FASTBUILD

Command Line Options

FBuild.exe

Option	Summary
-cache[read|write]	Use the build cache.
-clean	Force a clean build.
-config [path]	Explicity specify the config file to use.
-dist	Enable distributed compilation.
-distverbose	Enable detailed logging for distributed compilation.
-fastcancel	[Experimental] Active fast cancellation on build failure.
-fixuperrorpaths	Reformat GCC/SNC/Clang error messages in Visual Studio format.
-forceremote	Force distributable jobs to only be built remotely.

-help	Show usage help.
-ide	Enable multiple options for IDE integration.
-j[x]	Explicitly set local worker thread count.
-noprogress	Don't show the progress bar while building.
-nostoponerror	Don't stop building on first error.
-nosummaryonerror	Don't print the -summary output if there is an error.
-quiet	Don't show build output.
-report	Output a report at build termination.
-showcmds	Show command lines used to launch external processes.
-showdeps	Show known dependency tree for specified targets.
-showtargets	Show primary targets defined in build configuration.
-summary	Show a summary at the end of the build.
-verbose	Show detailed diagnostic information for debugging.
-version	Print version and exit.
-vs	[Deprecated] Same as -ide.
-wait	Wait for a previous build to complete before starting.
-wrapper	Wrapper mode for Visual Studio. (Windows only)

FBuildWorker.exe

Option	Summary
-console	Disable UI. (Windows Only)
-cpus=[n|-n|n%]	Control worker CPUs allocation.
-mode=[disabled|idle|dedicated]	Control worker availability.
-nosubprocess	Don't spawn as a sub-process.

FBuild.exe Detailed

-cache[read|write]

Enable usage of the build cache. The cache options need to be configured in the build configuration file.

The cache can be enabled as read only or write only with '-cacheread' or '-cachewrite'. This can be useful for automated build systems, where you might like one machine to populate the cache for read-only use by other users.

Use of '-cache' is equivalent to '-cachread' and '-cachewrite' together.

-clean

Force a clean build. The build configuration file is re-parsed and all exsiting dependency information is discarded. A build is performed as if building for the first time with no built files present.

-config [path]

Explicity specify the config file to use. By default, FASTBuild looks for "fbuild.bff" in the current directory. This options allows a file to be explicitly specified instead.

-dist

Enable distributed compilation. Requires some build configuration.

-distverbose

Print detailed information about distributed compilation. This can help when investigating connectivity issues. Activates -dist if not already specified.

[Experimental] Active fast cancellation on build failure.

-fastcancel

Normally, when a build fails, FASTBuild will stop spawning new tasks, but will allow any other concurrently running processes to complete. In some cases, these tasks can take a long time to complete. When activated, the -fastcancel option will allow FASTBuild to instead terminate those concurrent jobs.

-fixuperrorpaths

Enables re-formatting of warnings, errors and notes for GCC/SNC & Clang to Visual Studio format. Additionally, relative paths are expanded to full paths. This allows these errors to be double-clickable inside Visual Studio.

Enabled automatically when '-vs' is used.

-forceremote

Prevents all local compilation of distributable jobs.

FASTBuild will normally utilize local CPU resources to compile distributable jobs in several situations, in order to improve performance:

- When there are no remote workers available (otherwise build would not complete)
- When all remote workers are busy and local CPUs would be idle
- When blocked on remote work

This option prevents local consumption of distributable jobs in all these cases. This will generally result in slower builds and may even prevent the build completing entirely. As such, this options should generally only be used for troubleshooting.

Additionally, this option disabled use of the cache.

NOTE: This option can prevent builds from completing (if no workers are available for example).

NOTE: This option will generally degrade build performance.

-help

Prints command line usage information, as per the summary at the top of this page.

-ide

IDE integration mode. Enables several options that are commonly desired when running from within an IDE such as VisualStudio, XCode or kDevelop. The following options are enabled:

- -noprogress
- -fixuperrorpaths (Windows only)
- -wrapper (Windows only)

-j[x]

Generally, the default behaviour will give best performance, and this option should only be used in very specific situations.

The -j[x] option allows you to artificially control local parallelism by modifying the local thread pool size.

FASTBuild will normally determine the optimal number of local threads to use by detecting the number of hardware cores present on the host. The -j option allows you to override this.

Positive values for x can be used to set the number of tasks which can be performed locally in parallel. This can be used to limit CPU usage on a machine that needs to perform other work while compilation is in progress. Values greater than the number of physical processors are also accepted, but will almost always result in degraded performance.

A value of 0 for x indicates that no additional threads should be spawned, and build graph processing and compilation should occur on the same thread. This can be

useful for build process debugging, especially when combined with the '-verbose' option.

This option has no direct bearing on distributed compilation, but modifying local parallelism will reduce the ability of FASTBuild to distribute work efficiently.

