

Configuring the Compiler

In this section:

- [Overview](#)
- [Specifying Multiple Compilers](#)
- [Working with Custom Compiler Configurations](#)

Overview

You must configure C/C++test for specific C and C++ compilers, including compiler versions, before running the analysis. See [Running Static Analysis](#) for basic information on configuring compilers. For a full list of supported compilers, see [Supported Compilers](#).

Specifying Multiple Compilers

If multiple compilers need to be used during a single test run, they can be configured with the following extended syntax of the `-compiler / cpptest.compiler.family` option where `<COMPILER_COMMAND>` is the path to the original compiler executable and `*` is match-all character:

```
-compiler <COMPILER_ID>:<COMPILER_COMMAND>;<COMPILER_ID>:<COMPILER_COMMAND>;...
cpptest.compiler.family=<COMPILER_ID>:<COMPILER_COMMAND>;<COMPILER_ID>:<COMPILER_COMMAND>;...
```

Example

```
-compiler vc_9_0:*cl.exe;gcc_3_4:/usr/bin/gcc.exe
cpptest.compiler.family=vc_9_0:*cl.exe;gcc_3_4:/usr/bin/gcc.exe
```

Working with Custom Compiler Configurations

When working with custom compiler configurations, the configuration files should be copied into a directory defined with the `cpptest.compiler.dir.user` configuration entry:

```
cpptest.compiler.dir.user=/home/custom_compilers
```

Each custom compiler should be located in a dedicated subdirectory with unique name:

```
/home/custom_compilers (cpptest.compiler.dir.user configuration entry)
custom_gcc
  c.psrc
  cpp.psrc
  gui.properties
custom_iar
  c.psrc
  cpp.psrc
  gui.properties
```