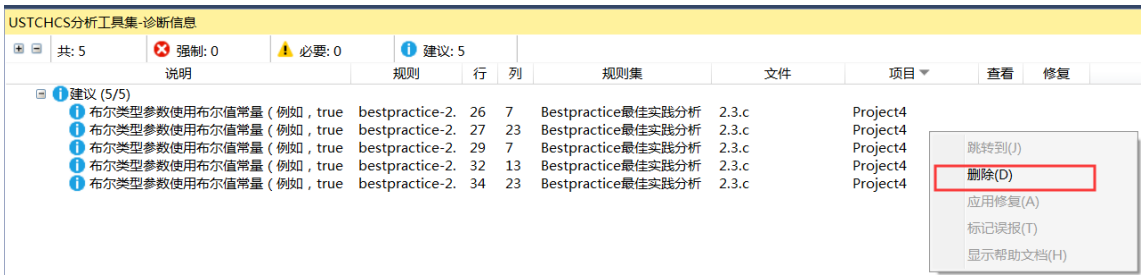

图 42

也可以在【USTCHCS 分析工具集】→【清除】中，选择要清除的标记类型。

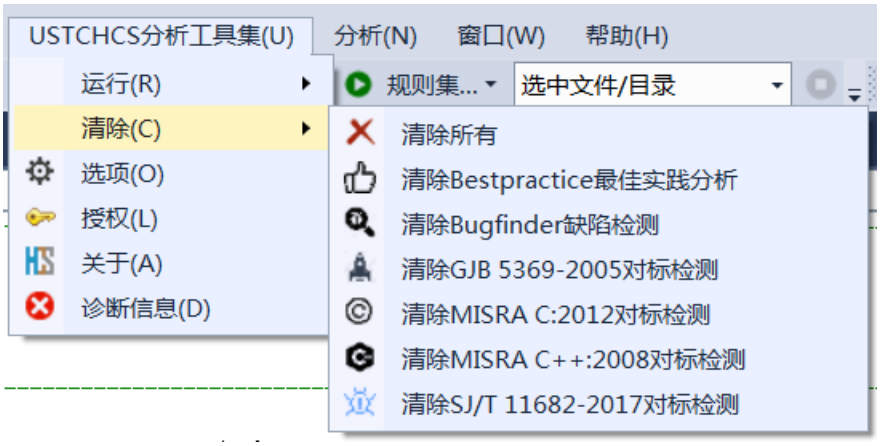

图 43

另外用户还可以在【选项】窗口的左侧点击【1.常规】，然后双击【分析前清空错误列表】选项，选择是否在分析前清空错误列表。

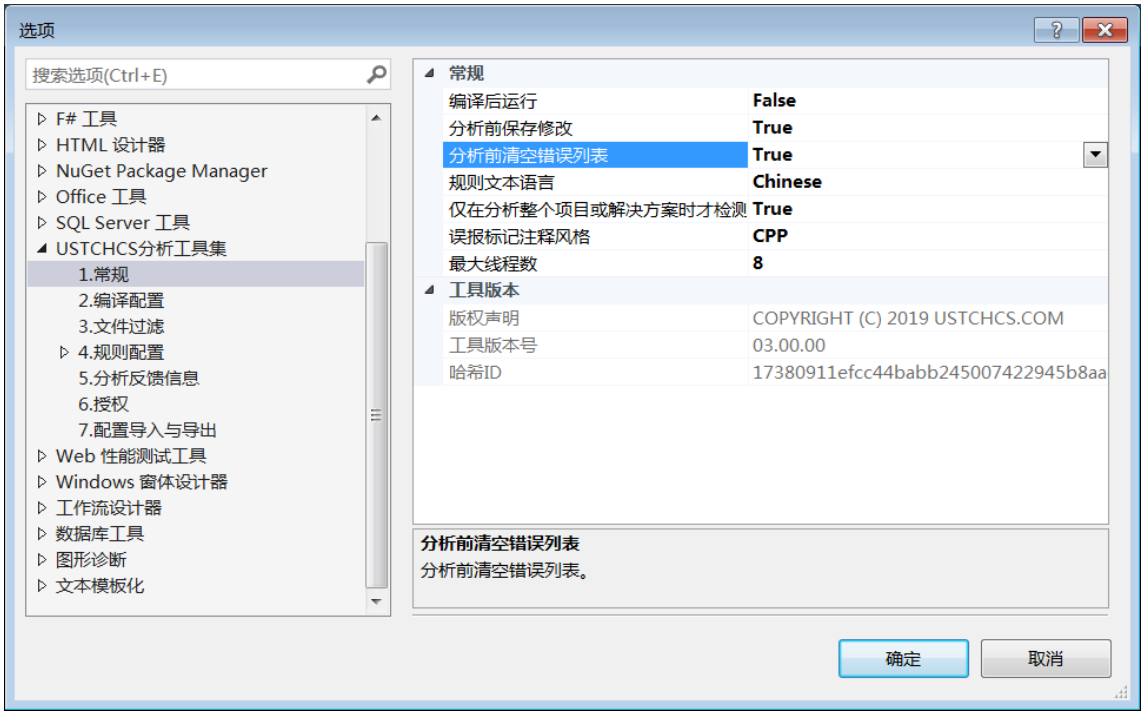

图 44

2.1.11 帮助功能

USTCHCS 分析工具集菜单中提供了帮助菜单，可以快捷的调出系统帮助，方便用户查阅；

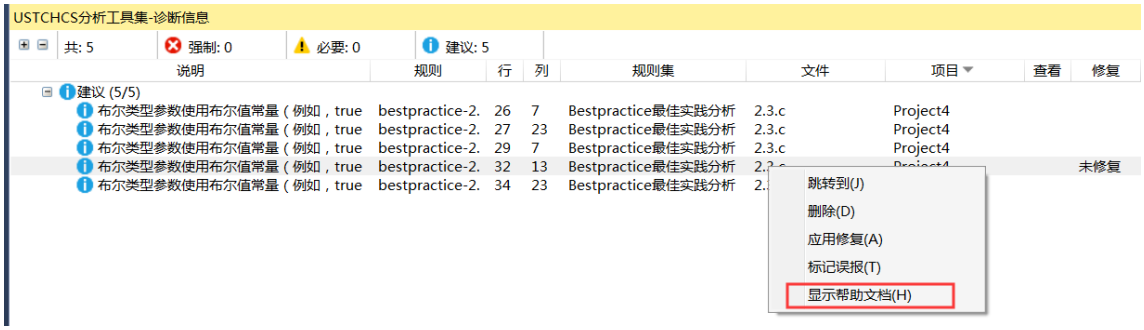

图 45

下图为 bestpractice-2.3 规则的帮助文档。

图 46

2.1.12 分析结果导出功能

本工具提供分析结果导出功能，用户可以在直接在诊断信息窗口中右键，选择【导出】选项，即可导出全部分析结果；

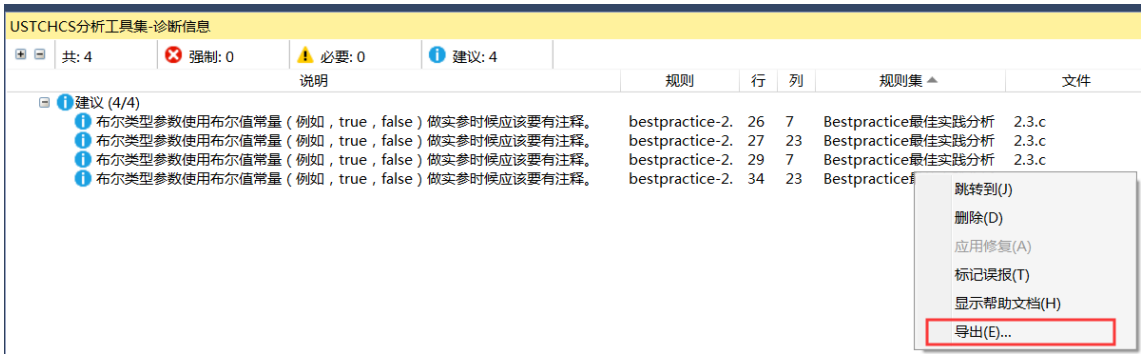

图 47

分析结果为 csv 格式。

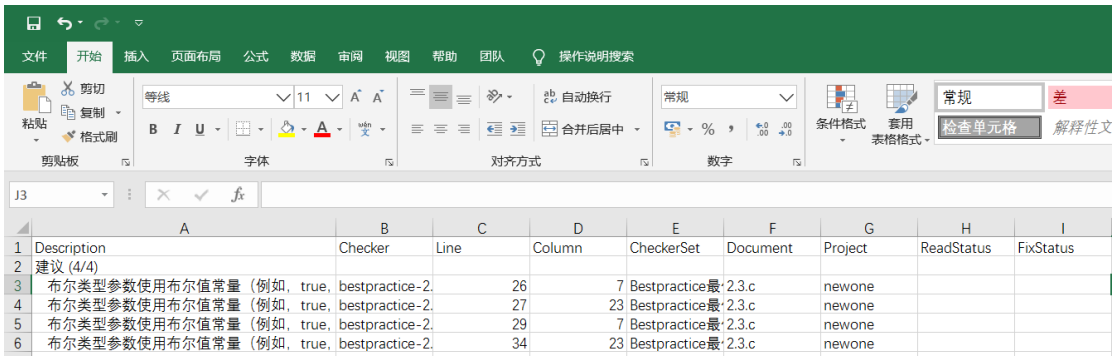

图 48

3. 注意事项

3.1 规则集对于语言的限制

表 10：规则集对于语言的限制

规则集名称	支持的语言标准
Bestpractice 最佳实践分析	C 标准：C90，C99（默认），C11 C++标准：C++03，C++11（默认），C++14，C++17
Bugfinder 缺陷分析	C 标准：C90，C99（默认），C11 C++标准：C++03，C++11（默认），C++14，C++17
代码度量检测	C 标准：C90，C99（默认），C11 C++标准：C++03，C++11（默认），C++14，C++17
GJB 5369-2005 对标检测	C 标准：C90，C99（默认） C++标准：不支持
MISRA C:2012 对标检测	C 标准：C90，C99（默认） C++标准：不支持
MISRA C++:2008 对标检测	C 标准：不支持 C++标准：C++03（默认），C++11，C++14，C++17
SJ/T 11682-2017 对标检测	C 标准：C90，C99，C11（默认） C++标准：C++03，C++11（默认），C++14，C++17

3.2 规则集禁用规则说明

MISRA C:2012 中未实现的规则及部分实现规则：

表 11：MISRA C:2012 中未实现的规则及部分实现规则

规则集	规则编号	禁用原因
Bestpractice 最佳实践分析	5.1	误报较多，待完善
	6.9	误报较多，待完善
	10.1	未实现
	10.3,10.4	误报较多，待完善
Bugfinder 缺陷分析	1.2	未实现
	5.16	误报较多，待完善
	10.5	未实现
