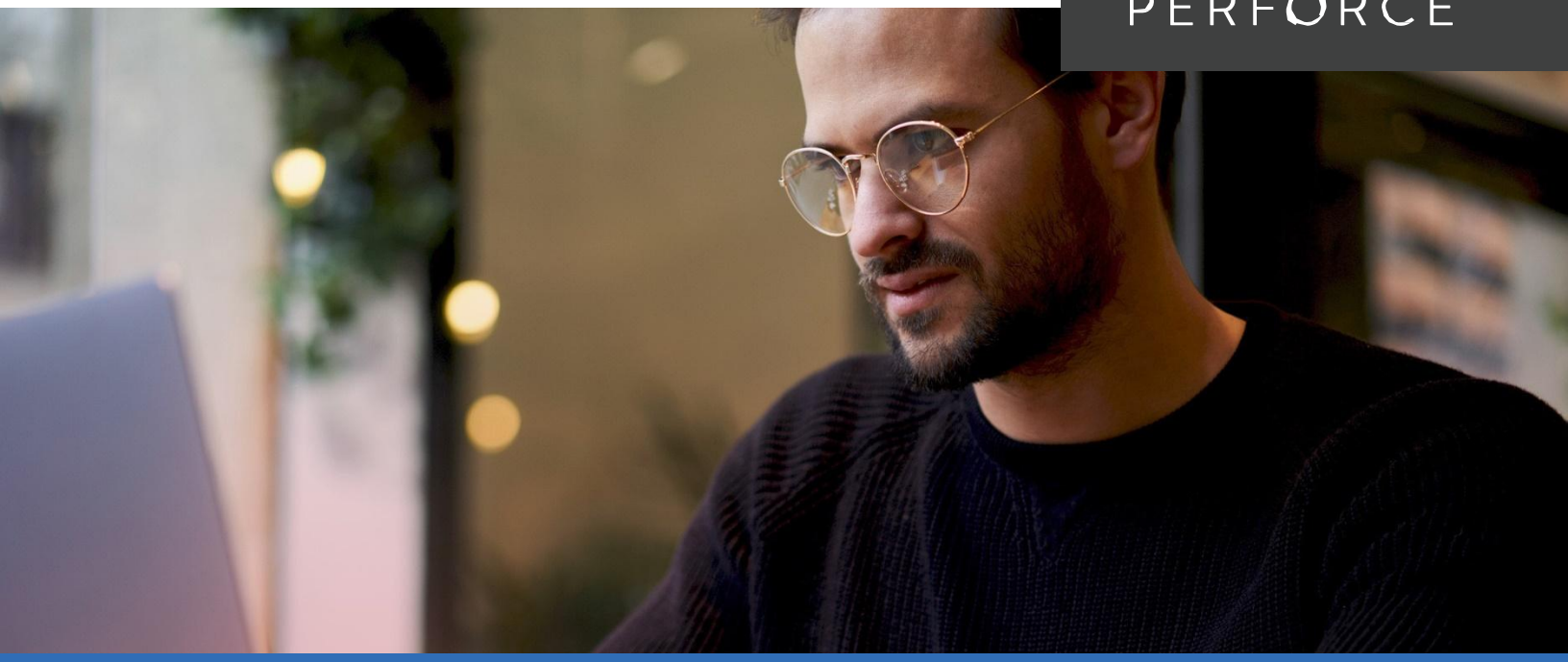
TRINITY

PERFORCE



白皮书

向现代 C++转变：

通过 AUTOSAR 编码指南确保软件安全

引言

许多高安全系统开发工程师结合既定的编码标准，其中包括 MISRA C++、Joint Strike Fighter Air Vehicle C++（简称 JSF AV C++），使用 C++03 语言。如今，他们想利用现代 C++(C++11/14)中可用的语言特征的优势，因此需要选择适当的编码标准来保证安全性。本文探讨了 AUTOSAR C++14 新编码指南和 MISRA C++之间的关系，还解释了新语言特征的使用存在高度的共通性，只有细微的差异。本文得出结论：如果采用一种适当的自动化静态分析解决方案，从 MISRA C++ 向 AUTOSAR 的转变并没有那么困难。

C++在逐渐演变.....

