1. 初始化配置文件（对于新增应用）
2. 配置文件放置的位置在install.xml中使用INIT\_FILES参数指定，一个初始化文件对应一个配置文件，若是有多个初始化文件，则在INIT\_FILES参数中指定多个配置文件
3. 配置文件结构：

APP\_CONFIG：项目中需要初始化的配置文件（/项目名/文件在项目的位置）

CONFIG\_TEMPLET：模板文件在补丁程序中的位置

TEMPLET\_variables:模板中需要替换的变量，变量名必须与IContext中的一致

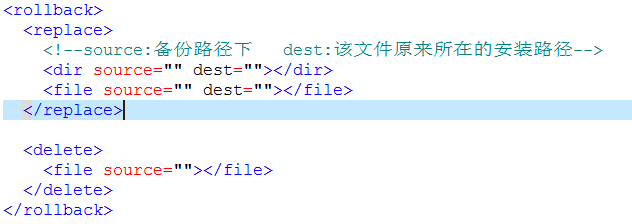
1. 修改应用服务器下以及bosssoft\_home目录下的项目配置文件
2. 回滚的处理
3. 需要备份的文件：resource\_info.xml+install.xmlh中的参数extra\_File(需要额外备份的文件)+版本信息文件

备份目录：insatll.xml中的参数BACKUP\_DIR指定(必有参数)

备份操作：根据备份文件创建回滚的配置文件（固定为：rollback/rollback.xml）：

记录如下替换文件：

升级的资源文件+extra\_File+app版本信息以及产品版本信息文件



War:dest=备份目录+appName

资源文件：dest=备份目录+appName+文件名

extra\_File：dest=source中将磁盘名那一部分替换为备份目录

版本文件：dest=备份目录+“version”+原版本信息名

1. 回滚：

增：记录需要删除的操作，最后将信息写入到文件中

操作：

根据删除记录文件和备份配置文件执行相应操作

（1）抛出异常时，执行回滚

运行过程中报错：

1. 执行相应action的rollback

整体流程：

1. 加载补丁配置文件（resource\_info.xml）
2. 配置文件结构:



1. 处理流程

（1）根据<product>获取产品名---->读取BOSSSOFT\_HOME目录下的产品安装信息文件productname\_version.xml--->从productname\_version.xml中获取产品版本、安装路径，应用部署路径、应用服务端口，将这些信息存入context中

（2）解析<app>标签，构造PatchApp对象（详细）

二、检查是否可更新

1.配置文件（固定位置：install/depend.xml）



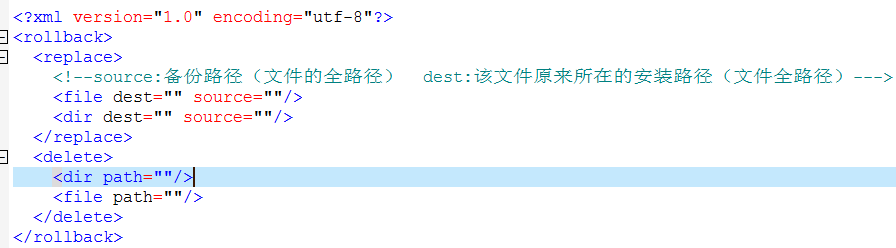
三、备份

1.整体流程

获取所有需要备份的文件以及备份目录，以此构建配置文件rollback.xml-->读取rollback.xml执行文件拷贝

2.配置文件（rollback.xml）固定位置：rollback/rollback.xml

（该步骤只创建<replace>标签）



3.详细步骤

（1）构建配置文件rollback.xml

备份目录：install.xml中使用BACKUP\_DIR参数配置

需要备份的文件：升级的应用war包 +应用和产品的版本信息文件（B）+由于新增应用而需要修改的一些配置文件（C）

A：遍历所有PatchApp，若是已安装，则需要备份

B：context中获取产品名，以及所有需要升级的应用名，拼接版本信息文件

C：install.xml中使用EXTRA\_FILES参数配置，使用逗号分隔多个文件

1. 执行文件拷贝

四、更新应用

（1）流程

对于每一个iType的实现类：

记录回滚时需要删除的文件--->执行更新

需要删除的文件：（保存在Recorder类中的集合成员变量中）

War：应用服务器下程序包+外部路径下的应用

Resource：应用服务器下对应的文件

执行更新：在更新的同时，使用Recorder类记录操作日志。

五、更新版本信息

1.版本信息配置文件version.xml。（固定位置：install/version.xml）



2.流程

解析version.xml，若外部路径下没有对应应用的版本信息文件，则新建文件；否则修改

六、完善更新

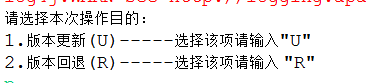
1、流程

Recorder类将操作日志保存到context中-->Recorder类将所记录的回滚时需要删除的文件写入到rollback.xml中。（若此时rollback.xml不存在，则需要创建该文件）

七．回滚：

1、程序运行时选择回滚：

非图形化：



2、产品更新时抛出异常：（UpdateProduct和UpdateVersion的rollback（））

Recorder类将所记录的回滚时需要删除的文件写入到rollback.xml中--->实例化RollBack类，执行回滚操作