	12.5	未实现
GJB 5369-2005 对标检测	1.1.1,1.1.2	误报较多，待完善
	2.1.1	未实现
	2.1.8	未实现
	4.1.1	未实现
	6.1.14	未实现
	6.1.18	未实现
	10.2.1	未实现
MISRA C:2012 对标检测	1.1~1.3	未实现
	2.2	未实现
	2.3~2.5	误报较多，待完善
	5.1,5.7~5.9	误报较多，待完善
	8.3,8.5~8.7	误报较多，待完善
	8.13	未实现
	17.2	误报较多，待完善
	18.1~18.3,18.6	未实现
	22.1~22.6	未实现
MISRA C++:2008 对标检测	0-1-6，0-1-9	未实现
	0-2-1	未实现
	0-3-1,0-3-2	未实现
	0-4-1,0-4-3	未实现
	1-0-1,1-0-2	未实现
	2-2-1	未实现
	2-5-1	未实现
	3-1-1	未实现

	3-2-2,3-2-3	未实现
	3-3-1	未实现
	3-4-1	未实现
	7-1-1,7-1-2	未实现
	10-1-2	未实现
	14-6-1,14-6-2	未实现
	14-7-1~14-7-3	未实现
	14-8-1,14-8-2	未实现
	15-0-1	未实现
	15-3-2	未实现
	15-5-3	未实现
SJ/T 11682-2017 对标检测	6.3.1～6.3.9	未实现
	6.3.12	未实现
	6.8.1	未实现
	6.8.2	未实现

3.3 工具异常处理

表 12：工具异常处理

异常提示	异常处理
IDE 无响应	一般情形下是磁盘速度比较慢导致响应迟缓，可以等待或重启 IDE, 推荐通过配置 IDE 的内存设置进行优化。
点击分析后没有出现进度条，也没有分析结果	确认项目路径是否有中文字符，本工具集暂不支持中文路径。将代码路径改为英文。
分析中有运行崩溃提示	工具集可能存在 bug。本工具以文件为单位分析，一个文件出现分析崩溃不影响其他文件的分析。欢迎反馈此类问题，以帮助我们完善工具。
分析有误报	静态分析无法做到 100%不误报，欢迎反馈此类问题，以帮助我们完善工具。
分析有漏报	静态分析无法做到 100%不漏报，欢迎反馈此类问题，以帮助我们完善工具。
Toolbar 未自动在菜单栏中显示	选择菜单栏中的下拉按钮→添加或移除按钮→自定义；勾选【USTCHCS】后，即可在菜单栏中显示 ToolBar 工具栏
提示语言标准冲突检测	冲突原因:编译配置里有语言标准,每个规则集有自己的语言标准,生成任务时,只要一套语言标准(一个 C 语言标准,以及一个 C++语言标准) 冲突解决方式: 1.编译配置有选项:编译语言标准与规则集语言标准冲突时,若配置为使用项目编译配置,只有一套,无冲突 2.若为使用规则集语言标准,在勾选规则集时提示冲突; 3.若运行时,存在冲突,终止运行。
Client ID 变动，授权失效	可能由磁盘变动（如插入 U 盘、硬盘等）引起，建议移除磁盘，重新启动 VS。

