@jridgewell/trace-mapping

Trace the original position through a source map

`trace-mapping` allows you to take the line and column of an output file and trace it to the original location in the source file through a source map.

You may already be familiar with the [`source-map`] package's `SourceMapConsumer`. This provides the same `originalPositionFor` and `generatedPositionFor` API, without requiring WASM.

Installation

```
npm install @jridgewell/trace-mapping
```

Usage

```
import {
  TraceMap,
  originalPositionFor,
  generatedPositionFor,
  sourceContentFor,
  isIgnored,
```

```
} from '@jridgewell/trace-mapping';

const tracer = new TraceMap({
  version: 3,
  sources: ['input.js'],
  sourcesContent: ['content of
        input.js'],
  names: ['foo'],
  mappings: 'KAyCIA',
  ignoreList: [],
});

// Lines start at line 1, columns at
        column 0.
const traced =
        originalPositionFor(tracer,
        { line: 1, column: 5 });
assert.deepEqual(traced, {
  source: 'input.js',
  line: 42,
  column: 4,
  name: 'foo',
});

const content =
        sourceContentFor(tracer,
        traced.source);
assert.strictEqual(content,
        'content for input.js');

const generated =
        generatedPositionFor(tracer, {
  source: 'input.js',
  line: 42,
```

```
  column: 4,
});
assert.deepEqual(generated, {
  line: 1,
  column: 5,
});

const ignored = isIgnored(tracer,
        'input.js');
assert.equal(ignored, false);
```

We also provide a lower level API to get the actual segment that matches our line and column. Unlike `originalPositionFor`, `traceSegment` uses a 0-base for `line`:

```
import { traceSegment } from
        '@jridgewell/trace-mapping';

// line is 0-base.
const traced = traceSegment(tracer, /*
        line */ 0, /* column */ 5);

// Segments are [outputColumn,
        sourcesIndex, sourceLine,
        sourceColumn, namesIndex]
// Again, line is 0-base and so is
        sourceLine
assert.deepEqual(traced, [5, 0, 41, 4,
        0]);
```

SectionedSourceMaps

The sourcemap spec defines a special `sections` field that's designed to handle concatenation of output code with associated sourcemaps. This type of sourcemap is rarely used (no major build tool produces it), but if you are hand coding a concatenation you may need it. We provide an `AnyMap` helper that can receive either a regular sourcemap or a `SectionedSourceMap` and returns a `TraceMap` instance:

```
import { AnyMap } from '@jridgewell/
        trace-mapping';
const fooOutput = 'foo';
const barOutput = 'bar';
const output = [fooOutput,
        barOutput].join('\n');

const sectioned = new AnyMap({
  version: 3,
  sections: [
    {
      // 0-base line and column
      offset: { line: 0, column: 0 },
      // fooOutput's sourcemap
      map: {
        version: 3,
        sources: ['foo.js'],
        names: ['foo'],
        mappings: 'AAAAA',
      },
    },
    {
```



```
source-map-0.8.0: encoded
originalPositionFor x 124,204 ops/sec
±0.90% (92 runs sampled)
Chrome dev tools: encoded
originalPositionFor x 230,087 ops/sec
±2.61% (93 runs sampled)
Fastest is trace-mapping:    decoded
originalPositionFor
```

```
Object input

Trace speed (random):
trace-mapping:    decoded
originalPositionFor x 257,019 ops/sec
±0.97% (93 runs sampled)
trace-mapping:    encoded
originalPositionFor x 179,163 ops/sec
±0.83% (92 runs sampled)
source-map-js:    encoded
originalPositionFor x 73,337 ops/sec
±1.35% (87 runs sampled)
source-map-0.6.1: encoded
originalPositionFor x 38,797 ops/sec
±1.66% (88 runs sampled)
source-map-0.8.0: encoded
originalPositionFor x 107,758 ops/sec
±1.94% (45 runs sampled)
Chrome dev tools: encoded
originalPositionFor x 188,550 ops/sec
±1.85% (79 runs sampled)
Fastest is trace-mapping:    decoded
originalPositionFor

Trace speed (ascending):
trace-mapping:    decoded
originalPositionFor x 447,621 ops/sec
±3.64% (94 runs sampled)
trace-mapping:    encoded
originalPositionFor x 323,698 ops/sec
±5.20% (88 runs sampled)
source-map-js:    encoded
originalPositionFor x 78,387 ops/sec
±1.69% (89 runs sampled)
source-map-0.6.1: encoded
originalPositionFor x 41,016 ops/sec
±3.01% (25 runs sampled)
```



```
        // barOutput's sourcemap will not
           affect the first line, only the
           second
        offset: { line: 1, column: 0 },
        map: {
          version: 3,
          sources: ['bar.js'],
          names: ['bar'],
          mappings: 'AAAAA',
        },
      },
    ],
});

const traced =
        originalPositionFor(sectioned, {
  line: 2,
  column: 0,
});

assert.deepEqual(traced, {
  source: 'bar.js',
  line: 1,
  column: 0,
  name: 'bar',
});
```

