



动态代码分析工具 C-RUN

通过动态代码分析，确保代码质量

通过动态代码分析，您可以在运行时发现实际错误。IAR 动态代码分析工具 C-RUN 很容易根据客户要求使用，并具有广泛的功能。C-RUN 通过直接在开发环境中检查应用程序执行来进行动态代码分析。它检查算术问题、边界问题和堆完整性，并告诉您出了什么问题以及在哪里出了问题。

关键功能亮点：

- 分析 C 和 C++ 代码
- 直观易用的设置
- 独特的测试代码优化，最小化代码大小开销
- 全面且详细的运行时错误信息
- 每个发现的错误提供调用堆栈信息
- 编辑器中的代码相关性和图形反馈
- 灵活的错误过滤管理
- 边界检查，确保对数组和其他对象的访问在边界内
- 缓冲区溢出检测
- 在类型转换时检测值的更改
- 检查计算中的溢出和环绕
- 发现位移操作中的位丢失
- 堆和内存泄漏检查
- 支持的产品版本：
 - IAR Embedded Workbench for Arm 7.20 版本及以上
 - IAR Embedded Workbench for RX 3.10 版本及以上

算术问题、边界问题和堆检查

算术问题包括溢出、环绕、转换错误、除零和 switch 语句中缺少 default 标签。可以通过在潜在错误可能发生的所有地方插入特定的测试代码来检测此类错误。源代码级别的测试通常会插入一个 if 语句或等效语句，检查条件，并将一些内容打印到 stdout 或将特殊值写入端口以记录问题。类似地，编译器可以插入指令来检查条件，并在运行时报告问题。代码大小将随要检查的操作数量更多或更少地呈线性增加。

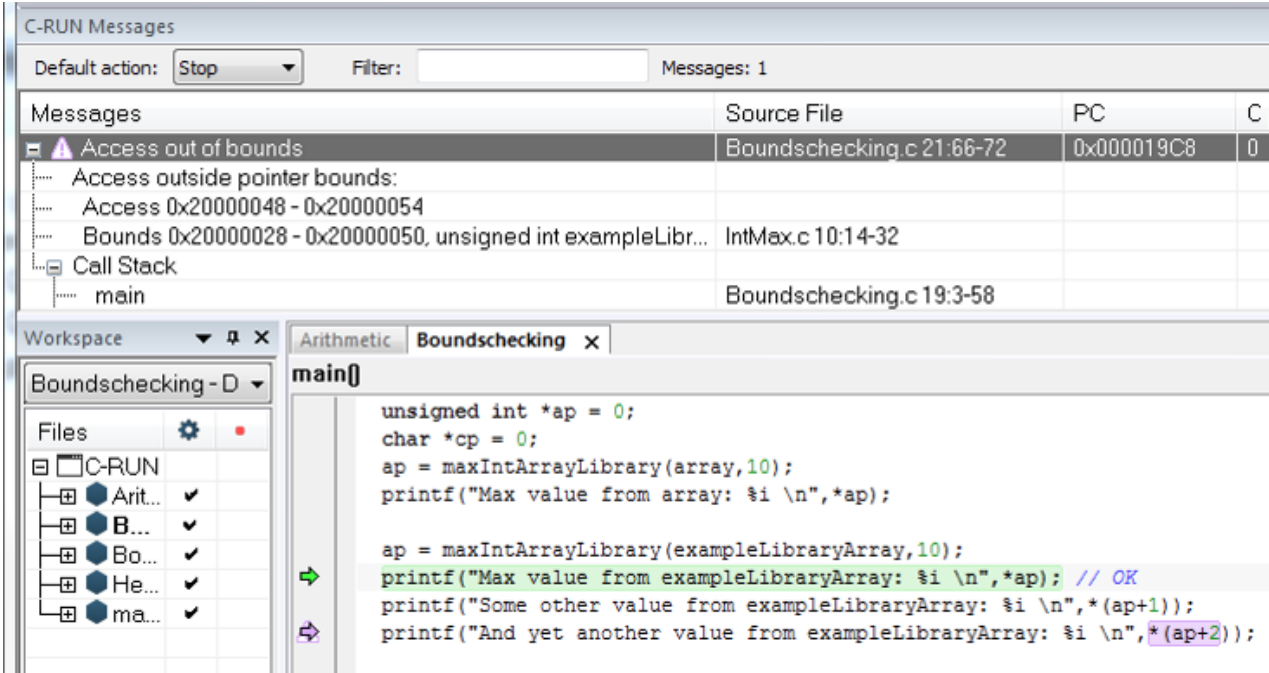
边界问题是一个非常广泛的问题类别，包括典型的越界问题，如在定义的数组边界外写入或读取。但是，越界概念可以推广到通过指针访问的任何内容，无论类型或大小如何。这包括指向堆栈上的标量对象的指针，因此如果您更改指针，或者有恶意意图的人为您更改指针，指向堆栈上的某个内容，那么先进的边界检查器可以检测到指针的新值是否在有效对象的边界内。

