MISRA C:2012

MISRA C:2012 是由汽车产业软件可靠性协会 (MISRA) 提出的 C 语言开发标准。MISRA C:2012 的第三版发布于 2019 年 2 月，整合了此前发布的 AMD1、TC1 的内容，包含 173 条规则。MISRA C:2012 AMD2 发布于 2020 年，增加了 2 条新的规则，支持 ISO/IEC 9899:2011。

NaiveSystems Analyze 支持绝大多数 MISRA C:2012 规则的静态检查，包括 AMD1 和 AMD2 的内容。注意：在 175 条规则中，有 6 条规则是不完全适用于静态检查的，包括 Dir 1.1、Dir 2.1、Dir 3.1、Dir 4.1、Dir 4.2 和 Rule 1.1。

MISRA C:2012 规则覆盖情况

	可判定		不可判定		小计		覆盖比例
	支持	总计	支持	总计	支持	总计	
所有类别	121	121	36	54	157	175	90%
强制 (Mandatory)	5	5	11	11	16	16	100%
要求 (Required)	88	88	23	32	111	120	93%
建议 (Advisory)	28	28	2	11	30	39	77%

MISRA C:2012 规则支持情况

Analyze 编码	规则编码	规则名称	类别	可判定性	是否支持
C2301	Dir 1.1	如果程序的输出取决于某实现定义的行为，则必须记录并理解该行为	要求	不可判定	否
C2302	Dir 2.1	所有源文件编译过程中不得有编译错误	要求	不可判定	否
C2303	Dir 3.1	所有代码必须可被追溯到文档化的需求	要求	不可判定	否
C2304	Dir 4.1	必须尽量减少运行错误	要求	不可判定	否
C2305	Dir 4.2	应记录所有汇编语言的使用	建议	不可判定	否
C2306	Dir 4.3	汇编语言必须被封装并隔离	要求	不可判定	否
C2307	Dir 4.4	不应“注释掉”代码段	建议	不可判定	否
C2308	Dir 4.5	在同一命名空间内，应确保外形重合的标识符的排版不易混淆	建议	不可判定	否
C2309	Dir 4.6	应使用表示大小和符号性的类型定义（typedef）代替基本数据类型	建议	不可判定	否
C2310	Dir 4.7	如果函数返回了错误信息，那么必须检测该错误信息	要求	不可判定	否
C2311	Dir 4.8	如果一个翻译单元内，指向结构体或联合体的指针永不被解引用，那么应该隐藏该对象的实现	建议	不可判定	否
C2312	Dir 4.9	在可以使用函数或类函数宏的情况下，应优先使用函数	建议	不可判定	否
C2313	Dir 4.10	应采取措施预防头文件的内容被多次包含	要求	不可判定	是
C2314	Dir 4.11	必须检查传递给库函数的值的有效性	要求	不可判定	否
C2315	Dir 4.12	不得使用动态内存分配	要求	不可判定	是
C2316	Dir 4.13	应以适当顺序调用对资源进行运算的函数	建议	不可判定	否
C2317	Dir 4.14	应检查来源于外部的值的有效性	要求	不可判定	否
C2201	Rule 1.1	程序不得违反C语言标准语法和约束，不得超出实现的翻译限制	要求	可判定	是
C2202	Rule 1.2	不应使用语言扩展	建议	不可判定	否
C2203	Rule 1.3	不得出现未定义或严重的未指定行为	要求	不可判定	否
C2204	Rule 1.4	不得使用新涌现的语言特性	要求	可判定	是
C2007	Rule 2.1	项目不得含有不可达代码	要求	不可判定	是
C2006	Rule 2.2	不得有死代码	要求	不可判定	是
C2005	Rule 2.3	项目不应含有未使用的类型声明	建议	可判定	是
C2004	Rule 2.4	项目不应含有未使用的标签（tag）声明	建议	可判定	是
C2003	Rule 2.5	项目不应含有未使用的宏声明	建议	可判定	是
C2002	Rule 2.6	项目不应含有未使用的标记（label）声明	建议	可判定	是
C2001	Rule 2.7	函数中不应有未使用的形参	建议	可判定	是

