

Concepts

The following concepts are prerequisites to understanding the API docs. They will be referenced throughout, refer back to this page for detailed explanations.

If you're new here, begin with the [Getting Started Guide](#).

Vinyl

Vinyl is a metadata object that describes a file. The main properties of a Vinyl instance are path and contents - core aspects of a file on your file system. Vinyl objects can be used to describe files from many sources - on a local file system or any remote storage option.

Vinyl adapters

While Vinyl provides a way to describe a file, a way to access these files is needed. Each file source is accessed using a Vinyl adapter.

An adapter exposes:

- A method with the signature `src(globs, [options])` and returns a stream that produces Vinyl objects.

- A method with the signature `dest(folder, [options])` and returns a stream that consumes Vinyl objects.

- Any extra methods specific to their input/output medium - such as the `symlink` method `vinyl-fs` provides. They should always return streams that produce and/or consume Vinyl objects.

Tasks

Each gulp task is an asynchronous JavaScript function that either accepts an error-first callback or returns a stream, promise, event emitter, child process, or observable. Due to some platform limitations, synchronous tasks aren't supported.

For a more detailed explanation, see [Creating Tasks](#).

Globs

A glob is a string of literal and/or wildcard characters, like `*`, `**`, or `!`, used to match filepaths. Globbing is the act of locating files on a file system using one or more globs.

If you don't have experience with globs, see [Explaining Globs](#).

Glob base

A glob base - sometimes called glob parent - is the path segment before any special characters in a glob string. As such, the glob base of `/src/js/**/*.js` is `/src/js/`. All paths that match the glob are guaranteed to share the glob base - that path segment can't be variable.

Vinyl instances generated by `src()` are constructed with the glob base set as their `base` property. When written to the file system with `dest()`, the base will be removed from the output path to preserve directory structures.

For more in depth information, see the [glob-parent repository](#).

File system stats

File metadata is provided as an instance of Node's `fs.Stats`. It is available as the `stat` property on your Vinyl instances and used internally to determine if a Vinyl object represents a directory or symbolic link. When written to the file system, permissions and time values are synchronized from the Vinyl object's `stat` property.

File system modes

File system modes determine what permissions exist for a file. Most files and directories on your file system will have a fairly permissive mode, allowing gulp to read/write/update files on your behalf. By default, gulp will create files with the same permissions as the running process, but you can configure the modes through options in `src()`, `dest()`, etc. If you're experiencing permission (EPERM) issues, check the modes on your files.

Modules

Gulp is made up of many small modules that are pulled together to work cohesively. By utilizing semver within the small modules, we can release bug fixes and features without publishing new versions of gulp. Often, when you don't see progress on the main repository, work is being done in one of these modules.

If you're having trouble, ensure your current modules are updated using the `npm update` command. If the problem persists, open an issue on the individual project repository.

undertaker - the task registration system

vinyl - the virtual file objects

vinyl-fs - a vinyl adapter to your local file system

glob-watcher - the file watcher

bach - task orchestration using `series()` and `parallel()`

last-run - tracks the last run time of a task

vinyl-sourcemaps - built-in sourcemaps support

gulp-cli - the command line interface for interacting with gulp