⚠*注：*

对于同一个文件 F，若使用规则集 RA 检测出现 50 个信息，若使用规则集 RB 检测出现 60 条信息，但当同时使用 RA 和 RB 检测时，仅会出现<50+60 条信息。出现此状况的原因为部分规则使用的是 common 规则做代理，当同时运行多个规则集时重复规则仅显示一次。

4. 软件卸载

4.1 软件卸载

工具的卸载步骤如下：

1）打开 VS2013 帮助菜单【工具】中的【扩展和更新】选项；

图 49

2）在【已安装应用】中找到【USTCHCS 分析工具集】，选择【卸载】；

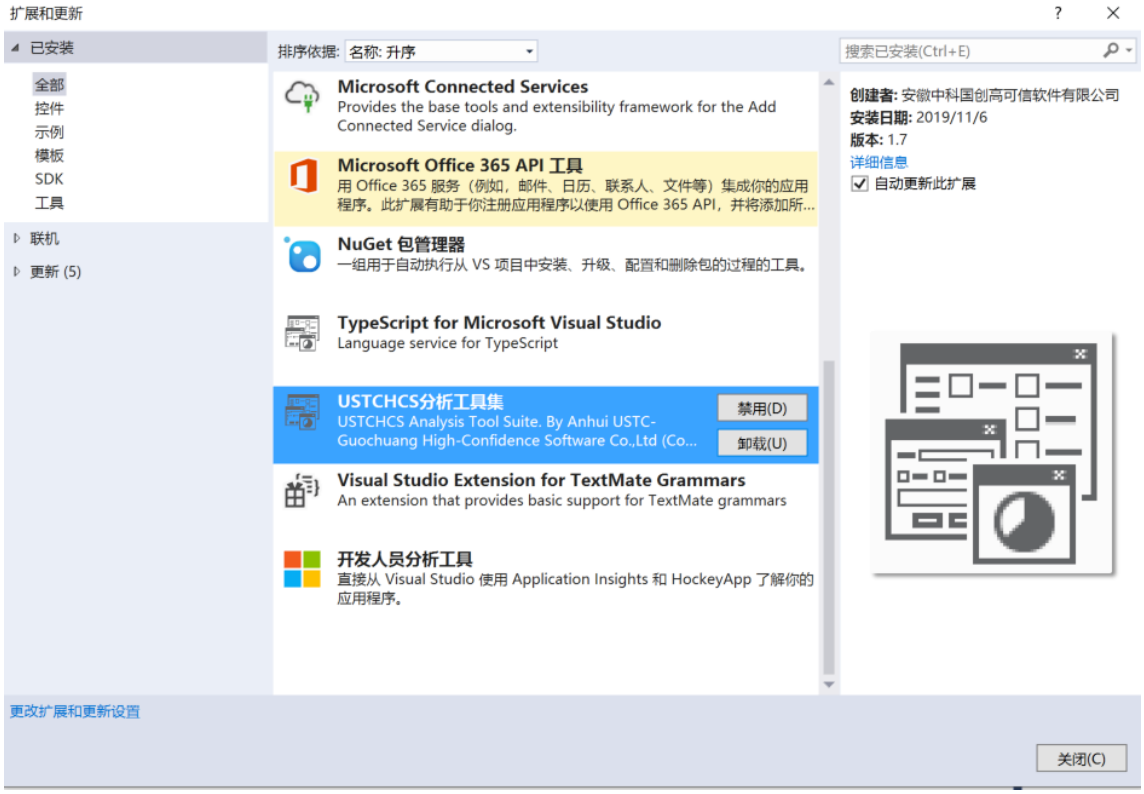

图 50

3）在弹出的对话框中选择【是】，等待片刻后即完成卸载;