Analyze 编码	规则编码	规则名称	类别	可判定性	是否支持
C2102	Rule 3.1	不得在注释中使用字符序列/*和//	要求	可判定	是
C2101	Rule 3.2	不得在//注释中使用行拼接	要求	可判定	是
C1002	Rule 4.1	八进制和十六进制转义序列必须被终止	要求	可判定	是
C1001	Rule 4.2	不应使用三字母词（trigraphs）	建议	可判定	是
C1109	Rule 5.1	不得使用重名的外部标识符	要求	可判定	是
C1108	Rule 5.2	在同一作用域和命名空间内声明的标识符不得重名	要求	可判定	是
C1107	Rule 5.3	内部作用域声明的标识符不得隐藏外部作用域声明的标识符	要求	可判定	是
C1106	Rule 5.4	宏标识符不得重名	要求	可判定	是
C1105	Rule 5.5	标识符不得与宏的名称重名	要求	可判定	是
C1104	Rule 5.6	类型定义（typedef）名称必须是唯一标识符	要求	可判定	是
C1103	Rule 5.7	标签名称必须是唯一标识符	要求	可判定	是
C1102	Rule 5.8	必须使用唯一标识符定义含有外部链接的对象或函数	要求	可判定	是
C1101	Rule 5.9	应使用唯一标识符定义含有内部链接的对象或函数	建议	可判定	是
C0702	Rule 6.1	只得使用合适的类型来声明位域（bit-fields）	要求	可判定	是
C0701	Rule 6.2	用一位命名的位域不得为有符号类型	要求	可判定	是
C0904	Rule 7.1	不得使用八进制常量	要求	可判定	是
C0903	Rule 7.2	所有表现为无符号类型的整型常量都必须使用“u”或“U”后缀	要求	可判定	是
C0902	Rule 7.3	小写字母“l”不得用作字面量后缀	要求	可判定	是
C0901	Rule 7.4	不得将字符串字面量赋值给对象，除非对象类型为“指向 const 修饰的 char 的指针”	要求	可判定	是
C0514	Rule 8.1	必须明确指定类型	要求	可判定	是
C0513	Rule 8.2	函数类型必须为带有命名形参的原型形式	要求	可判定	是
C0512	Rule 8.3	对同一对象或函数的所有声明必须使用同样的名字和类型修饰符	要求	可判定	是
C0511	Rule 8.4	对含有外部链接的对象或函数进行定义时，必须有可见的兼容声明	要求	可判定	是
C0510	Rule 8.5	外部对象或函数只得在一个文件里声明一次	要求	可判定	是
C0509	Rule 8.6	含有外部链接的标识符必须有且只有一个外部定义	要求	可判定	是
C0508	Rule 8.7	不应使用外部链接定义仅在一个翻译单元中引用的函数和对象	建议	可判定	是
C0507	Rule 8.8	对含有内部链接的对象和函数进行的所有声明都必须使用静态（static）存储类说明符	要求	可判定	是
C0506	Rule 8.9	如果对象标识符只在一个函数中出现，那么应该在块作用域（block scope）中定义该对象	建议	可判定	是
C0505	Rule 8.10	必须使用静态存储类别声明内联函数	要求	可判定	是
C0504	Rule 8.11	对含有外部链接的数组进行定义时，应显式指定其大小	建议	可判定	是
C0503	Rule 8.12	枚举列表里一个隐式指定的枚举常量的值应是唯一的	要求	可判定	是
C0502	Rule 8.13	指针应尽量指向const修饰的类型	建议	不可判定	否
C0501	Rule 8.14	不得使用restrict类型修饰符	要求	可判定	是
C1205	Rule 9.1	对于具有自动存储周期的对象，不得在设定它的值之前读取它的值	强制	不可判定	是
C1204	Rule 9.2	聚合或联合体的初始化器应包含在大括号“{}”内	要求	可判定	是
C1203	Rule 9.3	不得对数组进行部分初始化	要求	可判定	是
C1202	Rule 9.4	最多只得初始化一次对象的元素	要求	可判定	是
C1201	Rule 9.5	对数组对象进行指定初始化时，必须显式指定数组大小	要求	可判定	是
C0808	Rule 10.1	操作数不得为不合适的基本类型	要求	可判定	是