C++编程语言在持续演变。这意味着，能够帮助开发工程师编写安全可靠的代码的编码标准，还需要演变以反映更改的以及新的语言特征使用的最佳实践。2017 年，AUTOSAR 发布其新的 C++14 编码指南。此新编码标准是 MISRA C++:2008 的延伸，为使用现代 C++语言编程的高安全和高可靠性嵌入式系统提供了最佳实践指导。

本文探讨了 AUTOSAR 编码指南、MISRA C++以及其他既定的编码标准之间的关系。

AUTOSAR 编码指南与既定的编码标准有何不同？

AUTOSAR 指南得以引进，原因在于 AUTOSAR 无法为关键的安全相关软件中现代 C++ (C++11/14) 的使用找到适当的编码标准，例如那些开发应用于高度连通的自动驾驶汽车的自适应平台（Adaptive Platform）软件架构所需的编码标准。汽车和其它行业中最常用的标准是 MISRA C++。MISRAC++:2008 是专门为 C++03 的使用和强制要求而设计的规范。

AUTOSAR 指南借鉴了大量既定的编码标准。2018 年 3 月发布的版本明确了 402 条规则。其中 138 条直接引用了 MISRA C++:2008。AUTOSAR 规则还参考了其他一些编码标准和资源，包括：

- Joint Strike Fighter Air Vehicle C++编码标准（JSF AV C++）
- High Integrity C++编码标准 4.0 版（HICPP）
- CERT C++编码标准（CERT）
- C++ Core 指南（C++ CORE）
- Google C++ 风格指南

AUTOSAR 指南包含一个章节“对现有标准的可追溯性”。该部分详述了 MISRA、HICPP、JSF、C++ Core 指南以及 CERT 编码标准中每一条规则如何与 AUTOSAR 规则构成相关性，并将规则划分为“完全一样”、“轻微差异”、“明显差异”和“被移除”四类。

下表对这一信息进行了总结（从 2018 年 3 月发布的版本起）。

该表显示，91%的 MISRAC++:2008 规则中至少有一部分已融入 AUTOSAR 指南。

编码标准	规则数目	与 AUTOSAR 编码指南存在共同之处的规则数目			共有的规则所占比例
		完全一样	轻微差异	明显差异	
MISRAC++	229	138	38	32	91%
HICPP	155	0	99	25	80%
JSF	226	0	143	28	76%
C++CORE	412	0	174	49	54%
CERT	156	0	75	33	69%

事实上，如指南第 1 章节所述，“本指南被定义为 MISRAC++:2008 的升级更新版本……对本指南的读者而言，MISRAC++:2008 是必要的先决条件”。

我的团队应当运用 AUTOSAR 编码指南吗？

通常，开发工程师会想要使用更新的语言功能，但管理层却要求他们使用一个完善的、经过验证的编码标准。许多情况下，原因在于：系统需要获取功能安全认证。如果您的项目将使用 C++14 进行编码，且您需要证明对诸如 ISO 26262 等功能安全标准的合规性，那么除了 AUTOSAR，您可以选择的切实可行的编码标准却很有限。C++ Core 指南和 CERT C++支持 C++14。但是，在 AUTOSAR 之前，没有任何其它为支持 C++14 的安全或者任务关键型软件专门设计的标准。

编码标准所支持的语言版本：
MISRAC++:2008 明确规定了是对 C++03 的要求（规则 1-0-1：“所有代码应该符合 ISO/IEC 14882:2003”）。MISRAC++:2008 中指出“对某些语言扩展的使用必须要声明偏离”，但不允许使用后来的语言规范。

编码标准	支持的语言版本
AUTOSAR	C++14
MISRAC++	C++03
HICPP	C++11
JSF	C++03
C++ CORE	C++17/C++14/C++11
CERT C++	C++14

这意味着：严格来说，任何使用 C++14 编译器编写的代码都无法满足 MISRA 合规性。

如果您的确在 C++14 代码上运行了您现有的 MISRA 检测工具，就会出现违反情况。除了违反规则 1-0-1 外，您要针对每一种违反情况给出正当的理由，从而声明您的代码是符合 MISRA 编码标准的。在您给出的正当理由中，您需要表明：您已经使用其他一些方法对代码进行了检测。换句话说，可以声明 C++14 代码对 MISRA 的合规性，但是很显然还需要付出更多努力。

AUTOSAR 编码指南缓解了这种困境。以下是 AUTOSAR 允许使用的几个 C++14 功能：

可变参数模板

可变参数模板使您可以编写这样的函数：以类型安全的方式使用任意数量的参数并且在编译时（而不是运行时）就解析所有参数处理逻辑。

二进制字面值

二进制字面值使阅读和理解正在执行逐位运算的代码变得更容易。在 C++14 之前，并不支持诸如 0b11111100 之类的二进制字面值，而且您必须使用八进制(0374)或者十六进制(0xFC)符号表示字面值。

