

watch()

Allows watching globs and running a task when a change occurs. Tasks are handled uniformly with the rest of the task system.

Usage

```
const { watch } = require('gulp');

watch(['input/*.js', 'input/something.js'], function(cb) {
  // body omitted
  cb();
});
```

Signature

```
watch(globs, [options], [task])
```

Parameters

parameter	type	note
globs	string	Globs to watch on the file system.
(required)	array	
options	object	Detailed in Options below.
task	function string	A task function or composed task - generated by series()and parallel().

Returns

An instance of chokidar for fine-grained control over your watch setup.

Errors

When a non-string or array with any non-strings is passed as globs, throws an error with the message, "Non-string provided as watch path".

When a string or array is passed as task, throws an error with the message, "watch task has to be a function (optionally generated by using gulp.parallel or gulp.series)".

Options

name	type	default	note
ignoreInitial	boolean	true	If false, the task is called during instantiation as file paths are discovered. Use to trigger the task during startup. Note: This option is passed to chokidarbut is defaulted to true instead of false.
delay	number	200	The millisecond delay between a file change and task execution. Allows for waiting on many changes before executing a task, e.g. find-and-replace on many files.
queue	boolean	true	When true and the task is already running, any file changes will queue a single task execution. Keeps long running tasks from overlapping.
events	string array	['add', 'change', 'unlink']	The events being watched to trigger task execution. Can be 'add', 'addDir', 'change', 'unlink', 'unlinkDir', 'ready', and/or 'error'. Additionally 'all' is available, which represents all events other than 'ready' and 'error'. <i>This option is passed directly to chokidar.</i>
persistent	boolean	true	If false, the watcher will not keep the Node process running. Disabling this option is not recommended. <i>This option is passed directly to chokidar.</i>

name	type	default	note
ignored	array string RegExp function		Defines globs to be ignored. If a function is provided, it will be called twice per path - once with just the path, then with the path and the fs.Stats object of that file. <i>This option is passed directly to chokidar.</i>
followSymlinks	boolean	true	When true, changes to both symbolic links and the linked files trigger events. If false, only changes to the symbolic links trigger events. <i>This option is passed directly to chokidar.</i>
cwd	string		The directory that will be combined with any relative path to form an absolute path. Is ignored for absolute paths. Use to avoid combining globs with path.join(). <i>This option is passed directly to chokidar.</i>
disableGlobbing	boolean	false	If true, all globs are treated as literal path names, even if they have special characters. <i>This option is passed directly to chokidar.</i>
usePolling	boolean	false	When false, the watcher will use fs.watch() (or fsevents on Mac) for watching. If true, use fs.watchFile() polling instead - needed for successfully watching files over a network or other non-standard situations. Overrides the useFsEvents default. <i>This option is passed directly to chokidar.</i>
interval	number	100	Combine with usePolling: true. Interval of file system polling. <i>This option is passed directly to chokidar.</i>
binaryInterval	number	300	Combine with usePolling: true. Interval of file system polling for binary files. <i>This option is passed directly to chokidar.</i>
useFsEvents	boolean	true	When true, uses fsevents for watching if available. If explicitly set to true, supersedes the usePolling option. If set to false, automatically sets usePolling to true. <i>This option is passed directly to chokidar.</i>
alwaysStat	boolean	false	If true, always calls fs.stat() on changed files - will slow down file watcher. The fs.Stat object is only available if you are using the chokidar instance directly. <i>This option is passed directly to chokidar.</i>
depth	number		Indicates how many nested levels of directories will be watched. <i>This option is passed directly to chokidar.</i>
awaitWriteFinish	boolean	false	Do not use this option, use delay instead. <i>This option is passed directly to chokidar.</i>
ignorePermissionErrors	boolean	false	Set to true to watch files that don't have read permissions. Then, if watching fails due to EPERM or EACCES errors, they will be skipped silently. <i>This option is passed directly to chokidar.</i>
atomic	number	100	Only active if useFsEvents and usePolling are false. Automatically filters out artifacts that occur from "atomic writes" by some editors. If a file is re-added within the specified milliseconds of being deleted, a change event - instead of unlink then add - will be emitted. <i>This option is passed directly to chokidar.</i>

Chokidar instance

The watch() method returns the underlying instance of chokidar, providing fine-grained control over your watch setup. Most commonly used to register individual event handlers that provide

the path or stats of the changed files.

When using the chokidar instance directly, you will not have access to the task system integrations, including async completion, queueing, and delay.

```
const { watch } = require('gulp');

const watcher = watch(['input/*.js']);

watcher.on('change', function(path, stats) {
  console.log('File ${path} was changed');
});

watcher.on('add', function(path, stats) {
  console.log('File ${path} was added');
});

watcher.on('unlink', function(path, stats) {
  console.log('File ${path} was removed');
});

watcher.close();
```

`watcher.on(eventName, eventHandler)`

Registers eventHandler functions to be called when the specified event occurs.

parameter	type	note
eventName	string	The events that may be watched are 'add', 'addDir', 'change', 'unlink', 'unlinkDir', 'ready', 'error', or 'all'.
eventHandlerfunction		Function to be called when the specified event occurs. Arguments detailed in the table below.

argument type note

path	string	The path of the file that changed. If the cwd option was set, the path will be made relative by removing the cwd.
stats	object	An fs.Stat object, but could be undefined. If the alwaysStatoption was set to true, stats will always be provided.

`watcher.close()`

Shuts down the file watcher. Once shut down, no more events will be emitted.

`watcher.add(globs)`

Adds additional globs to an already-running watcher instance.

parameter type note

globs	string array	The additional globs to be watched.
-------	-----------------	-------------------------------------

`watcher.unwatch(globs)`

Removes globs that are being watched, while the watcher continues with the remaining paths.

parameter type note

globs	string array	The globs to be removed.
-------	-----------------	--------------------------