Analyze 编码	规则编码	规则名称	类别	可判定性	是否支持
C0807	Rule 10.2	不得在加减运算中不恰当地使用基本字符类表达式	要求	可判定	是
C0806	Rule 10.3	表达式的值不得赋给更窄的基本类型，也不得赋给不同的基本类型类别	要求	可判定	是
C0805	Rule 10.4	执行寻常算术转换的运算符的两个操作数必须属于同一基本类型类别	要求	可判定	是
C0804	Rule 10.5	表达式的值不应强制转换为不合适的基本类型	建议	可判定	是
C0803	Rule 10.6	复合表达式的值不得赋给具有更宽基本类型的对象	要求	可判定	是
C0802	Rule 10.7	寻常算术转换中，如果运算符的一个操作数为复合表达式，则另一个操作数不得具有更宽类型	要求	可判定	是
C0801	Rule 10.8	复合表达式的值不得强制转换为不同基本类型类别或更宽类型	要求	可判定	是
C1409	Rule 11.1	指向函数的指针和任何其他类型之间不得相互转换	要求	可判定	是
C1408	Rule 11.2	指向不完整类型的指针和任何其他类型之间不得相互转换	要求	可判定	是
C1407	Rule 11.3	指向对象类型的指针和指向不同对象类型的指针之间不得强制转换	要求	可判定	是
C1406	Rule 11.4	指向对象的指针和整数类型之间不应相互转换	建议	可判定	是
C1405	Rule 11.5	指向void的指针不应转换为指向对象的指针	建议	可判定	是
C1404	Rule 11.6	指向void的指针和算术类型之间不得强制转换	要求	可判定	是
C1403	Rule 11.7	指向对象的指针和非整数类型的算术类型之间不得强制转换	要求	可判定	是
C1402	Rule 11.8	强制转换不得移除指针所指向类型的任何const或volatile修饰	要求	可判定	是
C1401	Rule 11.9	宏NULL必须为整数类型空指针常量的唯一允许形式	要求	可判定	是
C0605	Rule 12.1	应明确表达式中操作数的优先级	建议	可判定	是
C0604	Rule 12.2	移位运算符的右操作数的范围下限为零，上限须比左操作数的基本类型的位宽度小一	要求	不可判定	是
C0603	Rule 12.3	不得使用逗号运算符（,）	建议	可判定	是
C0602	Rule 12.4	对常量表达式进行求值不应导致整数回绕	建议	可判定	是
C0601	Rule 12.5	sizeof运算符的操作数不得是声明为“数组类型”的函数形参	强制	可判定	是
C1606	Rule 13.1	初始化器列表不得含有持续的副作用（persistent side effect）	要求	不可判定	是
C1605	Rule 13.2	采用不同的求值顺序时（只要允许采用该顺序），表达式的值和表达式的持续的副作用必须相等	要求	不可判定	是
C1604	Rule 13.3	含有一个自增（++）或自减（--）运算符的完整表达式，除因自增或自减运算符引起的副作用外，不应含有其他潜在副作用	建议	可判定	是
C1603	Rule 13.4	不得使用赋值运算符的结果	建议	可判定	是
C1602	Rule 13.5	逻辑与（&&）和逻辑或（||）运算符的右操作数不得含有持续的副作用	要求	不可判定	是
C1601	Rule 13.6	sizeof运算符的操作数不得包含任何有潜在副作用的表达式	强制	可判定	是
C1704	Rule 14.1	循环计数器不得为基本浮点类型	要求	不可判定	是
C1703	Rule 14.2	for循环必须格式良好	要求	不可判定	是
C1702	Rule 14.3	控制表达式不得为不变量	要求	不可判定	是
C1701	Rule 14.4	if语句和迭代语句的控制表达式必须为基本布尔类型	要求	可判定	是
C1807	Rule 15.1	不应使用goto语句	建议	可判定	是
C1806	Rule 15.2	同一函数中，goto语句只得跳转到在其后声明的标记（label）	要求	可判定	是
C1805	Rule 15.3	goto语句引用的标记必须在同一代码块或上级代码块中声明	要求	可判定	是
C1804	Rule 15.4	对于任何迭代语句，最多只应使用一个break或goto语句进行终止	建议	可判定	是
C1803	Rule 15.5	函数结尾应只有一个退出点	建议	可判定	是
C1802	Rule 15.6	迭代语句或分支语句的主体必须为复合语句	要求	可判定	是
C1801	Rule 15.7	所有if … else if构造都必须以一个else语句终止	要求	可判定	是
C1907	Rule 16.1	所有switch语句必须格式良好（well-formed）	要求	可判定	是

