

DllReference

[DllPlugin documentation](#)

This is the *reference* bundle (with the manifests) for [dll user example](#)

webpack.config.js

```
var path = require("path");
var webpack = require("../..");
module.exports = {
  // mode: "development" || "production",
  resolve: {
    extensions: [".js", ".jsx"]
  },
  entry: {
    alpha: ["/alpha", "/a", "module"],
    beta: ["/beta", "/b", "/c"]
  },
  output: {
    path: path.join(__dirname, "dist"),
    filename: "MyDll.[name].js",
    library: "[name]_[fullhash]"
  },
  plugins: [
    new webpack.DllPlugin({
      path: path.join(__dirname, "dist", "[name]-manifest.json"),
      name: "[name]_[fullhash]"
    })
  ]
};
```

dist/MyDll.alpha.js

```
var alpha_32ae439e7568b31a353c;
/******/ (() => { // webpackBootstrap
/******/   var __webpack_modules__ = ([
/* 0 */
/*!*****!\
*** dll alpha ***!
\*****/
/*! unknown exports (runtime-defined) */
/*! runtime requirements: __webpack_require__, module */
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

module.exports = __webpack_require__;
```

```

/***/ }},
/* 1 */
/*!*****!\
  !*** ./alpha.js ***!
  \*****/
/*! unknown exports (runtime-defined) */
/*! runtime requirements: module */
/*! CommonJS bailout: module.exports is used directly at 1:0-14 */
/***/ ((module) => {

  module.exports = "alpha";

/***/ }},
/* 2 */
/*!*****!\
  !*** ./a.js ***!
  \*****/
/*! unknown exports (runtime-defined) */
/*! runtime requirements: module */
/*! CommonJS bailout: module.exports is used directly at 1:0-14 */
/***/ ((module) => {

  module.exports = "a";

/***/ }},
/* 3 */
/*!*****!\
  !*** ../node_modules/module.js ***!
  \*****/
/*! unknown exports (runtime-defined) */
/*! runtime requirements: module */
/*! CommonJS bailout: module.exports is used directly at 1:0-14 */
/***/ ((module) => {

  module.exports = "module";

/***/ })
/*****/
    ]);

```

► /* webpack runtime code */

```

/*****/
/*****/          // startup
/*****/          // Load entry module and return exports
/*****/          // This entry module doesn't tell about it's top-level declarations
so it can't be inlined
/*****/          var __webpack_exports__ = __webpack_require__(0);
/*****/          alpha_32ae439e7568b31a353c = __webpack_exports__;
/*****/
/*****/ } } ()
;

```

dist/alpha-manifest.json

```
{ "name": "alpha_32ae439e7568b31a353c", "content": { "./alpha.js": { "id": 1, "buildMeta": {} }, "./a.js": { "id": 2, "buildMeta": {} }, "../node_modules/module.js": { "id": 3, "buildMeta": {} } } }
```

Info

Unoptimized

```
asset MyDll.alpha.js 2.58 KiB [emitted] (name: alpha)
asset MyDll.beta.js 2.55 KiB [emitted] (name: beta)
chunk (runtime: alpha) MyDll.alpha.js (alpha) 84 bytes [entry] [rendered]
  > alpha
  dependent modules 72 bytes [dependent] 3 modules
  dll alpha 12 bytes [built] [code generated]
    [used exports unknown]
  dll entry
  used as library export
chunk (runtime: beta) MyDll.beta.js (beta) 80 bytes [entry] [rendered]
  > beta
  dependent modules 68 bytes [dependent] 3 modules
  dll beta 12 bytes [built] [code generated]
    [used exports unknown]
  dll entry
  used as library export
webpack 5.51.1 compiled successfully
```

Production mode

```
asset MyDll.alpha.js 313 bytes [emitted] [minimized] (name: alpha)
asset MyDll.beta.js 303 bytes [emitted] [minimized] (name: beta)
chunk (runtime: alpha) MyDll.alpha.js (alpha) 84 bytes [entry] [rendered]
  > alpha
  dependent modules 72 bytes [dependent] 3 modules
  dll alpha 12 bytes [built] [code generated]
    dll entry
    used as library export
chunk (runtime: beta) MyDll.beta.js (beta) 80 bytes [entry] [rendered]
  > beta
  dependent modules 68 bytes [dependent] 3 modules
  dll beta 12 bytes [built] [code generated]
    dll entry
    used as library export
webpack 5.51.1 compiled successfully
```