Benchmarks

```
node v20.10.0
```

```
amp.js.map - 45120 segments

Memory Usage:
trace-mapping decoded         414164
bytes
trace-mapping encoded        6274352
bytes
source-map-js               10968904
bytes
source-map-0.6.1            17587160
bytes
source-map-0.8.0             8812155
bytes
Chrome dev tools             8672912
bytes
Smallest memory usage is trace-mapping
decoded

Init speed:
trace-mapping:    decoded JSON input x
205 ops/sec ±0.19% (88 runs sampled)
trace-mapping:    encoded JSON input x
405 ops/sec ±1.47% (88 runs sampled)
trace-mapping:    decoded Object input x
4,645 ops/sec ±0.15% (98 runs sampled)
trace-mapping:    encoded Object input x
458 ops/sec ±1.63% (91 runs sampled)
source-map-js:    encoded Object input x
75.48 ops/sec ±1.64% (67 runs sampled)
source-map-0.6.1: encoded Object input x
39.37 ops/sec ±1.44% (53 runs sampled)
Chrome dev tools: encoded Object input x
150 ops/sec ±1.76% (79 runs sampled)
Fastest is trace-mapping:    decoded
Object input

Trace speed (random):
```



```
***


vscode.map - 2141001 segments

Memory Usage:
trace-mapping decoded        5206584
bytes
trace-mapping encoded      208370336
bytes
source-map-js              278493008
bytes
source-map-0.6.1           391564048
bytes
source-map-0.8.0           257508787
bytes
Chrome dev tools           291053000
bytes
Smallest memory usage is trace-mapping
decoded

Init speed:
trace-mapping:    decoded JSON input x
1.63 ops/sec ±33.88% (9 runs sampled)
trace-mapping:    encoded JSON input x
3.29 ops/sec ±36.13% (13 runs sampled)
trace-mapping:    decoded Object input x
103 ops/sec ±0.93% (77 runs sampled)
trace-mapping:    encoded Object input x
5.42 ops/sec ±28.54% (19 runs sampled)
source-map-js:    encoded Object input x
1.07 ops/sec ±13.84% (7 runs sampled)
source-map-0.6.1: encoded Object input x
0.60 ops/sec ±2.43% (6 runs sampled)
Chrome dev tools: encoded Object input x
2.61 ops/sec ±22.00% (11 runs sampled)
Fastest is trace-mapping:    decoded
```

```
originalPositionFor x 230,983 ops/sec
±0.62% (54 runs sampled)
source-map-0.6.1: encoded
originalPositionFor x 158,145 ops/sec
±0.80% (46 runs sampled)
source-map-0.8.0: encoded
originalPositionFor x 343,801 ops/sec
±0.55% (96 runs sampled)
Chrome dev tools: encoded
originalPositionFor x 659,649 ops/sec
±0.49% (98 runs sampled)
Fastest is Chrome dev tools: encoded
originalPositionFor

Trace speed (ascending):
trace-mapping:    decoded
originalPositionFor x 2,368,079 ops/sec
±0.32% (98 runs sampled)
trace-mapping:    encoded
originalPositionFor x 2,134,039 ops/sec
±2.72% (87 runs sampled)
source-map-js:    encoded
originalPositionFor x 290,120 ops/sec
±2.49% (82 runs sampled)
source-map-0.6.1: encoded
originalPositionFor x 187,613 ops/sec
±0.86% (49 runs sampled)
source-map-0.8.0: encoded
originalPositionFor x 479,569 ops/sec
±0.65% (96 runs sampled)
Chrome dev tools: encoded
originalPositionFor x 2,048,414 ops/sec
±0.24% (98 runs sampled)
Fastest is trace-mapping:    decoded
originalPositionFor
```



```
trace-mapping:    decoded
originalPositionFor x 44,946 ops/sec
±0.16% (99 runs sampled)
trace-mapping:    encoded
originalPositionFor x 37,995 ops/sec
±1.81% (89 runs sampled)
source-map-js:    encoded
originalPositionFor x 9,230 ops/sec
±1.36% (93 runs sampled)
source-map-0.6.1: encoded