Analyze 编码	规则编码	规则名称	类别	可判定性	是否支持
C1906	Rule 16.2	switch标记只得出现在形成switch语句主体的复合语句最外层	要求	可判定	是
C1905	Rule 16.3	每个switch子句（switch-clause）都必须以一个无条件break语句终止	要求	可判定	是
C1904	Rule 16.4	每个switch语句都必须有default标记	要求	可判定	是
C1903	Rule 16.5	在switch语句中，default标记必须是第一个或最后一个switch标记	要求	可判定	是
C1902	Rule 16.6	每个switch语句都必须有两个或以上switch子句	要求	可判定	是
C1901	Rule 16.7	switch表达式不得是基本布尔类型	要求	可判定	是
C1508	Rule 17.1	不得使用<stdarg.h>的特性	要求	可判定	是
C1507	Rule 17.2	函数不得直接或间接调用自身	要求	不可判定	是
C1506	Rule 17.3	不得隐式声明函数	强制	可判定	是
C1505	Rule 17.4	所有函数退出路径，如果为非空（non-void）返回类型，则必须有一个包含表达式的显式return语句	强制	可判定	是
C1504	Rule 17.5	如果函数形参声明为数组类型，其对应的实参必须具有适当数量的元素	建议	不可判定	是
C1503	Rule 17.6	声明数组形参时，[ ]内不得包含关键字static	强制	可判定	是
C1502	Rule 17.7	函数返回值若不为非空返回类型（non-void return type），则必须被使用	要求	可判定	是
C1501	Rule 17.8	不应修改函数形参	建议	不可判定	是
C1308	Rule 18.1	对指针操作数进行算术运算得来的指针只得用于寻址同一数组的元素	要求	不可判定	是
C1307	Rule 18.2	指针之间的减法运算只得用于寻址同一数组元素的指针	要求	不可判定	是
C1306	Rule 18.3	大小比较运算符>，>=，<和<=不得用于指针类型的对象，除非两个指针指向同一对象	要求	不可判定	是
C1305	Rule 18.4	+, -, +=和-=运算符不得用于指针类型的表达式	建议	可判定	是
C1304	Rule 18.5	声明应含有最多两层嵌套指针	建议	可判定	是
C1303	Rule 18.6	不得将自动存储对象的地址复制给在该对象不复存在后仍然存在的另一个对象	要求	不可判定	是
C1302	Rule 18.7	不得声明灵活数组成员（flexible array members）	要求	可判定	是
C1301	Rule 18.8	不得使用变长数组（variable-length array）	要求	可判定	是
C0302	Rule 19.1	不得将对象赋值或复制给与其重叠的对象	强制	不可判定	是
C0301	Rule 19.2	不应使用关键字union	建议	可判定	是
C0114	Rule 20.1	#include指令之前仅应出现预处理指令或注释	建议	可判定	是
C0113	Rule 20.2	头文件名中不得出现字符'，"或\，以及字符序列/*和//	要求	可判定	是
C0112	Rule 20.3	#include指令后面必须是<filename>或"filename"序列	要求	可判定	是
C0111	Rule 20.4	定义宏名称时不得与关键字同名	要求	可判定	是
C0110	Rule 20.5	不得使用#undef	建议	可判定	是
C0109	Rule 20.6	宏实参中不得有形似预处理指令的词符	要求	可判定	是
C0108	Rule 20.7	宏形参扩展得到的表达式必须在括号内	要求	可判定	是
C0107	Rule 20.8	预处理指令#if或#elif的控制表达式求值结果必须为0或1	要求	可判定	是
C0106	Rule 20.9	预处理指令#if或#elif的控制表达式中的所有标识符必须被#define定义才能求值	要求	可判定	是
C0105	Rule 20.10	不应使用预处理运算符#和##	建议	可判定	是
C0104	Rule 20.11	如果宏形参后面紧跟#运算符，则不得再紧跟##运算符	要求	可判定	是
C0103	Rule 20.12	用作 # 或 ## 运算符的操作数的宏形参，如果自身需要进一步进行宏替换，则只得作为 # 或 ## 的操作数使用	要求	可判定	是
C0102	Rule 20.13	以#开始的代码行必须为有效预处理指令	要求	可判定	是

