

使用mop

[Mop](#)是MontageJS用来将应用从开发阶段转换成产品阶段的命令行工具。在转换过程中Mop会压缩JavaScript，HTML和CSS来减少应用的加载时间。Mop的预加载是异步的，应用的主体框架加载完成后，用户就能够看到应用界面，同时异步加载其它模块，这样的方式保证应用界面能够很快显示出来。

mop工作原理

当你完成项目的编码工作之后，你的项目文件夹中包含：

- 引导程序
- 初始化模块以及它们的依赖
- 你编写的程序

在一个大项目中以上文件数量非常巨大。Mop会重组这些文件：

- 首先标识出你项目中引用了哪些MontageJS模块。
- 然后压缩源码，把所有的源码和依赖库打包。打包生成的文件包含应用运行需要的最小依赖，可以把打好的包部署到服务器。

把项目打包有利于在浏览器中运行项目，这样就不需要每个模块都通过一个HTTP请求来加载，只需要加载少数的几个打包文件就可以了（你可以使用调试工具查看加载了哪些打包文件）。

安装mop

Mop不包含在MontageJS安装文件中。你可以通过下面的步骤安装它：

打开命令行窗口运行

```
npm install -g mop
```

Mop会被全局安装到你的系统中，这样你就可以在任意的项目中使用它。

使用 mop

你可以在MontageJS项目中的任意目录运行mop。

1. 在命令行中切换到项目目录。
2. 在命令行窗口输入： `mop` 然后按回车键。

Mop分析源码的依赖关系，标识出应用引用了哪些模块，然后在项目目录中创建一个当前版本对应的打包文件夹。在打包文件夹中包含应用运行需要的最小依赖。

部署应用：双击 `builds/app-name` 文件夹连接，复制全部文件到web服务器。