# 代码热更新

目前项目中的代码热更新方案使用的是InjectFix，选择该技术的原因是我们目前没有大量热更新功能的需求，一般上线之后都是为了修复BUG，这个用起来比较方便；

## 客户端热更新逻辑

客户端首先加载本地资源，这个时候文字表可能还不存在，所以所有的文字都加载Assets\AssetBundles\DayZ\PlatformDefault\Default\Config\PreLanguage这个文件夹的预设文字，这些文字都用在热更新界面，后面遇到问题了可以看下这个；

## InjectFix的使用

在Unity中，点击菜单栏的InjectFix按钮，会弹出选项，包含一个Inject和三个Fix，每一个Fix对应不同的平台。

InjectFix的原理是在代码中注入桩代码，然后可以通过Map表查询补丁代码进行更新。

注入逻辑已经写在了代码自动打包的逻辑中，不用管。

生成补丁需要在打包之前手动操作，然后会在Assets\AssetBundles\DayZ\PlatformDefault\Default\InjectFix这个文件夹下，生成补丁文件，然后就可以用我们的AB打包工具生成并更新。

代码中加载热更新的地方是InjectFixManager这个脚本，具体逻辑可以自行查看。

## DataAtlas更新

正常都是通过AB更新的方式来热更，但是如果只是配置表错了，提供了一个默认的服务器下发的方式更新，就不用走AB更新流程比较方便。代码在DataAtlasManager中的 public void OverrideServerData(List<object> modifyDatas, string lang)可以看。

## 未来的替换策略

如果未来有复杂的热更新需求，那么建议升级成HybirdCLR，这个框架不用区分热更新程序集和非热更新程序集，更加适合我们的项目。