**[工行2013年3月（ctp5项目）：关于class文件热更新分享](http://wiki.primeton.com/pages/viewpage.action?pageId=11632732)**

一.分享的动力

      看过《新人入职指引》一文，颇有感慨，尤其是其中一点：

“不懂-------->去做”，在这样的过程中你不会得到锻炼，

最好的过程是“不懂------------->改进”。

      感觉自己分享太少，故将在工行（ctp5项目）做的一个class文件热更新需求分享如下，希望各位前辈多多批评指教。

二.分享内容简介

      1)背景

           \* 工行这次有一个ctp应用的class文件热更新需求：java类、javastep和opstep（类似于EOS的逻辑单元）对应的编译文件。

           \* 之前的ctp4中每个ctp应用都是一个标准的J2EE部署结构，由于本次升级考虑要将应用中对应的java类热更新（以达到不用

                 有一些小改动之类的操作都要重启服务），顾增加一个动态类目录（dynclasses），部署ctp应用时将需要热更新的类部署

                 到此目录下。

      2)解决方案

           \* 自定义类加载器，将此类加载固定classpath为动态类目录（dynclasses）。

           \* ctp server启动时，先去扫描此目录记录每个class文件的时间戳（根据时间戳的变化判断需热更新的class文件）。

           \* 与消息事件驱动热更新的区别是，ctp5采用控制台发生MBean消息的方式进行热更新。

          \* 类似EOS的逻辑单元会缓存在逻辑流中类似，ctp中的javastep与opstep会缓存到交易流中（Operation），

                 顾需要一个映射关系来记录每个javastep或opstep都被哪个交易流使用，以达到热更新时卸载掉之前Operation中的缓存并

                 替换更新后的javastep或opstep。

           \* 自定义类加载器采用jsp热更新的方式，每个class文件对应一个类加载器，已达到充分卸载更新的类。

           \* 目前只支持无状态的java类的热更新。

           \* 热更新过程的前提是需安全停机或在查看目前正在执行的交易确保无影响后强制执行。安全停机：即跑完所有交易，再停机。

      3)测试

           \* 生成了10000个java类进行测试，用jconsole观察非堆内存的变化情况，确定已经回收。

           \* 10000个opstep类和ctp server的400左右个类：非堆内存最大43M并保持，

                 堆内存最大150M（之后稳定在130M左右），装入类13911个。

      4)总结

           解决class热更新问题需要考虑的几个方面：

                 \* 一般需要自定义类加载器。

                 \* 查找需更新class的缓存并卸载。

                 \* 生成大量的类进行测试，观察内存是否回收了。

三.感谢

      感谢陈文权和洪水侨老师的指导。

四.代码

    [^ctp-classloader.zip]