

Appendix E. Triple-Slash Directives

Triple-slash directives are just regular JavaScript comments that TypeScript looks for to do things like adjust compiler settings for a specific file, or indicate that your file depends on another file. Put your directives at the top of your file, before any code. Triple-slash directives look like this (each directive is a triple-slash, `///`, followed by an XML tag):

```
/// <directive attr="value" />
```

TypeScript supports a handful of triple-slash directives. [Table E-1](#) lists the ones you are most likely to use:

`amd-module`

Head over to [“The amd-module Directive”](#) to learn more.

`lib`

The `lib` directive is a way to indicate to TypeScript which of TypeScript’s `lib`s your module depends on, which you may want to do if your project doesn’t have a *tsconfig.json*. Declaring the `lib`s you depend on in your *tsconfig.json* is almost always a better option.

`path`

When using TSC’s `outFile` option, use the `path` directive to declare a dependency on another file, so that the other file appears earlier in your compiled output than the dependent file does. If your project uses `import`s and `export`s, you likely won’t ever use this directive.

`type`

Head over to [“The types Directive”](#) to learn more about the `type` directive.

Table E-1. Triple-slash directives

Directive	Syntax	Use it to...
<code>amd-module</code>	<code><amd-module name="MyComponent" /></code>	Declare export names when compiling to AMD modules
<code>lib</code>	<code><reference lib="dom" /></code>	Declare which of TypeScript's built-in <code>lib</code> s your type declarations depend on
<code>path</code>	<code><reference path="./path.ts" /></code>	Declare which TypeScript files your module depends on
<code>type</code>	<code><reference types="./path.d.ts" /></code>	Declare which type declaration files your module depends on

Internal Directives

You will probably never use the `no-default-lib` directive ([Table E-2](#)) in your own code.

Table E-2. Internal triple-slash directives

Directive	Syntax	Use it to...
<code>no-default-lib</code>	<code><reference no-default-lib="true" /></code>	Tell TypeScript to not use any <code>lib</code> s at all for this file

Deprecated Directives

You should never use the `amd-dependency` directive ([Table E-3](#)), and instead stick to a regular `import`.

Table E-3. Deprecated triple-slash directives

Directive	Syntax	Instead use...
amd- dependency	<pre><amd-dependency path="./a.ts" name="MyComponent" /></pre>	import