

- Non-Build-Commands:
 - `cry_waf.exe --help`
 - `cry_waf.exe configure`
 - `cry_waf.exe msvs`
 - `cry_waf.exe generate_uber_files`
- Build-Commands:
 - `cry_waf.exe build_<platform>_<compiler>_<configuration>`
 - `cry_waf.exe clean_<platform>_<compiler>_<configuration>`

This section covers all command supported by WAF which are not related to building the CryEngine.

`cry_waf.exe --help`

Shows the help screen, an overview about all commands and options supported by WAF.

`cry_waf.exe configure`

Performs a configure step. This operations checks some compiler settings and caches those.

Very much like the `./configure` operation used for linux projects.

`cry_waf.exe msvs`

Generates a Visual Studio Solution and Projects which can be used to build the CryEngine with WAF.

`cry_waf.exe generate_uber_files`

Generates Uber files which are used to speed up compilation by merging multiple translation units into a single large translation units.

This section covers all commands relevant for building projects with WAF. It features also an overview about all build options, since those are mostly relevant for

building projects.

`cry_waf.exe build_<platform>_<compiler>_<configuration>`

This command performs a build operation; all dependencies are checked, and all needed compile tasks are executed to generate the final executables and libraries.

- **<platform>**: refers to one of the supported target platforms.
- **<compiler>**: refers to one of the supported target compilers for the selected platform.
- **<configuration>**: refers to one of the supported build configurations.

`cry_waf.exe clean_<platform>_<compiler>_<configuration>`

This command performs a clean operation; all intermediate files previously generated by another build command. Follows the same pattern for `<platform>` and `<configuration>` as the build command.