originalPositionFor x 8,057 ops/sec
±0.84% (96 runs sampled)
source-map-0.8.0: encoded
originalPositionFor x 28,198 ops/sec
±1.12% (91 runs sampled)
Chrome dev tools: encoded
originalPositionFor x 46,276 ops/sec
±1.35% (95 runs sampled)
Fastest is Chrome dev tools: encoded
originalPositionFor

Trace speed (ascending):
trace-mapping:    decoded
originalPositionFor x 204,406 ops/sec
±0.19% (97 runs sampled)
trace-mapping:    encoded
originalPositionFor x 196,695 ops/sec
±0.24% (99 runs sampled)
source-map-js:    encoded
originalPositionFor x 11,948 ops/sec
±0.94% (99 runs sampled)
source-map-0.6.1: encoded
originalPositionFor x 10,730 ops/sec
±0.36% (100 runs sampled)
source-map-0.8.0: encoded
originalPositionFor x 51,427 ops/sec
±0.21% (98 runs sampled)
```

```
Chrome dev tools: encoded
originalPositionFor x 162,615 ops/sec
±0.18% (98 runs sampled)
Fastest is trace-mapping:    decoded
originalPositionFor


***


babel.min.js.map - 347793 segments

Memory Usage:
trace-mapping decoded          18504
bytes
trace-mapping encoded       35428008
bytes
source-map-js               51676808
bytes
source-map-0.6.1            63367136
bytes
source-map-0.8.0            43158400
bytes
Chrome dev tools            50721552
bytes
Smallest memory usage is trace-mapping
decoded

Init speed:
trace-mapping:    decoded JSON input x
17.82 ops/sec ±6.35% (35 runs sampled)
trace-mapping:    encoded JSON input x
31.57 ops/sec ±7.50% (43 runs sampled)
trace-mapping:    decoded Object input x
867 ops/sec ±0.74% (94 runs sampled)
trace-mapping:    encoded Object input x
33.83 ops/sec ±7.66% (46 runs sampled)
```



```
source-map-js               2563944
bytes
source-map-0.6.1            2150864
bytes
source-map-0.8.0              88680
bytes
Chrome dev tools            1149576
bytes
Smallest memory usage is trace-mapping
decoded

Init speed:
trace-mapping:    decoded JSON input x
1,887 ops/sec ±0.28% (99 runs sampled)
trace-mapping:    encoded JSON input x
4,749 ops/sec ±0.48% (97 runs sampled)
trace-mapping:    decoded Object input x
74,236 ops/sec ±0.11% (99 runs sampled)
trace-mapping:    encoded Object input x
5,752 ops/sec ±0.38% (100 runs sampled)
source-map-js:    encoded Object input x
806 ops/sec ±0.19% (97 runs sampled)
source-map-0.6.1: encoded Object input x
418 ops/sec ±0.33% (94 runs sampled)
Chrome dev tools: encoded Object input x
1,524 ops/sec ±0.57% (92 runs sampled)
Fastest is trace-mapping:    decoded
Object input

Trace speed (random):
trace-mapping:    decoded
originalPositionFor x 620,201 ops/sec
±0.33% (96 runs sampled)
trace-mapping:    encoded
originalPositionFor x 579,548 ops/sec
±0.35% (97 runs sampled)
source-map-js:    encoded
```

```
±0.14% (98 runs sampled)
Fastest is trace-mapping:    decoded
originalPositionFor

Trace speed (ascending):
trace-mapping:    decoded
originalPositionFor x 319,960 ops/sec
±0.16% (100 runs sampled)
trace-mapping:    encoded
originalPositionFor x 302,153 ops/sec
±0.18% (100 runs sampled)
source-map-js:    encoded
originalPositionFor x 35,574 ops/sec
±0.19% (100 runs sampled)
source-map-0.6.1: encoded
originalPositionFor x 19,943 ops/sec
±0.12% (101 runs sampled)
source-map-0.8.0: encoded
originalPositionFor x 54,648 ops/sec
±0.20% (99 runs sampled)
Chrome dev tools: encoded