图 51

4.2 维护与更新

工具的版本更新步骤如下：

1）打开 VS2013 帮助菜单【工具】中的【扩展和更新】选项；

图 52

2）在【更新】中找到【USTCHCS】，展开后可看到【USTCHCS 分析工具】，选择【更新】；

图 53

3）下载完成后在弹出的对话框中选择【安装】；

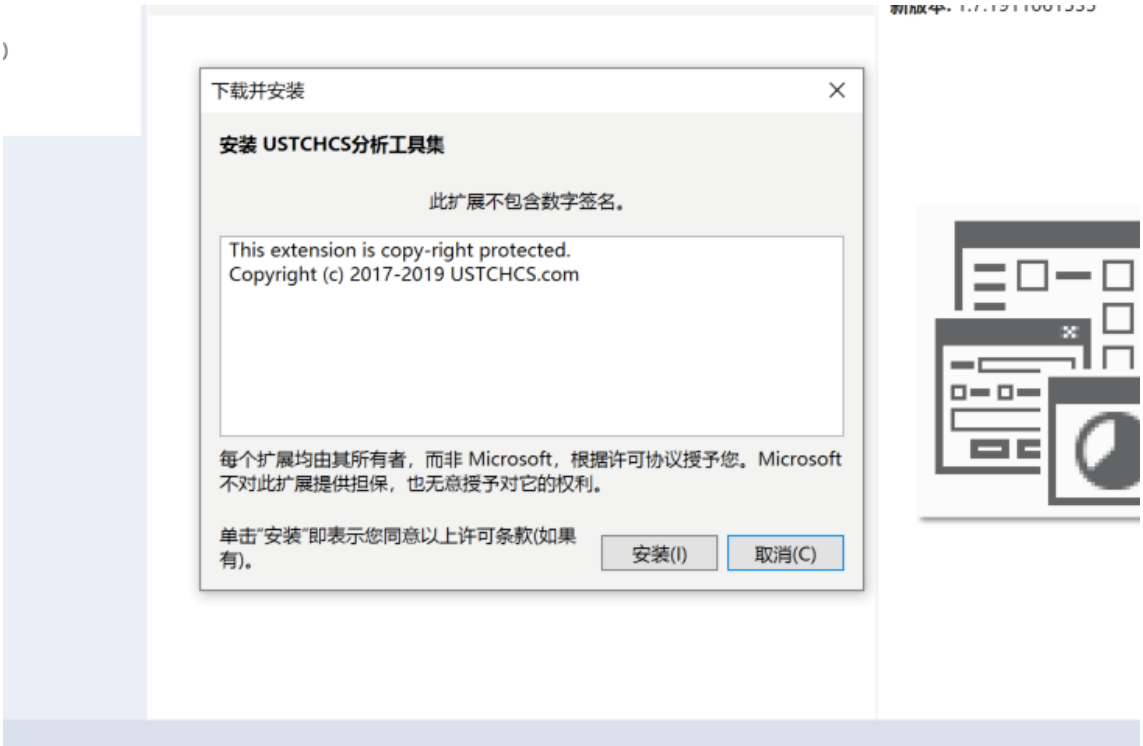

图 54

4）安装完成后立即重新启动即完成 USTCHCS 工具的更新。

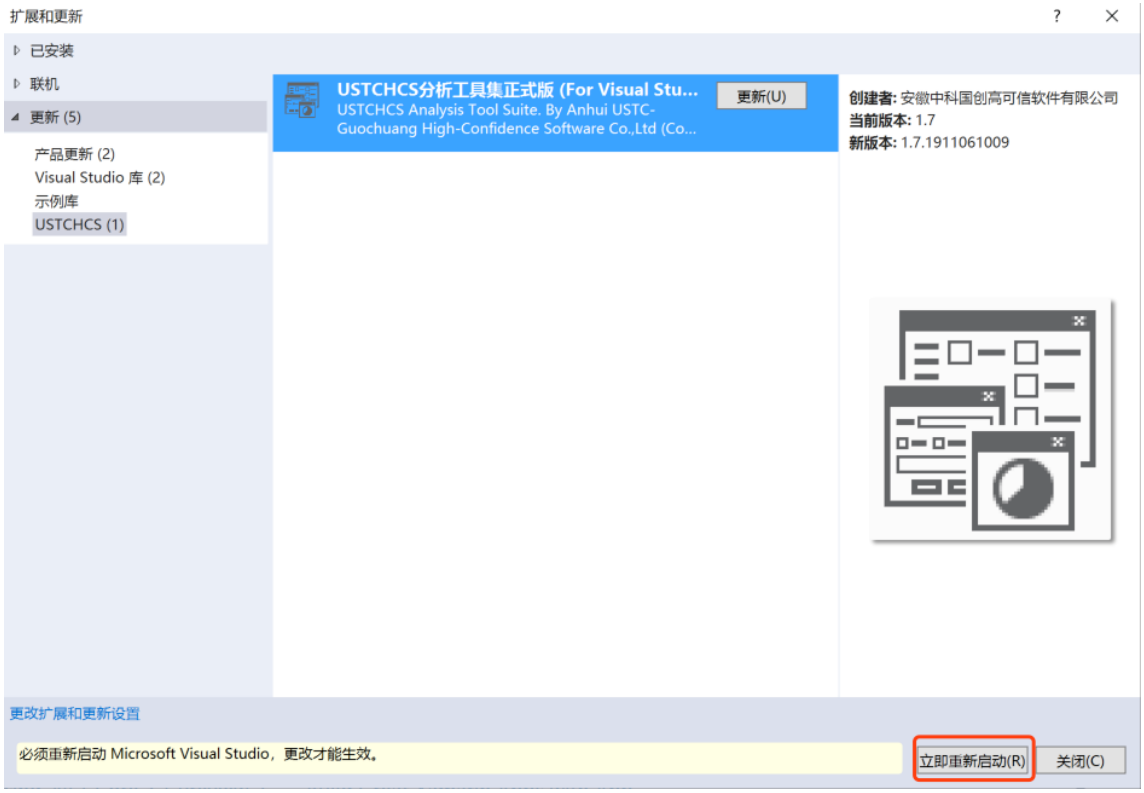

图 55

5. 附录

5.1 插件更新日志

20191108 (Ver 1.7)

新增功能

（1）支持 GJB 5369_2005 等规则集

（2）支持分析范围设置（选定区域）

（3）全新错误列表，支持个性化分组、过滤、标记等等

（4）全新工具栏快捷命令，支持多个工具集、多种分析范围的快捷选择

（5）全新配置页面

（6）全新的菜单设计

（7）排除文件/文件夹的快捷右键菜单

（8）项目语言标准与规则配置语言标准冲突检测与配置

改进功能

（1）改进的编译器警报/错误显示

（2）修复多线程分析中性能/同步问题

（3）规则帮助窗口停靠

（4）授权状态提醒与临期提示

（5）MISRA C:2012/USTCHCS 最佳实践分析/USTCHCS Bugfinder 缺陷分析分析精度和问题改进

20190505 (Ver 1.5)

新增功能

（1）支持 SJ/T、MISRA C，MISRA C++等规则集

（2）支持显示编译器警报/错误

（3）支持分析范围设置（排除文件/文件夹、按时间排除文件）

（4）支持根据 git diff 进行代码差异区域警报显示

（5）支持编译后仅对本次编译的代码进行报警

（6）增加分析进度窗口

（7）改进错误列表，支持警报额外信息跳转（更快排查）

（8）支持带有自动修复选项的错误警报

（9）支持误报标记

（10）支持配置项的导入/导出功能

改进功能

（1）解决插件升级需要卸载再安装的问题

（2）C/C++标准选择改进（可以分别配置）

（3）对菜单项进行易用性优化

（4）错误列表相关警报的显示优化

5.2 分析工具更新日志

20191031[ver 3.0]
【新功能】
* 新增 MISRA C++:2008 2-10-1 规则（Issue #867）
* 新增 MISRA C++:2008 10-3-1 规则（Issue #1192）
* 底层编译框架升级为 LLVM 9.0

【改进项】
* 修正 SJ/T 11682-2017 6.3.11 规则适配 LLVM 9.0 问题（Issue #1179）
* 修正 bestpractice-6.12 规则误报问题(Issue #1219)
* 修正 bestpractice-8.2 规则误报问题(Issue #1145 #1223)
* 修正 bestpractice-9.6 规则误报问题(Issue #1157)
* 修正 bestpractice-11.2 规则误报问题(Issue #1213)
* 修正 GJB 5369-2005 1.1.16 规则误报问题（Issue #1170）
* 优化 GJB 5369-2005 1.2.5 规则分析性能（Issue #1205）
* 优化 GJB 5369-2005 5.1.2 规则实现（MR!1653）
* 优化 GJB 5369-2005 6.1.7 规则实现（MR!1639）
* 优化 GJB 5369-2005 6.2.1 规则实现（MR!1645）
* 优化 GJB 5369-2005 9.1.1 规则实现（Issue #1202）
* 优化 GJB 5369-2005 9.1.4 规则实现（Issue #1166）
* 修正 MISRA C++:2008 0-1-8 规则误报问题（Issue #1171）
* 修正 MISRA C++:2008 0-1-11 规则误报问题（Issue #1173）
* 优化 MISRA C++:2008 2-10-1 规则性能（MR!1637）
* 修正 MISRA C++:2008 5-0-2 规则误报问题（Issue #1186）
* 修正 MISRA C++:2008 5-0-6 规则误报问题（MR!1471）
* 优化 MISRA C++:2008 5-0-7，5-0-8，5-0-9 规则实现（MR!1473）
* 优化 MISRA C++:2008 5-0-13 规则自动修复（Issue #1181）
* 修正 MISRA C++:2008 5-0-20 规则误报问题（Issue #1184）
* 修正 MISRA C++:2008 5-2-6 规则误报问题（MR!1493）
* 优化 MISRA C++:2008 5-2-7 规则实现（MR!1494）
* 修正 MISRA C++:2008 5-3-1 自动修复（Issue #1195）
* 修正 MISRA C++:2008 6-5-2 规则搜索范围问题（MR!1636）
* 修正 MISRA C++:2008 9-3-1 规则误报问题（Issue #1189）
* 修正 MISRA C++:2008 9-3-2 规则误报问题（Issue #1188）
* 修正 MISRA C++:2008 9-3-3 规则误报问题（Issue #1187）
* 增加 MISRA C++:2008 8-0-1 规则诊断额外信息（Issue #1198）
* 修正 MISRA C++:2008 10-3-2 规则误报问题（Issue #1190）
* 修正 MISRA C++:2008 11-0-1 规则误报问题（Issue #1193）
* 修正 MISRA C++:2008 12-1-2 规则误报问题（Issue #1197）
* 修正 MISRA C++:2008 12-8-2 规则误报问题（Issue #1190）
* 修正 MISRA C++:2008 15-0-2 规则误报问题（Issue #1200）
* 优化 MISRA C++:2008 15-3-4 规则性能（MR!1503）
* 修正 MISRA C++:2008 16-0-6 规则误报问题（Issue #1182）
* 优化 MISRA C++:2008 16-2-2 规则实现算法（Issue #1183）

