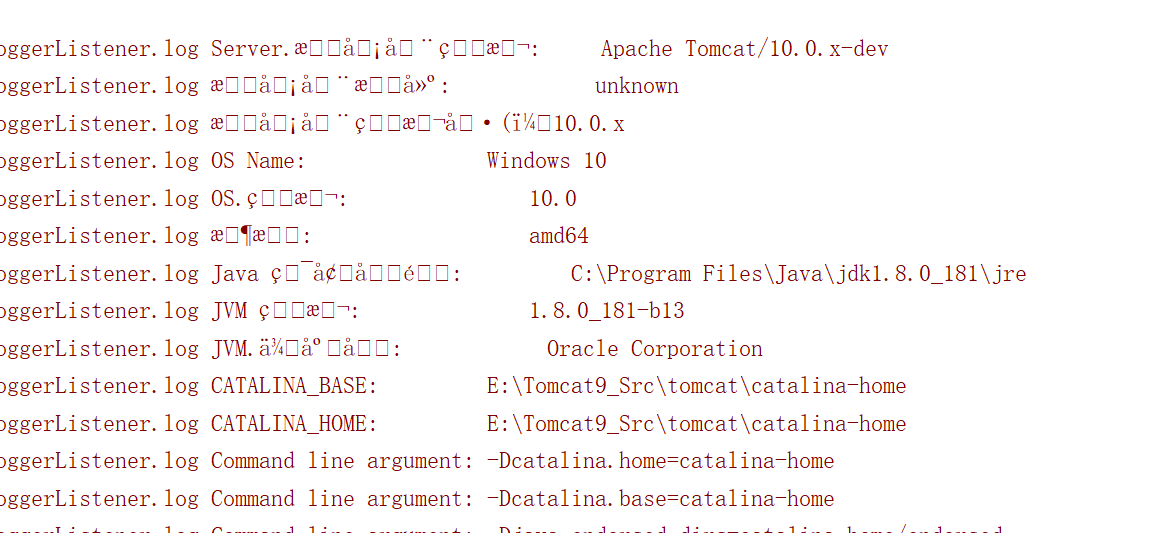
遇到的乱码问题：



5. 可能出现的问题

  tomcat 显示乱码，-Dfile.encoding=UTF8 或者GBK 都不行，网上各种回答都不能根本解决问题。调试VersionLoggerListener进行发现



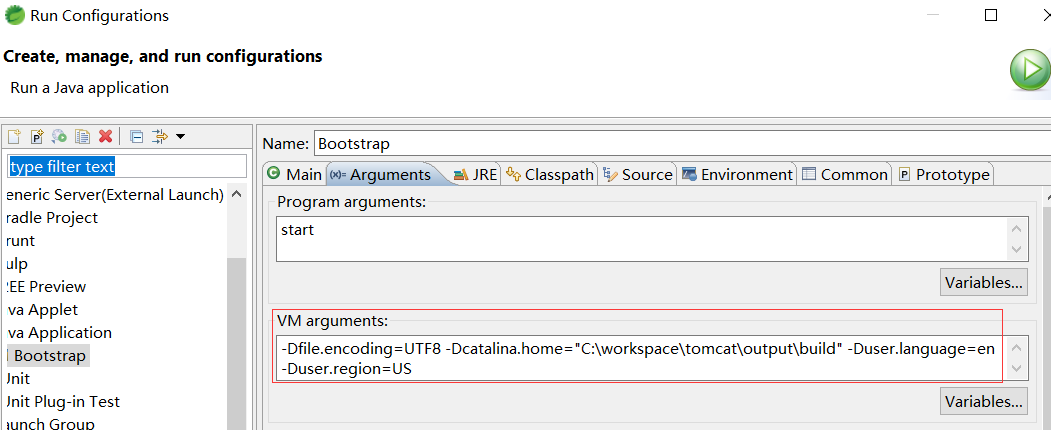
修改-Dfile.encoding=UTF8 或者GBK 都 能起效，但乱码仍然不能解决。

