**压缩工具问答**

**Q：常见应用压缩比**

**A：**一般情况下，加入x86后压缩和原始arm大小基本一致。以下是5个常见arm应用的压缩比较

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 应用 | 手机淘宝 | 百度 | 美颜相机 | 爱奇艺PPS | 微信 |
| 压缩前（kb） | 30352 | 19889 | 21029 | 20086 | 28506 |
| 压缩后（kb） | 27422 | 17893 | 18942 | 17392 | 27256 |

已经使用此工具的应用<http://shouji.baidu.com/soft/item?docid=7252981&from=as&f=search_app_moboplayer2%40listsp_1_title%401%40header_all_input>

下载这个应用，体验一下。压缩对用户使用完全没有任何影响

**Q：如何整合压缩sdk**

**A：**整合很简单，但是要注意的是一定要确保在load库之前解压完成，可以使用DecRawso.GetInstance().waitdecoding();在DecRawso.NewInstance之后，并且一定要保证DecRawso.NewInstance尽可能的前。

**Q：在service内load library**

**A：**因为service是在不同的进程空间，需要在service启动时调用DecRawso.NewInstanceInService

**Q：整合x86的库**

**A：**一定要确保在尽可能前的地方调用DecRawso.NewInstance，然后调用waitdecoding阻塞应用。或者可以仅仅把x86放在网络侧，压缩工具-noarm表示不压缩arm库，-x86 httplink address 表示把x86库放在网络侧。（此工具并不提供网络存储空间）

**Q：提示porting error**

**A：**工具会检测arm库和x86库的交叉引用，以及可能存在的x86库缺库现象，请根据提示修复这些问题。 X86库直接调用arm的库是有风险的，因为x86和arm的数据并不一致，比如double在arm上是8字节，但是在x86上是4字节

**Q：java压缩工具的使用**

**A：**最简单的用法 java –jar ApkCompress.jar –a [your apk path]. 会使用eclipse的默认android签名。所以保证要安装了android sdk + eclipse