* 优化 MISRA C++:2008 17-0-1，17-0-2，17-0-3 规则实现（MR!1625）
* 优化 MISRA C++:2008 18-7-1，18-4-1 规则实现（MR!1622）
* 重构系统函数判定算法（MR!1624)
* 重构表达式 Expr 相等判定算法，以提高性能（MR!1671)
* 重构禁用宏判定算法（MR!1623)
* 修复 MISRA C2012 D.4.4, Bugfinder-7.1, Bestpactice-13.4 规则文本的打字错误(MR!1634)
* 修复 ForbidNonIntegerSwitchCtrlExpr 误报问题（MR!1617）
* 修复局部符号执行部分误报问题（MR!1618）

20190930[ver 2.9]
【新功能】
* 增加对 MISRA C2012 Dir 规则组的支持(原规则前加 R)
* 新增 MISRA C:2012 标准规则支持: 11.7, 13.2
* 新增 MISRA C++:2008 标准规则支持: 5-0-2
* 支持编译诊断信息存入分析数据库

【改进项】
* 增加 bestpractice-0.2 规则自定义参数(Issue #1102)
* 修正 bestpractice-1.1 规则误报，优化诊断信息问题(Issue #1034)
* 增加 bestpractice-1.2 规则自定义参数(Issue #1100)
* 修正 bestpractice-1.2 规则漏报、误报问题(Issue #1142)
* 修正 bestpractice-4.*规则可配置参数显示问题(MR!1495)
* 修正 bestpractice-5.1 规则诊断信息精读(Issue #1022)
* 修正 bestpractice-5.3 规则自动修复问题(Issue #1042, Issue #1046)
* 修正 bestpractice-6.3 诊断信息重复问题(Issue #1139)
* 修正 bestpractice-6.4 自动修复问题(Issue #1125)
* 修正 bestpractice-6.12 规则配置项缺失问题(Issue #1029)
* 修正 bestpractice-7.1 规则文本，并扩充测试例(MR!1544)
* 修正 bestpractice-7.3 规则算法，并扩充测试例(MR!1547)
* 优化 bestpractice-7.5 规则算法(MR!1552)
* 修正 bestpractice-8.1 规则误报问题(Issue #1090)
* 修正 bestpractice-8.5/8.6 规则自动修复问题(MR!1560，MR!1561)
* 优化 bestpractice-12.3 规则算法(MR!1548)
* 优化 bestpractice-13.2～13.4 规则报警位置(Issue #1105)
* 修正 Bugfinder 5.16 规则误报问题 (Issue #1156)
* 修正 Bugfinder 6.1 规则某些情况自动修复错误问题 (Issue #1137)
* 修正 Bugfinder 6.3 规则漏报问题 (Issue #1026)
* 修正 Bugfinder 7.1 规则文本 (Issue #1140)
* 修正 Bugfinder 7.3 规则误报问题 (Issue #1141)
* 修正 Bugfinder 8.2 规则漏报问题 (Issue #932)
* 修正 Bugfinder 11.5 规则误报问题 (Issue #1011，#1144)
* 修正 GJB5369-2005 15.1.2/15.1.3 规则崩溃问题（MR!1469）
* 修正 MISRA C2012 2.6 规则性能问题(MR!1533)

* 修正 MISRA C2012 2.7 规则自动修复问题(Issue #1040)
* 修正 MISRA C2012 3.1/3.2 规则自动修复错误问题(Issue #1051, Issue #1052)
* 修正 MISRA C2012 4.6 规则自动修复问题(Issue #1044)
* 修正 MISRA C2012 5.2/5.3 规则误报问题(Issue #1055, Issue #1056)
* 修正 MISRA C2012 5.4 规则误报问题(MR!1538)
* 修正 MISRA C2012 7.1 规则自动修复/漏报/崩溃问题(Issue #1058, Issue #1071)
* 修正 MISRA C2012 7.3 规则误报问题(MR!1489)
* 修正 MISRA C2012 7.4 规则误报/性能问题、自动修复位置问题(Issue #1059)
* 修正 MISRA C2012 8.2 规则性能/额外信息问题(MR!1535)
* 修正 MISRA C2012 8.8 规则自动修复位置问题(MR!1536)
* 修正 MISRA C2012 8.10 规则自动修复位置问题(Issue #1063)
* 修正 MISRA C2012 10.3 规则误报问题(Issue #983)
* 修正 MISRA C2012 11.2 规则检测算法不正确问题(MR!1470)
* 修正 MISRA C2012 12.2 规则检测算法不正确问题(MR!1507)
* 修正 MISRA C2012 13.3 规则漏报问题(MR!1511)
* 修正 MISRA C2012 13.4 规则误报、性能问题(MR!1512)
* 修正 MISRA C2012 15.3 规则漏报问题(MR!1514)
* 修正 MISRA C2012 16.1 规则报警位置/报警信息问题(MR!1517)
* 修正 MISRA C2012 17.6 规则自动修复引入 C++特性代码的问题(MR!1523)
* 修正 MISRA C2012 18.4 规则漏报问题(Issue #1057)
* 修正 MISRA C2012 18.5 规则重复报警问题(MR!1531)
* 修正 MISRA C++:2008 5-0-3 规则漏报/误报问题（MR!1453）
* 优化 MISRA C++:2008 5-2-8，5-2-9 规则实现（MR!1501）
* 优化 MISRA C++:2008 8-4-3 规则实现（MR!1524）
* 修正 Common 规则 CheckNonVoidFuncRetvalUsage 自动修复报警位置问题（Issue #1086）
* 修正 Common 规则 ForbidUnsignedConstWrapAround 除零错引发的崩溃问题（Issue #1033）
* 重构 Common 规则 ForbidControlExprIsNotBoolType,优化算法,修复崩溃问题（MR!1465）
* 重构 Common 规则 ForbidStmtWithoutSideEffects 支持 C++、改进性能问题（Issue #1128）
* 重构并改进性能问题 Bestpractice 0.4, 21.2(Issue #1084)
* 重构并改进性能问题 MISRA C:2012 14.1, 21.2(Issue #1070)
* 重构并改进性能问题 MISRA C++:2008 3-2-4（MR!1440), 5-0-5（MR!1459), 5-2-5（MR!1492)
* 重构并改进性能问题 GJB5369-2005 6-2-3（MR!1505）
* 改进使用 Z3 规则的分析精读问题（降低误报）（MR!1464，MR!1595）
* 修正编译环境引入的宏定义导致诊断信息不准确的问题（MR!1539）
* 更新 MISRA C 规则集单元测试（MR!1532）
* 宏展开位置报警优化（Issue #1132)
* 修正 MISRA C/C++规则集规则文本打字错误（MR!1506）
* 修正一处 FreeScale 方言中_asm 关键字支持问题（MR!1543)