进入StringManager sm变量发现，local为zh\_cn时，启用的属性文件本身的编码就是乱码，尝试了不少解决办法如修改eclipse的默认方式为utf-8 都不行，网上各种方式也试过。

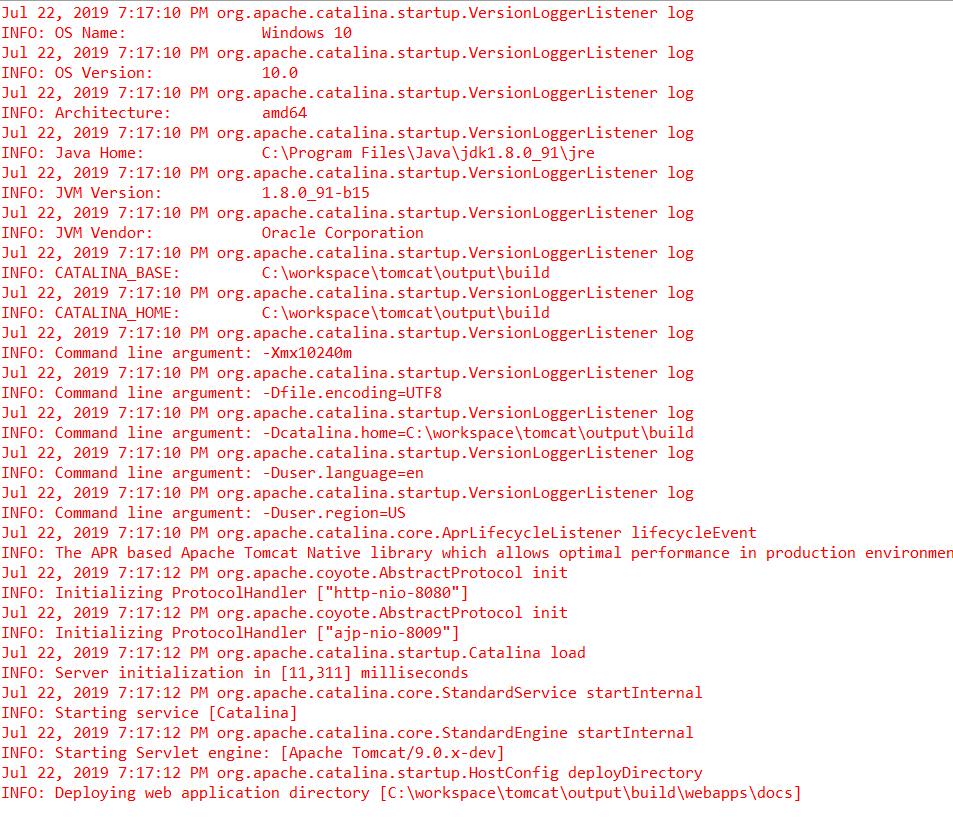
均不好用。那我们就绕过去吧，将local设置为us不容易报错。修改启动参数，如下所示。

-Duser.language=en

-Duser.region=US



 此时，启动显示如下：（全部英文，不再乱码）



参考资料：

【1】<https://www.cnblogs.com/davidwang456/articles/11068822.html>

【2】<http://tomcat.apache.org/tomcat-9.0-doc/building.html>

**记一次tomcat源码启动控制台中文乱码问题调试过程**

<https://blog.csdn.net/zhoutaoping1992/article/details/104751705/>

今天准备学习下tomcat源码，于是从官网下载了tomcat的源码，导入到IDEA中，使用maven工具build完项目之后，启动项目，控制台打印了tomcat日志，但是中文都是乱码。

一开始我怀疑是IDEA的问题，于是在网上找了各种解决办法尝试。大致有这几种：

1、修改run/debug configurations，添加VM options参数：-Dfile.encoding=utf-8;