Analyze 编码	规则编码	规则名称	类别	可判定性	是否支持
C0101	Rule 20.14	所有预处理指令#else，#elif和#endif都必须和它们对应的#if，#ifdef和#ifndef指令位于同一文件中	要求	可判定	是
C0420	Rule 21.1	#define和#undef不得用于保留标识符（reserved identifier）或保留宏名称（reserved macro name）	要求	可判定	是
C0419	Rule 21.2	不得声明保留标识符（reserved identifier）和保留宏名称（reserved macro name）	要求	可判定	是
C0418	Rule 21.3	不得使用<stdlib.h>提供的内存分配和回收（deallocation）函数	要求	可判定	是
C0417	Rule 21.4	不得使用标准头文件<setjmp.h>	要求	可判定	是
C0416	Rule 21.5	不得使用标准头文件<signal.h>	要求	可判定	是
C0415	Rule 21.6	不得使用标准库输入/输出函数	要求	可判定	是
C0414	Rule 21.7	不得使用<stdlib.h>提供的标准库函数atof，atoi，atol，以及atoll函数	要求	可判定	是
C0413	Rule 21.8	不得使用<stdlib.h>提供的标准库终止函数（termination function）	要求	可判定	是
C0412	Rule 21.9	不得使用<stdlib.h>提供的标准库函数bsearch和qsort函数	要求	可判定	是
C0411	Rule 21.10	不得使用标准库中的时间（time）和日期（date）函数	要求	可判定	是
C0410	Rule 21.11	不得使用标准头文件<tgmath.h>	要求	可判定	是
C0409	Rule 21.12	不应使用<fenv.h>的异常处理特性	建议	可判定	是
C0408	Rule 21.13	传递给<ctype.h>函数的值必须能够表示为无符号char或EOF类型	强制	不可判定	是
C0407	Rule 21.14	不得使用标准库函数 memcmp 比较空终止字符串（null terminated string）	要求	不可判定	是
C0406	Rule 21.15	指向标准库函数memcpy，memmove和memcmp的指针实参必须全部为指向兼容类型的限定或非限定版本的指针	要求	可判定	是
C0405	Rule 21.16	标准库函数memcmp的指针实参必须指向指针类型，或者指向基本有符号类型，基本布尔类型或基本枚举类型	要求	可判定	是
C0404	Rule 21.17	使用<string.h>提供的字符串处理函数（string handling functions）产生的访问不得超越指针形参引用的对象的边界	强制	不可判定	是
C0403	Rule 21.18	size_t实参若传递给任意<string.h>提供的函数，则必须有恰当的值	强制	不可判定	是
C0402	Rule 21.19	标准库函数localeconv，getenv，setlocale或strerror返回的指针只得作为const修饰类型的指针使用	强制	不可判定	是
C0401	Rule 21.20	标准库函数 asctime / ctime / gmtime / localtime / localeconv / getenv / setlocale / strerror 返回的指针后面不得紧跟对同一函数的调用	强制	不可判定	是
C0421	Rule 21.21	不得使用<stdlib.h>中的标准库函数system	要求	可判定	是
C0210	Rule 22.1	通过标准库函数动态获取的所有资源都必须被显式释放	要求	不可判定	是
C0209	Rule 22.2	只有通过标准库函数分配的内存块才能被释放	强制	不可判定	是
C0208	Rule 22.3	不得在不同文件流上同时打开同一文件进行读写访问	要求	不可判定	是
C0207	Rule 22.4	不得对只读文件流进行写入操作	强制	不可判定	是
C0206	Rule 22.5	不得解引用（dereference）指向FILE对象的指针	强制	不可判定	是
C0205	Rule 22.6	不得在相关文件流关闭后使用FILE指针的值	强制	不可判定	是
C0204	Rule 22.7	宏EOF只得与任何能够返回EOF的标准库函数的未修改返回值比较	要求	不可判定	是
C0203	Rule 22.8	在调用errno设置函数之前必须将errno值设置为零	要求	不可判定	是
C0202	Rule 22.9	调用errno设置函数后必须检测errno值是否为零	要求	不可判定	是
C0201	Rule 22.10	只有上一个被调用的函数是errno设置函数的情况下才能检测errno值	要求	不可判定	是