20190726[ver 2.8]

【新功能】

* 新增 GJB 2005 标准规则支持: 1.1.19, 1.1.20, 1.1.21, 1.1.22, 1.2.1, 1.2.2, 1.2.3, 1.2.4,
  1.2.5, 1.2.6, 1.2.7, 1.2.8, 1.2.9, 15.1.1, 15.1.2, 15.1.3, 15.1.4, 15.1.5, 15.1.6, 15.2.1, 15.2.2
* 新增 Bestpractice 2.0 标准规则支持: 1.10, 1.11, 1.12, 1.13, 2.3, 4.9, 4.10, 4.11, 4.12,
  5.6, 6.14, 6.15, 7.5, 9.9, 10.4, 13.1, 13.2, 13.3, 13.4, 13.5
* 新增 Bugfinder 2.0 标准规则支持：4.9, 5.6, 5.13, 5.14, 5.15, 5.16, 7.4, 7.5, 9.6, 10.7,
  13.16, 13.17, 13.19, 14.2, 15.1
* 新增 SJ/T 11682-2017 标准规则支持: 6.3.15
* 新增 MISRA C:2012 标准规则支持: 4.4, 4.6
* 新增 MISRA C++:2008 标准规则支持: 2-7-2, 2-7-3, 6-4-7
* 支持 Freescale C 方言

【改进项】

* 为 Bestpractice 1.1 规则误报问题（Issue #883）
* 为 Bestpractice 4.7 规则提供精细化参数控制 (Issue #951)
* 修正 Bestpractice 4.9 规则崩溃问题 (Issue #958)
* 修正 Bestpractice 4.12 规则崩溃问题 (Issue #963)
* 修正 Bestpractice 4.10 规则崩溃问题 (Issue #938)
* 为 Bestpractice 6.1 规则提供精细化参数控制 (Issue #394)
* 为 Bestpractice 6.4 规则提供自动修复功能 (Issue #705)
* 改进 Bestpractice 9.3 支持 C++17 [[fallthrough]]语句 (Issue #943)
* 修正 Bugfinder 1.1 规则误报/漏报问题 (Issue #953)
* 修正 Bugfinder 2.10 规则关于 C++默认参数误报问题（Issue #933）
* 修正 Bugfinder 4.2 规则误报问题（Issue #941）
* 修正 Bugfinder 4.5 规则部分报告重复问题（MR 1433）
* 修正 Bugfinder 5.7 规则误报问题 (Issue #877, #930, #947)
* 修正 Bugfinder 6.2 规则误报问题 (Issue #939)
* 修正 Bugfinder 7.1 规则误报问题 (Issue #948)
* 修正 Bugfinder 8.4 规则，默认支持括号表达式中为赋值语句不报告 (Issue #923)
* 为 Bugfinder 9.3 规则提供精细化参数控制（支持 continue）(Issue #929)
* 优化 MISRA C2012 10.1 规则的提示信息 (Issue #957)
* 修正 MISRA C2012 7.2 规则的实现逻辑并完善预处理代码检测 (Issue #980)
* 为 Bugfinder 11.4 规则提供精细化参数控制 (Issue #970)
* 为 Bugfinder 11.2 规则提供精细化警告信息 (Issue #942)
* 修正 Bugfinder 13.8 规则系统宏位置报告问题 (Issue #985)
* 修正 Bugfinder 13.14 规则误报问题 (Issue #904)
* 修正 Bugfinder 13.18 规则某些情况下崩溃问题 (Issue #1015)
* 修正 SJ/T 11682-2017 6.2.2 规则误报问题（Issue #976）
* 为 SJ/T 11682-2017 6.2.3 规则提供精细化警告信息（Issue #954）
* 修正 SJ/T 11682-2017 6.3.14 规则误报问题（Issue #974）

* 修正 SJ/T 11682-2017 6.4.1 规则对枚举类型的误报问题（Issue #975）
* 修正 SJ/T 11682-2017 6.6.5 规则误报问题（Issue #907）
* 修正 GJ/B 5369-2005 2.1.4 规则误报问题(Issue #919)
* 修正 GJ/B 5369-2005 6.1.9 规则误报问题(Issue #871, #909)
* 修正 GJ/B 5369-2005 13.1.2 规则误报问题(MR 1322)
* 修正 MISRA C:2012 7.2 规则误报问题（Issue #979）
* 优化 MISRA C:2012 10.1 规则的提示信息 (Issue #957)
* 修正 MISRA C:2012 10.4/10.7 规则误报问题（Issue #924）
* 修正 MISRA C:2012 12.4 规则误报问题（Issue #981）
* 修正 MISRA C++:2008 0-1-4 规则误报问题（Issue #875）
* 修正报警时的精细化警告信息可能出现乱码的问题（Issue #1005）
* 修正报警额外提示性信息路径显示问题（MR 1427）
* 修正规则文本翻译打字错误、统一术语等（MR 1401）

20190628[ver 2.7]
【新功能】
* 新增 GJB 2005 标准规则支持: 1.1.3, 1.1.4, 1.1.5, 1.1.6, 1.1.7, 1.1.8, 1.1.9, 1.1.10, 1.1.11, 1.1.12, 1.1.13, 1.1.14, 1.1.15, 1.1.16, 1.1.17, 1.1.18, 1.1.19, 1.1.20, 1.1.21, 1.1.22,1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.2.6, 1.2.7, 1.2.8, 1.2.9, 4.1.2, 4.1.3, 4.2.1, 4.2.2, 6.1.11, 6.1.12, 6.1.13, 6.1.15, 6.1.16, 6.1.17, 6.2.1, 6.2.2, 6.2.3, 6.2.4, 7.1.1, 7.1.2, 7.1.3, 7.1.4, 7.1.5, 7.1.6, 7.1.7, 7.1.8, 7.1.9, 7.1.10, 7.2.1, 7.2.2, 7.2.3, 8.1.1, 8.1.2, 8.1.3, 8.2.1, 8.2.2, 8.2.3, 8.2.4, 8.2.5, 8.2.6, 8.2.7, 8.2.8, 13.1.1, 13.1.2, 13.1.3, 13.1.4, 15.1.1, 15.1.2, 15.1.3, 15.1.4, 15.1.5, 15.1.6, 15.2.1, 15.2.2
* 新增 Bestpractice 2.0 标准规则支持: 6.15, 7.5, 9.9, 10.4, 13.1, 13.2

