

## Vinyl

A virtual file format. When a file is read by `src()`, a Vinyl object is generated to represent the file - including the path, contents, and other metadata.

Vinyl objects can have transformations applied using plugins. They may also be persisted to the file system using `dest()`.

When creating your own Vinyl objects - instead of generating with `src()` - use the external vinyl module, as shown in Usage below.

### Usage

```
const Vinyl = require('vinyl');

const file = new Vinyl({
  cwd: '/',
  base: '/test/',
  path: '/test/file.js',
  contents: new Buffer('var x = 123')
});

file.relative === 'file.js';

file.dirname === '/test';
file.dirname = '/specs';
file.path === '/specs/file.js';

file.basename === 'file.js';
file.basename = 'file.txt';
file.path === '/specs/file.txt';

file.stem === 'file';
file.stem = 'foo';
file.path === '/specs/foo.txt';
file.extname === '.txt';
file.extname = '.js';
file.path === '/specs/foo.js';
```

### Signature

```
new Vinyl([options])
```

#### Parameters

parameter type note

options object Detailed in Options below.

#### Returns

An instance of the Vinyl class representing a single virtual file, detailed in Vinyl instance below.

#### Errors

When any passed options don't conform to the instance property definitions (like if path is set to a number) throws as defined in the table.

#### Options

| name | type   | default       | note  |
|------|--------|---------------|---|
| cwd  | string | process.cwd() | The directory from which relative paths will be derived. Will be normalized and have trailing separators removed. |

| name     | type                             | default | note  |
|----------|----------------------------------|---------|---|
|          |                                  |         | Used to calculate the relativeinstance property. Falls back to the value of cwd if not set.   |
| base     | string                           |         | Typically set to the glob base. Will be normalized and have trailing separators removed.  |
| path     | string                           |         | The full, absolute file path. Will be normalized and have trailing separators removed.  |
| history  | array                            | [ ]     | An array of paths to pre-populate the history of a Vinyl instance. Usually comes from deriving a new Vinyl object from a previous Vinyl object. If pathand history are both passed, path is appended to history. Each item will be normalized and have trailing separators removed. |
| stat     | object                           |         | An instance of fs.Stats, usually the result of calling fs.stat() on a file. Used to determine if a Vinyl object represents a directory or symbolic link.  |
| contents | ReadableStream<br>Buffer<br>null | null    | The contents of the file. If contents is a ReadableStream, it is wrapped in a cloneable-readable stream.  |

Any other properties on options will be directly assigned to the Vinyl instance.

```
const Vinyl = require('vinyl');

const file = new Vinyl({ foo: 'bar' });
file.foo === 'bar';
```

## Vinyl instance

Each instance of a Vinyl object will have properties and methods to access and/or modify information about the virtual file.

### Instance properties

All internally managed paths - any instance property except contents and stat - are normalized and have trailing separators removed. See Normalization and concatenationfor more information.

| property | type                             | description   | throws  |
|----------|----------------------------------|---|---|
|          |                                  | Gets and sets the contents of the virtual file.   |   |
| contents | ReadableStream<br>Buffer<br>null | If set to a ReadableStream, it is wrapped in a cloneable-readable stream.   | If set to any value other than a ReadableStream, a Buffer, or null. |
| stat     | object                           | Gets and sets an instance of fs.Stats. Used when determining if a Vinyl object represents a directory or symbolic link. |   |
| cwd      | string                           | Gets and sets the current working directory. Used for deriving relative paths.  | If set to an empty string or any non-string value.                  |

| property | type   | description  | throws  |
|----------|--------|--|---|
| base     | string | Gets and sets the base directory. Used to calculate the relativeinstance property. On a Vinyl object generated by src() will be set to the glob base. If set to null or undefined, falls back to the value of the cwdinstance property.                                      | If set to an empty string or any non-string value (except null or undefined). |
| path     | string | Gets and sets the full, absolute file path. Setting to a value different from the current pathappends the new path to the historyinstance property.  | If set to any non-string value.   |
| history  | array  | Array of all pathvalues the Vinyl object has been assigned. The first element is the original path and the last element is the current path. This property and its elements should be treated as read-only and only altered indirectly by setting the pathinstance property. |   |
| relative | string | Gets the relative path segment between the base and the path instance properties.  | If set to any value. If accessed when path is not available.                  |
| dirname  | string | Gets and sets the directory of the pathinstance property.  | If accessed when path is not available.                                       |
| stem     | string | Gets and sets the stem (filename without extension) of the path instance property.   | If accessed when path is not available.                                       |
| extname  | string | Gets and sets the extension of the pathinstance property.  | If accessed when path is not available.                                       |
| basename | string | Gets and sets the filename (stem + extname) of the pathinstance property.  | If accessed when path is not available.                                       |
| symlink  | string | Gets and sets the reference path of a symbolic link.   | If set to any non-string value.   |

#### Instance methods

| method     | return type | returns  |
|------------|-------------|--|
| isBuffer() | boolean     | If the contents instance property is a Buffer, returns true. |
| isStream() | boolean     | If the contents instance property is a Stream, returns true. |
| isNull()   | boolean     | If the contents instance property is null, returns true.     |

| method           | return type | returns  |
|------------------|-------------|--|
| isDirectory()    | boolean     | If the instance represents a directory, returns true. An instance is considered a directory when <code>isNull()</code> returns true, the <code>stat</code> instance property is an object, and <code>stat.isDirectory()</code> returns true. This assumes a Vinyl object was constructed with a valid (or properly mocked) <code>fs.Stats</code> object.           |
| isSymbolic()     | boolean     | If the instance represents a symbolic link, returns true. An instance is considered symbolic when <code>isNull()</code> returns true, the <code>stat</code> instance property is an object, and <code>stat.isSymbolicLink()</code> returns true. This assumes a Vinyl object was constructed with a valid (or properly mocked) <code>fs.Stats</code> object.       |
| clone([options]) | object      | A new Vinyl object with all properties cloned. By default custom properties are deep cloned. If the <code>deep</code> option is false, custom attributes will be shallow cloned. If the <code>contents</code> option is false and the <code>contents</code> instance property is a <code>Buffer</code> , the <code>Buffer</code> will be reused instead of cloned. |
| inspect()        | string      | Returns a formatted interpretation of the Vinyl object.<br>Automatically called by Node's <code>console.log</code> .   |

#### Normalization and concatenation

All path properties are normalized by their setters. Concatenate paths with `/`, instead of using `path.join()`, and normalization will occur properly on all platforms. Never concatenate with `\` - it is a valid filename character on POSIX system.

```
const file = new File();
file.path = '/' + 'test' + '/' + 'foo.bar';

console.log(file.path);
// posix => /test/foo.bar
// win32 => \\test\\foo.bar
```