



Documentation:

Ag++ Manual:

Georg Blaschke

Version 0.1

Contents

1 Introduction	4
2 Invocation	4
2.1 Options	4
2.1.1 --gen_config	4
2.1.2 --weave_only	4
2.1.3 -p --path	6
2.1.4 -c	6
2.1.5 -v --verbose [<arg>]	6
2.1.6 --keep_acc	6
2.1.7 --c_compiler	6
2.1.8 --ac_compiler	6
2.1.9 --config <arg>	6
2.2 Examples	6

1 Introduction

The `ag++` program provides a more intuitive frontend to the AspectC++ weaver (`ac++`) in a GNU environment. The only preliminaries are a working installation of GNU C++ compiler, which also can run within a cygwin environment. It basically wraps the functionality of the aspect weaver and the c++ compiler into one single program.

2 Invocation

The usage of `ag++` is mainly influenced by the usage of the GNU `g++` compiler. Let's say, you want to **compile** a single file (here: `main.cc`) with `g++`, you have to run

```
g++ -c main.cc
```

in order to generate an object file.

To **weave** and **compile** a single file you invoke

```
ag++ -c main.cc
```

the same way like did before with `g++`.

2.1 Options

All available options are summed up in the options table¹. Options which are not listed here are accounted as `g++` options. The column labeled with *AC++* shows if the option is taken over from `ac++` by `ag++` ('X'), not supported by `ac++` ('-') or modified('!'). All options which are taken over, are not described in this document. Consult the AC++ Compiler Manual instead.

2.1.1 `--gen_config`

Just create a parser configuration and quit afterwards. The argument of the `-o` option specifies the name of the file. In any other case (no `--gen_config` and/or no `-o` option) a configuration file with the name 'puma.config' will be generated in the directory where `ag++` was invoked.invoked

2.1.2 `--weave_only`

Generate only woven source code files. With `-o` option and one file the generated output is named after the argument of the `-o` option.

Option	AC++	Description
--gen_config	-	Only generate Puma configuration file
--weave_only	-	Weave only
-c	!	Compile only
--keep_acc	-	Keep woven source code files
--c_compiler <arg>	-	Path to C++ compiler
--ac_compiler <arg>	-	Path to AspectC++ compiler
-p --path <arg>	!	Defines a project directory
-d --dest <arg>	X	Specifies a target directory for saving
-v --verbose <arg>	!	Level of verbosity (0-9)
-o --output <arg>	X	Name of the output file
-i --include_files	X	Generate manipulated header files
-a --aspect_header <arg>	X	Name of aspect header file or 0
-r --repository <arg>	X	Name of the project repository
--config <arg>	!	Parser configuration file
--no_line	X	Disable generation of #line directives
--problem...	X	enable back-end compiler problem workaround
--no_problem...	X	disable back-end compiler problem workaround
-I <arg>	X	Include file search path
-D <name>[=<value>]	X	Macro definitions
-U <name>	X	Undefine a macro
-include <arg>	X	Forced include

Table 1: ac++ Compiler Option Summary

2.1.3 `-p|--path`

This options differs only slightly from the `-p` option of `ac++`. In `ag++` it is mandatory to specify a project path, whereby `ag++` the current working directory is used as project path by default. See the AspectC++ Compiler Manual for a more detailed description of this option.

2.1.4 `-c`

Like the `-c` option of `g++`, this options effects the creation of object files of one or more source files.

2.1.5 `-v|--verbose [<arg>]`

Set the level of verbosity.

2.1.6 `--keep_acc`

Don't remove intermediate `.acc` files generated by `ac++`.

2.1.7 `--c_compiler`

Specify path to GNU C++ compiler' The default is `g++`.

2.1.8 `--ac_compiler`

Specify path to AspectC++ compiler' By default `ag++` assumes, that the `ac++` executable is located in the same directory like itself.

2.1.9 `--config <arg>`

Path to a puma configuration file. If this option is available the configuration file will not be generated automatically.

2.2 Examples

- `ag++ --help`
Displays all options with a short description.
- `ag++ -o test Test.cc main.cc`
Weave, compile and link the source files `Test.cc` and `main.cc`. The created executable will be named `'test'`.

- `ag++ --gen_config`
Create a puma configuration file named `puma.config` within the current working directory.
- `ag++ --gen_config -o my.config`
Create a puma configuration file named `my.config`.
- `ag++ -p src -i -d gen/includes`
Generate modified include files out of all include files found below `src` directory and store them under `'gen/includes'`.