2、修改run/debug configurations，添加Enviroment variables参数：JAVA\_TOOL\_OPTIONS:-Dfile.encoding=utf-8和JAVA\_OPTS:-Dfile.encoding=utf-8;

3、修改IDEA配置file encodings的3处编码为UTF-8;

4、修改IDEA的Custom VM options，添加-Dfile.encoding=utf-8;

5、修改IDEA的安装目录bin下的idea.exe.vmoptions和idea64.exe.vmoptions文件，添加-Dfile.encoding=utf-8；

6、修改项目下的.idea文件夹下的encodings.xml文件，不是UTF-8的改为UTF-8;

7、修改tomcat的配置文件logging.properties，将里面的UTF-8改为GBK;

8、修改完删除target文件夹重新编译;

9、修改完重启IDEA。

尝试完所有方法后，控制台日志乱码问题并没有解决，如图：

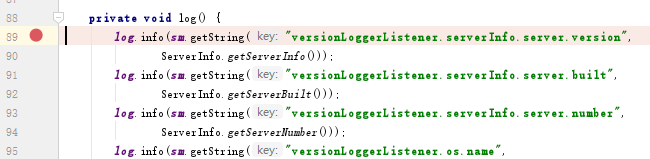


仔细观察后，发现日志左边的日志等级”信息”和”严重”之类的中文乱码解决了，但是日志中还有乱码。

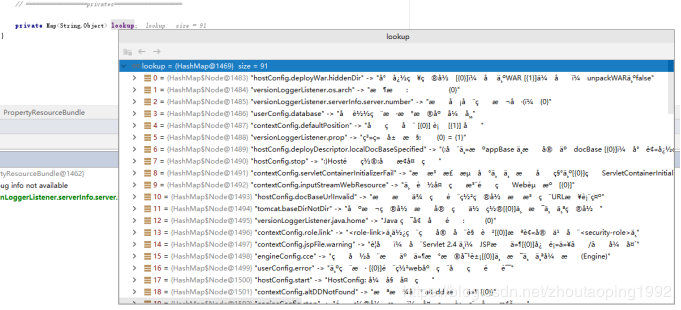
感觉应该是代码的问题，于是决定debugger代码，先从日志的第一行开始。

17-Feb-2020 10:10:08.585 信息 [main] org.apache.catalina.startup.VersionLoggerListener.log Server.æœåŠ¡å™¨ç‰ˆæœ¬:     Apache Tomcat/@VERSION@

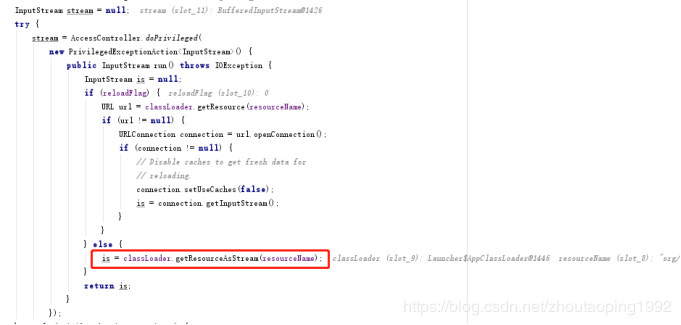
找到org.apache.catalina.startup.VersionLoggerListener类的log()方法，打断点一步一步跟踪



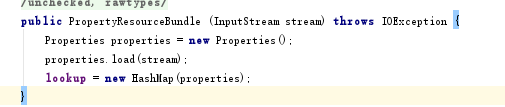
最终发现所有的值存在了PropertyResourceBundle类的lookup的map集合中，集合中的数据已经乱码了。



于是继续debugger查看lookup的加载，通过源码查看lookup集合中的数据是从properties文件中读取出来的。查看该properties文件编码也是UTF-8。于是继续查看源码。



ResourceBundle中的is = classLoader.getResourceAsStream(resourceName);加载的这个properties文件



