Javascript源码混淆压缩工具

## 一、目前比较流行的Javascript源码混淆压缩工具主要有以下四种：

1. JSMin

JSMin 仅对代码进行压缩，并不混淆。它的工作原理是去除Javascript文件中所有不必要的字符、注释和空白。

1. Dojo Shinksafe

Shinksafe也是仅对代码进行压缩。它的工作原理是去除Javascript文件中的空白，并替换变量名，但是不替换内部方法和属性名。

1. YUI Compressor

YUI Compressor 对代码进行压缩，几乎没有混淆。它的工作原理是：

1. 移除注释
2. 移除额外的空格
3. 细微优化代码
4. 标示符替换（仅替换函数名和变量名，其他的，如原始值、全局变量、属性名、关键字并不替换）

YUI Compressor 相比其他三款工具使用更为稳定的一款代码压缩工具，且使用方便。

1. Packer

Packer 对代码进行压缩，并进行混淆。它有.net/perl/wsh/php几个版本。在windows平台下，使用wsh版本更合适，并且不需要再安装任何插件，使用方便。

设置压缩/混淆的级别：

1）None

简单压缩，移除空格和注释，编码特殊字符，

2）Numeric (Base 10)

所有字符将转换成数字。

3）Normal (Base 62)

所有字符将转换成数字字母，这个是推荐设置。

4）High ASCII (Base 95)

将获得更高的压缩比率。（文件更小）

如果你使用这个选项，那么压缩后的代码必须使用 ISO-8859-1 编码，这是由于一个 IE 的 BUG 所致，内容页也需要同样的编码，如果你不能确定，那就使用 "Normal" 。

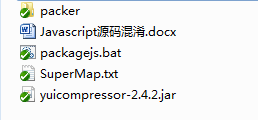
我们默认采用Normal级别。

## 二、混淆压缩过程

由于Packer工具自身对需要混淆的源码代码规范要求比较高，结果导致我们的源码在直接用Packer混淆压缩后出现错误，无法正常使用。因此，我们采用YUI Compressor和Packer联合应用的方式，先采用YUI Compressor 对源码进行压缩，且在压缩过程中对源码进行了细微的优化；然后用Packer对上一步的压缩结果进行混血压缩操作，最终得到混淆后的代码。

1. 目录结构

1）..\ Transformers10\compressor文件夹



需要混淆的源码列表

YUI Compressor压缩批处理程序

Packer工具

2）..\Transformers10\compressor\packer



Packer混淆压缩批处理程序

1. 操作步骤：

1）如果有新增的源码文件，先更新SuperMap.txt文件；

2）运行..\Transformers10\compressor文件夹下的packagejs.bat；

3）运行..\Transformers10\compressor\packer 文件夹下的pack.bat。

**最终在..\Transformers10\libs 目录下生成混淆压缩后的源码文件SuerpMap.js文件。**