【改进项】
* 修正 GJB5369-2005 7.1.9 规则误报问题 (Issue #898)
* 修正 GJB5369-2005 8.1.3 规则崩溃问题 (Issue #903)
* 修正 GJB5369-2005 7.1.8 规则误报问题 (Issue #870)
* 修正 GJB5369-2005 1.1.9 规则崩溃问题 (Issue #834)
* 修正 MISRA CPP2008 5-0-3 误报问题（Issue #769）
* 修正 MISRA CPP2008 5-18-1 误报问题（过滤系统文件的报警）
* 修正 SJ/T 11682-2017 6.2.12 规则误报问题（Issue #784）
* 修正 SJ/T 11682-2017 6.2.3 规则误报问题（Issue #790）
* 修正 SJ/T 11682-2017 6.5.13 规则漏报问题
* 完善 Bugfinder 5.5 支持更多问题报告（Issue #786）
* 修正 Bugfinder 10.4 误报问题 （Issue #787）
* 修正 Bugfinder 10.2/10.3 误报问题 （Issue #695）
* 修正 SJ/T 11682-2017 6.5.3 规则误报问题（Issue #788）
* 修正 SJ/T 11682-2017 6.3.14 规则误报问题（Issue #789）
* 修正 Bugfinder 4.5 误报问题 （Issue #793）
* 系统规则重构，修正系统规则在 Windows 系统下误报/漏报问题（Issue #794）

* 修正 MISRA C2012 15.6 误报问题（Issue #886）
* 修正一处由于获取不到正确文件名引起的错误报告不可定位问题
* 修复 GJB 2005 标准规则 6.1.9 误报问题（Issue #871,#909)
* 移除了不再使用的类 MultiLocationDiagnostic
* 修复 MISRA cpp2008 0-1-4 误报问题（#875）
* 修复 Bestpractice 1.1 误报问题（#883）
* 报警问题重构，针对每个规则可自定义是否对于系统的宏展开进行报警（默认不报警）。

20190531[ver 2.6]
【新功能】
* 新增 GJB 2005 标准规则支持:3.1.1, 3.1.2, 3.1.3, 3.1.6, 3.1.7, 3.1.8, 5.1.1, 5.2.1, 6.1.1
6.1.2, 6.1.3, 6.1.4 6.1.5, 6.1.6, 6.1.7, 6.1.8, 6.1.9, 6.1.10, 6.1.11, 6.1.12, 6.1.13, 6.1.15, 6.1.16, 6.1.17, 6.2.1, 6.2.2, 6.2.3, 6.2.4, 9.1.1, 9.1.2, 9.1.3, 9.1.4, 9.1.5, 10.1.1, 10.2.2, 11.1.1, 11.1.2, 11.2.2, 11.2.3, 12.1.1, 12.2.1, 12.2.2, 12.2.3, 14.1.1, 14.1.2, 14.1.3, 14.2.1,
* 新增 GJB 2005 标准规则 6.1.5, 6.1.4，通过符号执行技术实现
* 新增 MISRA CPP2008 5-0-16、5-0-17、5-0-18、5-2-3
* 新增 common ForbidControlExprIsNotBoolType, ForbidNonIntegerSwitchCtrlExpr, ForbidShiftOperationOverflow, ForbidSingleBitSignedTypedFieldDecl
* 修改 common ForbidArrayOutOfBoundAccess

【改进项】
* 系统规则修正
* 修正代理规则一个实现错误
* 修正自动修复提示数据库中存储偏移出错问题的 bug
* 修正辅助类 ExprUtils 中处理 ValueDependent 表达式时崩溃问题 (Issue #711/#714)
* 修正 MISRA CPP2008 2-13-3 误报问题 (Issue #768)
* 修正 GJB5369-2005 3.1.4 规则复用 CheckSwitchNoDefaultStmt（Issue #704）
* 修正 sjt11682-2017 6.2.16 规则误报问题（Issue #755）
* 修正 sjt11682-2017 6.3.14 规则误报问题（Issue #760）
* 修复 sjt11682-2017 6.3.20 规则崩溃问题（Issue #754）
* 修正 sjt11682-2017 6.4.1 规则误报问题（Issue #761）
* 修正 sjt11682-2017 6.5.23 规则误报问题（Issue #759）
* 修正 sjt11682-2017 6.6.4 规则误报问题（Issue #748）
* 修正 Bugfinder 4.2 规则误报问题（Issue #749
* 修复 Bugfinder 5.12 规则崩溃问题（Issue #739/#745）
* 修正 Bestpractice 10.2 规则文本问题
* 修正 Bestpractice 6.12 规则自动修复位置问题 (Issue #756)
* 修正 Bestpractice 99.X 系列规则误报问题 (Issue #613/#633)

20190426[ver 2.5]
【新功能】
* 新增 SJ/T 11682-2017 标准规则支持: 6.2.3, 6.2.7, 6.2.13, 6.2.17, 6.3.18, 6.3.20, 6.3.21, 6.4.1,
6.4.3, 6.4.4, 6.4.5, 6.5.17, 6.5.18, 6.5.24, 6.5.25, 6.5.26, 6.5.27, 6.5.29, 6.6.4, 6.7.1
* 新增 GJB 2005 标准规则支持: 2.1.2, 2.1.3, 2.1.4, 2.1.5, 2.1.6, 2.1.7, 2.1.9, 2.1.10, 2.2.1,
2.2.2, 3.1.4, 3.1.5, 5.1.2, 11.2.1
* 新增 Bugfinder 3.1, 3.2, 5,12, 11.5, 13.9, 14.1
* 新增 Bestpractice 8.3, 12.2
* 新增 MISRA C2012 9.1
* 新增 common CheckAsmIsolatedEncapsulation

【改进项】
* 修正 CI Test Language Std 检测（Issue #584）
* 修正 MISRA CPP2008 6.6.2 测试用例注释书写有误（Issue #584）
* 修正 Bugfinder 2.1 关于 goto 语句检测误报问题（Issue #548）
* 修正 Bugfinder 4.2 关于问号表达式检测误报/漏报问题（Issue #749）
* 修正 Bugfinder 5.11 关于匿名 enum 类型相关误报问题（Issue #543）
* 修正 Bugfinder 5.12 关于字符串常量的误报问题及处理依赖类型表达式的崩溃问题（Issue #739，#745）
* 修正 Bugfinder 6.1 误报和修复提示问题（Issue #458）
* Bugfinder 9.3 关支持用户自定义的路径终止函数（Issue #459）
* 修正 Bugfinder 10.3 关于 C++迭代器循环的相关误报问题（Issue #490）
* 修正 Bestpractice 1.1 关于匿名 enum 类型的误报和漏报问题（Issue #446）
* 修正 Bestpractice 1.9 关于字符串常量相关误报（Issue #535）
* 修正 Bestpractice 8.7 关于 callExpr（返回类型为指针）中，对于参数 0 的相关误报（Issue #662）
* 修正 Bestpractice 9.6 关于类成员初始化列表的相关误报（Issue #528）
* 修复 SJT 6.5.3 分析含有 128 位整数类型程序时的崩溃问题（Issue #650）
* 修复 SJT 6.5.23 分析依赖类型相关代码的崩溃问题（Issue #602）
* 修复 SJT 6.5.13 分析条件表达式为空表达式时的崩溃问题（Issue #598, #604）
* 修复 SJT 6.6.1 分析崩溃问题（Issue #651）
* Common: 修正类函数宏引入的 return 后空语句报警（Issue #549)
* Common: 修正由于 Z3 限制引入的误报问题（Issue #451, #603)
* Common: 更新关于无符号整型的回绕检测，涉及规则包括：MISRA CPP2008:5-9-1, MISRA C2012:12.4,
以及 Bugfinder:6.2
* Common: 优化了关于 Switch 语句检测是否覆盖所有枚举值的误报情况，并加入了提示信息，提示没有覆盖的枚举值