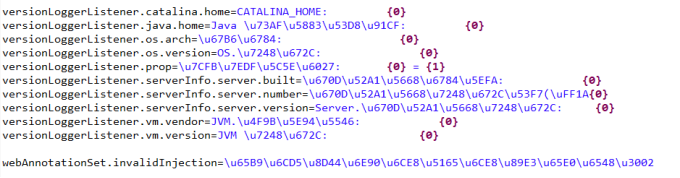
再通过PropertyResourceBundle构造方法加载的数据。

正准备修改这块代码时，发现这竟是JDK中的类，无法修改。（后来才知道ResourceBundle是用来做国际化的）。

后来查资料知道了：在java中, 读取文件的默认格式是iso8859-1, 而我们中文存储的时候一般是UTF-8. 所以导致读出来的是乱码。

解决方案有两种：

1、使用JDK下的工具native2ascii.exe将properties文件转为Unicode编码。转换后如图：



2、在代码中获取到了值之后手动重新编码解码下

        try {

            value = new String(value.getBytes("ISO-8859-1"), "UTF-8");

        }catch(Exception e){

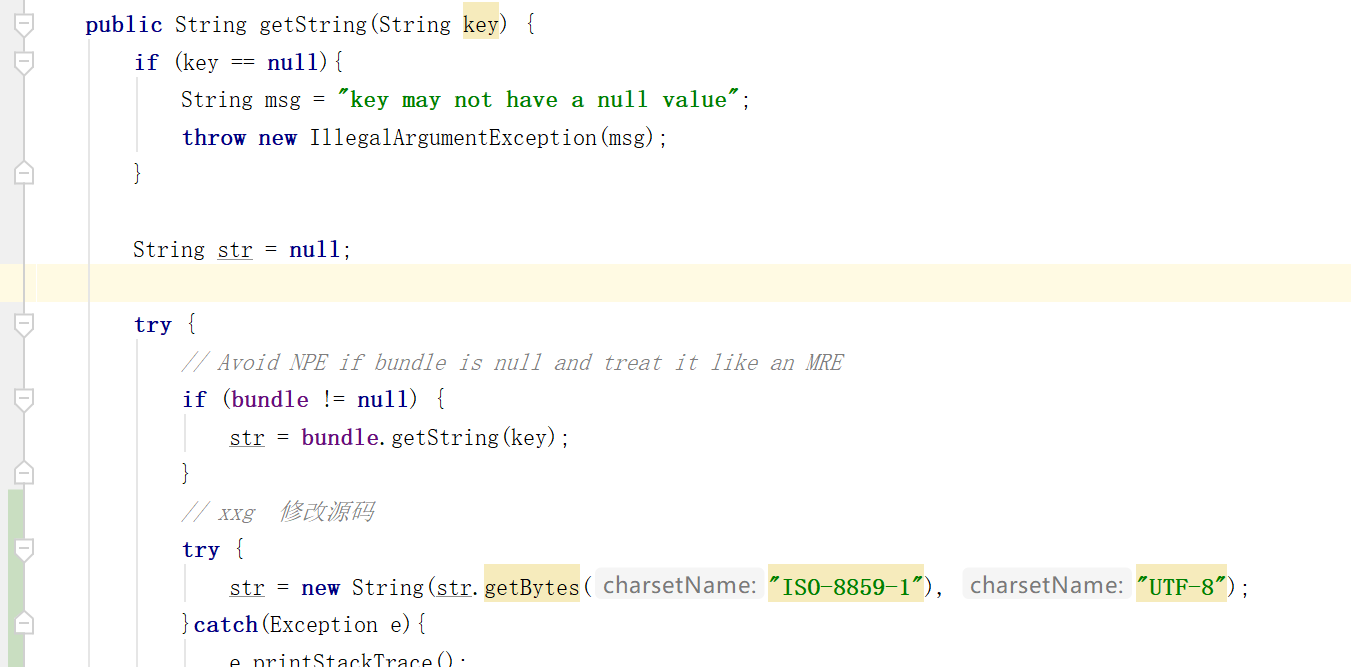
            e.printStackTrace();