originalPositionFor x 278,319 ops/sec
±0.17% (102 runs sampled)
Fastest is trace-mapping:    decoded
originalPositionFor


***


react.js.map - 5726 segments

Memory Usage:
trace-mapping decoded          10872
bytes
trace-mapping encoded         681512
bytes
```



```
source-map-js:    encoded Object input x
6.58 ops/sec ±3.31% (20 runs sampled)
source-map-0.6.1: encoded Object input x
4.23 ops/sec ±3.43% (15 runs sampled)
Chrome dev tools: encoded Object input x
22.14 ops/sec ±3.79% (41 runs sampled)
Fastest is trace-mapping:    decoded
Object input

Trace speed (random):
trace-mapping:    decoded
originalPositionFor x 78,234 ops/sec
±1.48% (29 runs sampled)
trace-mapping:    encoded
originalPositionFor x 60,761 ops/sec
±1.35% (21 runs sampled)
source-map-js:    encoded
originalPositionFor x 51,448 ops/sec
±2.17% (89 runs sampled)
source-map-0.6.1: encoded
originalPositionFor x 47,221 ops/sec
±1.99% (15 runs sampled)
source-map-0.8.0: encoded
originalPositionFor x 84,002 ops/sec
±1.45% (27 runs sampled)
Chrome dev tools: encoded
originalPositionFor x 106,457 ops/sec
±1.38% (37 runs sampled)
Fastest is Chrome dev tools: encoded
originalPositionFor

Trace speed (ascending):
trace-mapping:    decoded
originalPositionFor x 930,943 ops/sec
±0.25% (99 runs sampled)
trace-mapping:    encoded
originalPositionFor x 843,545 ops/sec
```

```
±0.34% (97 runs sampled)
source-map-js:    encoded
originalPositionFor x 114,510 ops/sec
±1.37% (36 runs sampled)
source-map-0.6.1: encoded
originalPositionFor x 87,412 ops/sec
±0.72% (92 runs sampled)
source-map-0.8.0: encoded
originalPositionFor x 197,709 ops/sec
±0.89% (59 runs sampled)
Chrome dev tools: encoded
originalPositionFor x 688,983 ops/sec
±0.33% (98 runs sampled)
Fastest is trace-mapping:    decoded
originalPositionFor


***


preact.js.map - 1992 segments

Memory Usage:
trace-mapping decoded          33136
bytes
trace-mapping encoded         254240
bytes
source-map-js                 837488
bytes
source-map-0.6.1              961928
bytes
source-map-0.8.0               54384
bytes
Chrome dev tools              709680
bytes
Smallest memory usage is trace-mapping
decoded
```



```
Init speed:
trace-mapping:    decoded JSON input x
3,709 ops/sec ±0.13% (99 runs sampled)
trace-mapping:    encoded JSON input x
6,447 ops/sec ±0.22% (101 runs sampled)
trace-mapping:    decoded Object input x
83,062 ops/sec ±0.23% (100 runs sampled)
trace-mapping:    encoded Object input x
14,980 ops/sec ±0.28% (100 runs sampled)
source-map-js:    encoded Object input x
2,544 ops/sec ±0.16% (99 runs sampled)
source-map-0.6.1: encoded Object input x
1,221 ops/sec ±0.37% (97 runs sampled)
Chrome dev tools: encoded Object input x
4,241 ops/sec ±0.39% (93 runs sampled)
Fastest is trace-mapping:    decoded
Object input

Trace speed (random):
trace-mapping:    decoded
originalPositionFor x 91,028 ops/sec
±0.14% (94 runs sampled)
trace-mapping:    encoded
originalPositionFor x 84,348 ops/sec
±0.26% (98 runs sampled)
source-map-js:    encoded
originalPositionFor x 26,998 ops/sec
±0.23% (98 runs sampled)
source-map-0.6.1: encoded
originalPositionFor x 18,049 ops/sec
±0.26% (100 runs sampled)
source-map-0.8.0: encoded
originalPositionFor x 41,916 ops/sec
±0.28% (98 runs sampled)
Chrome dev tools: encoded
originalPositionFor x 88,616 ops/sec
```