20190331[ver 2.4]

【新功能】
* 新增 SJ/T 11682-2017 标准规则支持: 6.2.1, 6.2.2, 6.2.4, 6.2.5, 6.2.6, 6.2.14, 6.5.2, 6.5.9,
6.5.11, 6.5.13, 6.5.16, 6.5.19, 6.6.1, 6.6.2, 6.6.3,

20190319[ver 2.3]
【新功能】
* 新增 SJ/T 11682-2017 标准规则支持:6.5.1, 6.5.3, 6.5.5~6.5.8, 6.5.10, 6.5.22, 6.6.4, 6.6.5,
6.7.2 （Issue #364）
* Bestpactice 1.2～1.7 规则实现
【改进项】
* Bugfinder 规则文本修正（Issue #488）

20190228[ver 2.2]
【新功能】
* 新增 SJ/T 11682-2017 标准规则支持:6.2.1, 6.2.8, 6.2.10~6.2.12, 6.2.14, 6.2.16，6.3.10,
6.3.13, 6.3.14, 6.3.17, 6.5.23, 6.5.27, 6.5.32, 6.6.5 （Issue #364）
* 新增 Bestpractice 规则 4.8, 98/99 类规则支持 （Issue #475）
【改进项】
* Bestpractice 4.4 误报修复（Issue #488）
* Bestpractice 4.8 崩溃修复（Issue #486）
* MISRA CPP2008 8-4-3 规则文本修正

20190218[ver 2.1]
【新功能】
* Bestpractice 第 4 章规则底层支持及 4.5～4.7 规则实现（Issue #475）
【改进项】
* MISRA C2012 10.X 崩溃修复（Issue #472、#477、#484、¥487）
* Bestpractice 4.2 崩溃修复（Issue #486）

20190131[ver 2.0]
【新功能】
* Bestpractice 第 4 章规则底层支持及 4.1～4.4 规则实现（Issue #318）
* Bestpractice 5.3 支持用户自定义路径 Terminator 函数（Issue #278）
* MISRA CPP2008 系统规则重现实现 （Issue #351）
* MISRA C2012 14.3/MISRA CPP2008 0-1-2/Bestpractice 6.3 规则实现（Issue #423）

* Bugfinder 4.5/5.11 新增提示性信息（Issue #334、#370、#380）
【改进项】
* 工具授权提示信息优化
* 共用规则 CheckIncompatibleFuncDecl 改进（Issue #437）
* Bestpractice/Bugfinder 单元测试改进（Issue #390）
* MISRA CPP2008 6-6-5 报警信息改进（Issue #320）
* MISRA CPP2008 8-5-2 误报漏报修复（Issue #333）
* MISRA C2012 16.4 误报漏报修复（Issue #343）
* Bugfinder 5.11/7.1/11.4 性能改进（Issue #433、#440）
* Bugfinder 4.5/5.3/5.8/5.11 崩溃修复 （Issue #428、#437）
* Bestpractice 0.2/0.3/0.4/8.7 误报漏报修复（Issue #432、#443）
* Bestpractice 6.2/6.6/8.9/8.7/8.11/0.2/3.1 性能改进（Issue #441、#449）
* MISRA C2012 10.6 崩溃修复 （Issue #430）
* MISRA CPP2008 0-1-10/5-0-3 崩溃修复 （Issue #439、#456）

20190117[ver 1.9]
【新功能】
* MISRA C2012 系统规则 2.3/2.4/5.1/5.8/5.9/8.3/8.5/8.7 的重新实现
* Bestpractice 新增规则 6.7/6.10/6.11/7.3/7.4 （Issue #317）
* Bugfinder 新增规则 11.3/11.4/12.1/12.2/12.3（Issue #407、#408）
【改进项】
* MISRA CPP2008 CValue 相关规则/5-0-10 重构
* MISRA CPP2008 Essential Type 相关规则的重构（Issue #350）
* MISRA CPP2008 5-0-3/9-3-3/7-4-2 误报漏报修复（Issue #347、#401、#383）
* MISRA C2012 14.4 修复提示的改进（Issue #347）
* Bestpractice 1.9 报警信息改进 （Issue #395）
* Bestpractice 2.2/6.6/8.7 误报、漏报修复 （Issue #241、#398、#403）
* Bugfinder 2.X/4.8/7.2/13.6 报警信息改进 （Issue #369、#373、#374、#375、#377）
* Bugfinder 4.4/4.5/5.3 误报修复 （Issue #392、#376、#378、#406、#396）
* Bugfinder 2.9/2.10/11.6 崩溃修复 （Issue #404、#410、#424）
* Common 规则误报修复（Issue #317、#402）
* 各规则集规则增加严重程度、适用语言、系统规则与否、性能指标等属性，便于 GUI 自定义过滤。（Issue #393）

20190103[ver 1.8]
【新功能】
* MISRA C2012/CPP2008/Bestpractice/Bugfinder 修复提示
【改进项】
* MISRA CPP2008/MISRA C2012/Bestpractice/Bugfinder 若干误报修复
* Bugfinder 规则在 Linux/Windows 平台下崩溃问题修复

* Bugfinder/Bestpractice/MISRA CPP2008/MISRA C2012 性能提升重构
* 改进 Bugfinder/Bestpractice/MISRA C/MISRA CPP 检测机制
* 各规则集修复提示的改进

20181220[ver 1.7]
【新功能】
* Bugfinder 新增 2.9、2.12、4.9、4.10、10.2～10.6、13.9～13.11、13.13 规则支持；
* Bugfinder 4.2 规则，支持指针参数后标注/*@non-null*/，参数名和标注之间可以有任意空格（同一行）。
* MISRA C2012/CPP2008/Bestpractice/Bugfiner 更多修复提示
【改进项】
* MISRA CPP2008/MISRA C2012/Bestpractice/Bugfinder 若干误报修复
* MISRA C2012 4.1 规则在 Windows 平台下崩溃问题修复
* MISRA CPP2008 性能提升重构
* 修复提示的改进

20181207[ver 1.6]
【新功能】
* Bugfinder 新增 2.10、2.11、13.12 规则支持；
* MISRA C2012/CPP2008 修复提示
* Bestpractice 修复提示
* Bugfiner 修复提示
【改进项】
* Bestpractice 若干崩溃问题、误报修复
* MISRA CPP2008 若干崩溃问题、误报修复
* MISRA C2012 误报修复
* Bugfinder 误报修复
* 授权算法问题修复

20181118[ver 1.5]
【新功能】
* MISRA C2012 新增 1.1、4.1 规则支持；
* Bestpractice 规则集第一版
* Bugfinder 缺陷检测第一版
* 新的授权机制
【改进项】
* MISRA C2012 若干崩溃问题
* MISRA CPP2008 若干崩溃问题