堆检查确保堆保持其完整性，并且随着时间的推移不会泄漏已分配的块。高效的堆检查本质上是库实现的一种练习，但是如果某些功能可以像其他编译器内部函数一样处理，则了解相关编译器的内部情况可能会有所裨益。完整性检查通常在每次调用 malloc、free 等函数时进行，无论是在 C 还是 C++ 环境中。如果堆很大，堆完整性检查可能会对性能产生真正的负面影响，因为检查可能涉及遍历整个堆，因此决定检查频率的方式对于某些应用程序可能至关重要。



使用 C-RUN 进行简便的动态代码分析和错误检查

C-RUN 是 IAR Embedded Workbench 的扩展，是您开发工作流程的自然组成部分，在传统的编辑/构建/调试周期、运行单元测试或进行集成测试时。C-RUN 在代码第一次进行测试时就为您提供极其有价值的反馈。得益于与 IAR Embedded Workbench 的紧密集成，C-RUN 可以成为任何开发人员日常工作的一部分。



The screenshot displays the C-RUN Messages window and the source code editor. The Messages window shows an error: "Access out of bounds" at line 21:66-72 of Boundschecking.c, with PC 0x000019C8 and C 0. The error details indicate an access outside pointer bounds: "Access 0x20000048 - 0x20000054" and "Bounds 0x20000028 - 0x20000050, unsigned int exampleLibr...". The Call Stack shows the error occurred in main at Boundschecking.c 19:3-58.

```
main()
{
    unsigned int *ap = 0;
    char *cp = 0;
    ap = maxIntArrayLibrary(array, 10);
    printf("Max value from array: %i \n", *ap);

    ap = maxIntArrayLibrary(exampleLibraryArray, 10);
    printf("Max value from exampleLibraryArray: %i \n", *ap); // OK
    printf("Some other value from exampleLibraryArray: %i \n", *(ap+1));
    printf("And yet another value from exampleLibraryArray: %i \n", *(ap+2));
}
```

IAR Embedded Workbench

IAR Embedded Workbench 是一个完整的 C/C++ 嵌入式应用开发工具链。该工具链提供领先的代码质量、卓越的尺寸和速度优化，以及强大的广泛的调试功能，配备完全集成的调试器，支持模拟和硬件调试。C-RUN 完全集成在 IAR Embedded Workbench IDE 中，帮助开发人员在早期阶段确保代码的安全性和高质量，从而帮助企业缩短产品上市时间，因为后期发现和修复错误会非常耗时且昂贵。

联系我们

爱亚系统开发（上海）有限公司

上海办公室

Tel: 021- 6375 8658

Email: marketing.cn@iar.com

深圳办公室

Tel: 0755 - 2641 3374

Email: marketing.cn@iar.com

