



## 7 Steps to reduce GNU GCC code-size

- ✓ **Enable size optimization in release mode.** Use the compiler option `-Os` to reduce the code-size in release mode.
- ✓ **Enable size optimization in debug mode.** Use the compiler option `-Og` to reduce the code-size for debuggable code.
- ✓ **Enable dead-code removal.** Use the compiler option `-ffunction-sections` and the linker option `--gc-sections` to make the linker remove unused functions.
- ✓ **Enable dead-data removal.** Use the compiler option `-fdata-sections` and the linker option `--gc-sections` to make the linker remove unused variables.
- ✓ **For C++: Disable RTTI and exception handling.** Use the compiler options `-fno-rtti` and `-fno-exceptions`.
- ✓ **Select a compact runtime library.** Link with Newlib Nano instead of the standard Newlib runtime library. Additionally, use the Atollic “tiny printf” implementation for further code-size reductions if you use the [Atollic TrueSTUDIO C/C++ IDE](#).
- ✓ **Try link time optimization (LTO).** LTO enables the linker to perform cross-file optimizations and code resequencing. Use the command line option `-flto` on both the compiler and linker to enable LTO, and add the command line options normally used with the compiler also to the linker.

**Click to read our free white paper on professional ECLIPSE/GNU tools for ARM Cortex development!**