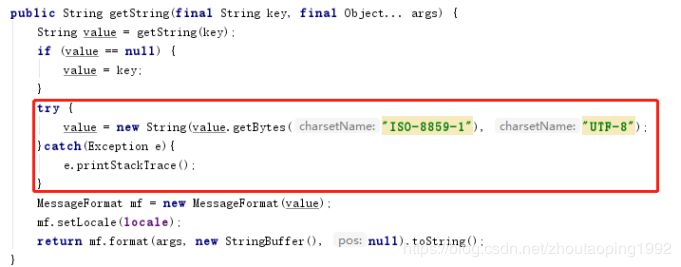
        }

经过测试，两种方法都可以解决问题。

因为tomcat中properties文件过多，我采用了第二种方法，修改了tomcat源码，修改如下：

1）org.apache.tomcat.util.res.StringManager类中的getString(final String key, final Object... args)方法。除了这个方法，其实还有一个方法也需要修改：

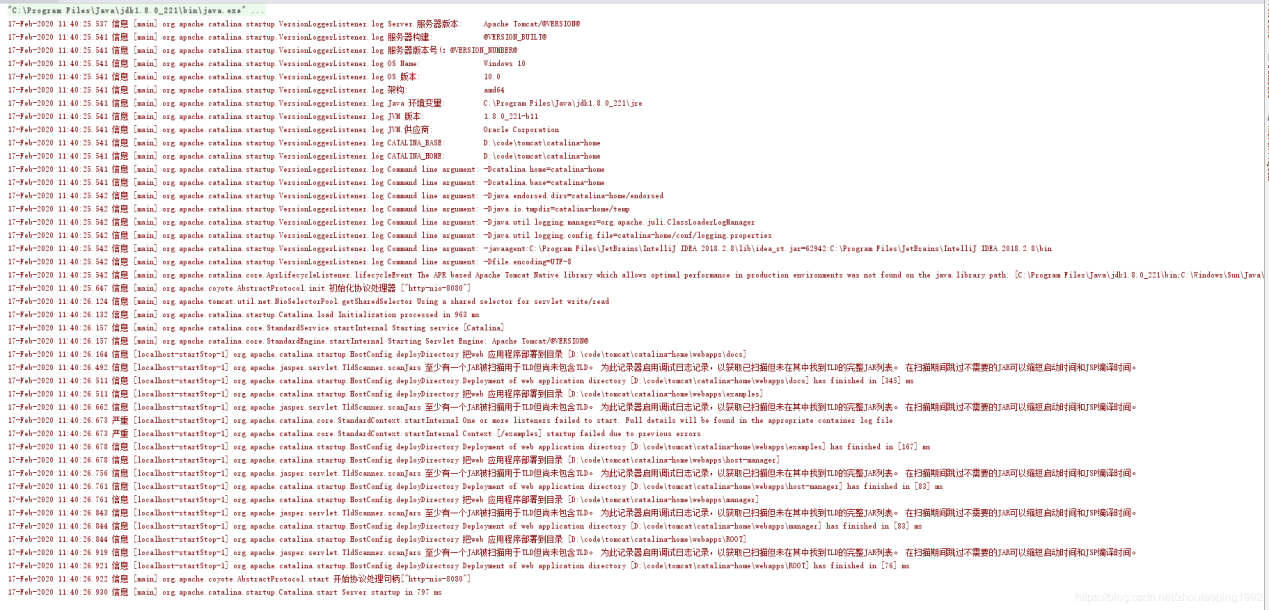




2）org.apache.jasper.compiler.Localizer类的getMessage(String errCode)方法



至此，乱码问题解决



————————————————

版权声明：本文为CSDN博主「zhoutaoping1992」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/zhoutaoping1992/java/article/details/104751705