

NAME

`gyp` – cross–platform makefile generator for Chromium

SYNOPSIS

`gyp` [options] [build_file...]

DESCRIPTION

Uses the specified build files to generate build scripts or project files in the current platform’s native format. If no build file is specified, all files in the current directory with a `.gyp` suffix are used.

OPTIONS

`-D<variable>=<value>`

Provide a default value for the given variable. This supplements settings from the `GYP_DEFINES` environment variable, which should be a space–separated list of `<variable>=<value>` pairs.

`-f<format>[(, <format>)...], --format=<format>[(, <format>)...]`

Generate output in the formats listed. Valid formats include `make` (GNU makefile), `xcode` (Xcode project file), `msvs` (Visual Studio project file), `scons` (SCons rule set), and `gypd` (`gyp` input, for debugging).

Defaults to the value of the `GYP_GENERATORS` environment variable, which should be a comma–separated list of formats, or the platform’s native format if unset (`xcode` for Mac OS X, `msvs` for Windows, including Cygwin, `make` for other Unix).

`-I<path>, --include=<path>`

Implicitly include the contents of the `<path>` file in all loaded `.gyp` files.

`--depth=<path>`

For Chromium, the `gyp` "depth" variable should always be a relative path to Chromium’s top–level `src` directory. This option allows one to specify a path the `DEPTH` variable should be set relative to.

Defaults to the first directory named `src` found as an ancestor of one of the build files passed on the command line.

`-d<debug mode>, --debug=<debug mode>`

Turn on debugging output. Valid debugging modes include "general" to list options used, "variables" to trace variable expansion, and "includes" to trace build file include directives.

`-S<suffix>, --suffix=<suffix>`

Insert the specified string in the names of generated files before the standard extension. For example, with the `make` backend and a suffix of ".foo", `gyp` will write the main makefile to `Makefile.foo` and target–specific snippets to `target.foo.mk`.

`-G<flag>=<value>`

Set a "generator flag", which is an option specific to the particular output format. All generator flags have names beginning with the name of the output format. This supplements the `GYP_GENERATOR_FLAGS` environment variable, which should be a space–separated list of `<flag>=<value>` pairs.

`--generator-output=<dir>`

Write the generated build files under the `<dir>` directory and adjust paths accordingly. This allows building in a separate directory from the source files.

This overrides the `GYP_GENERATOR_OUTPUT` environment variable.

`--ignore-environment`

Do not read options from environment variables.

`--check`

Take extra time to verify the syntax of each `build_file`.

`--parallel`

Use multiprocessing for speed (experimental).

`--toplevel-dir=<dir>`

Directory to use as the root of the source tree.

`--build=<configs>`

Configuration for build after project generation.

`--no-circular-check`

Disable the check for circular relationships between .gyp files. These relationships should not exist, but they've only been observed to be harmful with the Xcode output format. Chromium's .gyp files currently have some circular relationships on non-Mac platforms, so this option allows the strict behavior to be used on Macs and the lenient behavior to be used elsewhere.

This option is likely to be removed once all circular relationships in Chromium are eliminated.

ENVIRONMENT

GYP_DEFINES, GYP_GENERATORS, GYP_GENERATOR_FLAGS, GYP_GENERATOR_OUTPUT

See the `-D`, `--format`, `-G`, and `--generator-output` options.

FILES

`~/gyp/include.gypi`

Implicitly included before anything else in every .gyp file.

SEE ALSO

qmake(1), **cmake(1)**