-noprogress

Suppresses the progress bar that is normally shown while compiling.

This should be used when targetting compilation from within Visual Studio or another IDE. (or use -vs)

-nostoponerror

When encountering build errors, FASTBuild will normally stop as quickly as possible.

-nostoponerror instructs FASTBuild to instead build as much as possible before stopping when failures occur. This is useful if you want to see as many errors as possible in your compilation output.

NOTE: When specifying multiple targets to compile on the command line, -nostoponerror is implied.

-nosummaryonerror

The -nosummaryonerror option instructs FASTBuild to only print the -summary overview if the build completes successfully.

NOTE: When specifying -nosummaryonerror, -summary is implied if not already specified.

-quiet

Don't show build output. Information about which items are being built and the overall state of the build will be suppressed.

-report

Ouput a detailed report at the end of the build. The report is written to report.html in the current directory.

The build report contains details of:

- The build environment (version, cmd line used etc.)
- All items built.
- Cache utilization.

- Include file usage.

NOTE: This option will lengthen the total build time, depending on the complexity of the build.

-showcmds

Displays the full command lines passed to external tools as they are invoked.

This option is useful for debugging build configurations, where the -verbose mode is too spammy.

NOTE: This option may have an impact on build performance.

-showdeps

Displays the hierearchy of dependencies for the specified target(s). This can be useful for debugging build configurations.

Example:

```
fbuild.exe -showdeps Game-x86-Debug
```

If no target is specified, "all" is used.

NOTE: The dependencies shown will refelect the state as of the last completed build. i.e. dependencies that would be discovered during the next build will not be shown.

-showtargets

Displays the list of targets defined in the bff configuration file.

-summary

Displays a summary upon build completion.

-verbose

Show detailed diagnostic information for debugging.

This can be used to provide more information when debugging a build configuration problem. It will display detailed information as the build configuration is parsed, as well as detailed information during the build, including full command line arguments passed to external executables.

It is usually useful to combine this flag with -j0 to serialize the build process and output (avoiding the output of different threads being mixed together).

-version

Prints executable version information and exits. No configuration parsing or building is performed.

-vs

[Deprecated] VisualStudio mode - same is -ide. Use -ide instead.

-wait

Queue build after an already running build completes.

Only one instance of FASTBuild can run at a time within the same root working directory. If you launch another FASTBuild while one is already running, the error "Another FASTBuild is already running." will be emitted.

If you wish to build multiple targets, you should specify them together on the command line. This allows for paralellization across both targets.

Alternatively, the -wait command line arg allows you to queue the second build, so instead of failing, it will start after the first build completes. This will be slower than if

both targets were invoked together on the original command line.

-wrapper (Windows Only)

Spawns FASTBuild via an intermediate sub-process to be able to cleanly terminate a build from Visual Studio.

When cancelling a build in Visual Studio, the FASTBuild process will be killed, with no opportunity to save the build database, resulting in lost compilation work. Specifying this option spawns an orpaphaned child process to do the actual work. When the parent process is terminated by Visual Studio, the child process detects this and cleanly shuts down.

If a subsequent "wrapper mode" build is initiated before the first terminates, FASTBuild will wait for this first process to complete.

FBuildWorker.exe Detailed

-console

Disable worker UI.

The FBuildWorker can run in a UI-less mode on Windows. (On OSX and Linux, the worker currently always runs in UI-less mode)

-cpus=[n|-n|n%]

Control worker CPUs allocation.

The number of workers available is normally controlled through the UI of the FBuildWorker.exe. The "-cpus" command line option will override this as follows:

Syntax	Description
-cpus=n	Value n will be used.
-cpus=-n	Value of NUMBER_OF_PROCESSORS-n will be used.
-cpus=n%	Specify number of CPUs as a percentage of NUMBER_OF_PROCESSORS.

In all cases, the number used will be clamped between 1 and the NUMBER_OF_PROCESSORS environment variable.

NOTE: The newly overridden options will be saved and used on subsequent restarts of the worker.

-mode=[disabled|idle|dedicated]

Control worker availability.

The FBuildWorker.exe mode is normally controlled through the UI. The "-mode" command line option will override this as follows:

Syntax	Description
-mode=disabled	Worker will accept no tasks.
-mode=idle	Worker will accept tasks when PC is considerd idle.
-mode=dedicated	Worker will accept tasks regardless of PC state.

NOTE: The newly overridden options will be saved and used on subsequent restarts of the worker.

-nosubprocess

Don't spawn a sub-process copy of the worker.

By default, when the FBuildWorker is launched, it makes a copy of itself (FBuildWorker.exe.copy), launches the copy and terminates. The duplicate process monitors the original executable file for changes, and re-launches itself if the file is updated. In this way, the FBuildWorker.exe can be kept under revision control, and when synchronized to a new version, will automatically re-start.

The "-nosubprocess" option supresses this behaviour.