数字序列分隔符

在编写大的数字的时候使用分隔符可以改善可读性。C++14 允许使用单引号(‘)作为数字分隔符。

例如：

```
//C++14.All of the following variable sequal 1048576
long decval=1’048’576;//groups of three digits
long hexval=0x10’0000;//four digits
long octval=00’04’00’00’00;//two digits
long binval=0b100’000000’000000’000000;
//six digits
```

我们应当如何从现有的编码标准进行过渡？

如果您已经运用了 MISRA 编码标准，那么您已经掌握了大约 91 %的内容。然而，一个 AUTOSAR 检测工具将会在符合 MISRA 规范的现有代码中找到违反情况。这是因为，C++11/14 中引入的一些功能特征必须用在符合 AUTOSAR 规范的代码中。

例如，规则 A8-5-2：“使用大括号进行初始化{}，而不使用等号，适用于变量初始化”。

在 C++03 中，初始化往往通过以下表达式进行：

```
int32x1 = 0;
```

这个表达式不再合规了。相反地，您应该使用以下表达式：

```
std::int32_t x1{0};
```

这就避免了产生歧义以及安全隐患，因为可以防止收缩转换。例如：

```
std::int8_tc1 = 400;// ok?
std::int8_tc2{400};// error: narrowing conversion
```

这个初始化表达式之前只引入到 C++11 中。这是在 C++03 后引入的功能特征相关的好例子，C++03 改进了语言，应该被视为‘最佳实践’，但 MISRA C++并不支持。

另一个例子是 A3-9-1：“<cstdint>中表示大小和符号的固定宽度的整数类型，应当代替基本数值类型得以使用”。这取代了 MISRA C++规则 3-9-2，说明 typedef’d 类型应当代替基本数值类型得以使用。MISRA C++依赖用户或者实现定义的类型，而 AUTOSAR 却使用的是 ISO C++标准中定义的类型。

您将需要通过修改代码来处理上述等问题，或者提出偏离。

总结

现代 C++开创了一些有用的语言特征。当与 AUTOSAR 编码指南结合起来使用时，您可以提高您的代码的可读性和您的嵌入式系统的安全性。您将需要预留一些时间和精力，使您现有的代码符合该指南的要求，时间的长短以及精力的多少取决于您现有的代码是如何编写的。如果您的代码目前符合的是 MISRA 规范，您可能需要根据我们列举的一些例子，做出相应的改变。

您将需要更改或者升级您的 MISRA 检测工具，从而执行 AUTOSAR 检测。Perforce（之前称作 PRQA）是唯一的作为 AUTOSAR 开发合作伙伴的静态分析工具供应商。我们帮助制定 C++14 编码指南，而且已经针对 AUTOSAR 自适应平台的演示代码对我们的 AUTOSAR 解决方案进行过测试。如果您已经在使用适用于 C++ 的 Helix QAC（之前称作 QAC++），可以轻松地添加我们的 AUTOSAR 合规模块，并分析您的代码是否符合该指南的要求。

了解更多

访问网站 perforce.com，了解 Helix QAC 如何使现代 C++运用如此简单。

观看演示>
https://www.perforce.com/products/helix-qac/live-demo

创提信息科技（上海）有限公司
http://www.trinitytec.com.cn

创提信息科技（上海）有限公司 – Trinity Technologies (Shanghai) Limited

专注于嵌入式软件研发质量和自动化测试的方案和咨询服务，提供覆盖软件测试整个流程的完整的解决方案，包括从研发前期的代码级测试到后期的系统级测试，从静态分析到动态测试，从编码检查，单元测试、集成测试到性能测试和测试覆盖率分析等。

公司通过专业的自动化工具（如 DT10、VectorCAST、PRQA、SQUORE 等）和服务满足不同客户对软件质量和测试的需求，持续协助客户改进软件研发质量和效率。客户主要集中在高安全和高可靠性领域，如国防和航空航天、轨道交通、汽车电子、医疗器械、工业控制、通讯和电力电子等行业。公司提供的领先的解决方案不仅为数以百计的客户提高产品质量，还协助客户遵循高安全和高可靠性行业的合规性要求，如 DO-178B/C、IEC61508、EN50128、ISO26262、IEC62304 和 MISRA 等行业标准，并获得相关机构认可和认证。