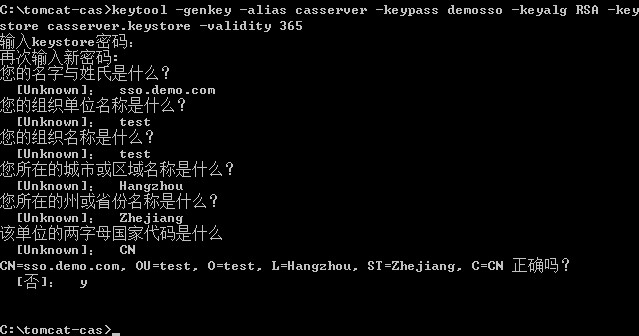
名词解释

单向认证：即CAS服务端采用HTTPS方式访问、客户端应用使用HTTP方式访问。

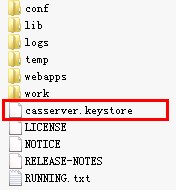
## 1.生成服务端密钥文件

登录服务器打开一个CMD窗口（开始菜单 -> 运行输入cmd后回车）并切换到tomcat安装目录下（如：c:\tomcat-cas），执行如下命令：

keytool -genkey -alias casserver -keypass demosso -keyalg RSA -keystore casserver.keystore -validity 365



执行后，可以看到tomcat安装目录下生成了一个casserver.keystore文件。如下图：



说明：-alias指定别名为casserver；-keyalg指定RSA算法；-keypass指定私钥密码；-keystore指定密钥文件名称为casserver.keystore；-validity指定有效期为365天。另外提示的输入keystore密码应与-keypass指定的相同；您的名字与姓氏是CAS服务器使用的域名（不能是IP，也不能是localhost），其它项随意填。

注意：服务器上如果有多个JDK，请确认环境变量中的JDK路径为tomcat所使用的JDK，如果不在环境变量中，也可切换到指定JDK的bin目录下执行命令；提示的输入keystore密码应与-keypass必须与指定的相同，否则后面tomcat启动会报IO异常（Cannot recover key）。

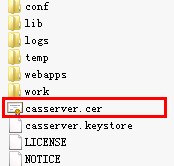
## 2.生成服务端证书

根据以上生成的服务端的密钥文件可以导出服务端证书，执行以下命令：

keytool -export -alias casserver -storepass demosso -file casserver.cer -keystore casserver.keystore

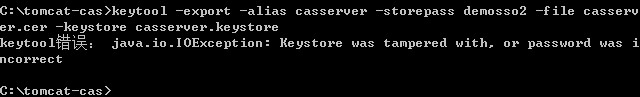


执行后，可以看到tomcat安装目录下生成了一个casserver.cer文件。如下图：



说明：-alias指定别名为casserver；-storepass指定私钥为demosso；-file指定导出证书的文件名为casserver.cer；-keystore指定之前生成的密钥文件的文件名。

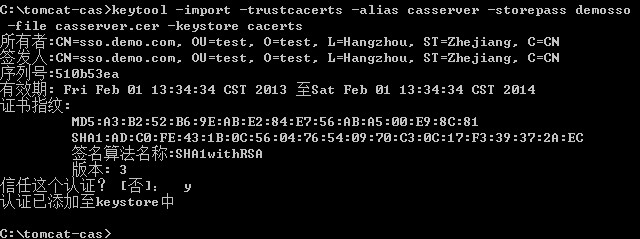
注意：-alias和-storepass必须为生成casserver.keystore密钥文件时所指定的别名和密码，否则证书导出失败，报如下错误：



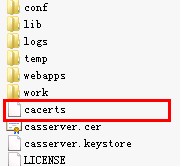
## 3.导入证书文件到cacerts 密钥库文件

导入以上生成的服务端的证书文件到一个cacerts密钥库文件，执行以下命令：

keytool -import -trustcacerts -alias casserver-storepass demosso -file casserver.cer–keystore cacerts



执行后，可以看到tomcat安装目录下生成了一个cacerts文件。如下图：

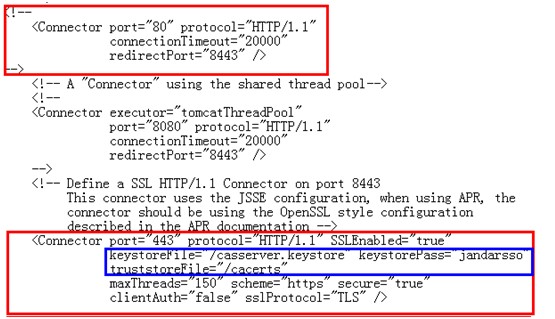


## 4.服务端Tomcat配置

在制作完成密钥文件、证书文件、密钥库文件后即可进行服务端Tomcat的配置。打开$CATALINA\_HOME/conf/server.xml文件，注释掉如下代码段：

<Connector port="80"protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443"/>

并取消注释<Connectorport="8443" protocol="HTTP/1.1" SSLEnabled="true"…/>代码段，修改后如下：

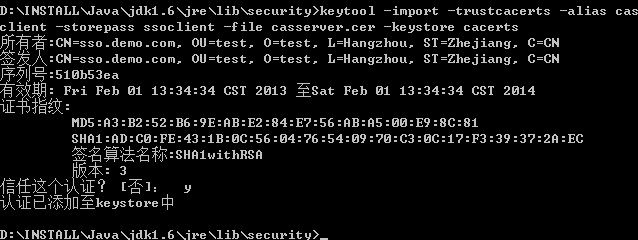


说明：port一般为8443或443，最常用的是443端口（https默认端口），这样https方式访问的时候可以不加端口号（如：https://sso.demo.com/cas/login）；keystoreFile为tomcat目录下的密钥文件；keystorePass为私钥密码；truststoreFile为生成的信任文件，如果此处不指定则默认为$JAVA\_HOME/jre/lib/security/cacerts文件；其它属性默认即可。

## 5.生成客户端密钥库文件

单向认证的客户端配置只需生成客户端信任文件caserts即可。首先将服务端生成的证书文件（之前生成的casserver.cer文件）复制到$JAVA\_HOME/jre/lib/security下，然后打开CMD窗口切换到$JAVA\_HOME/jre/lib/security下并执行命令：

keytool -import -trustcacerts -alias casclient-storepass ssoclient -file casserver.cer -keystore cacerts



执行后，可以看到$JAVA\_HOME/jre/lib/security目录下生成了一个cacerts文件。如下图：

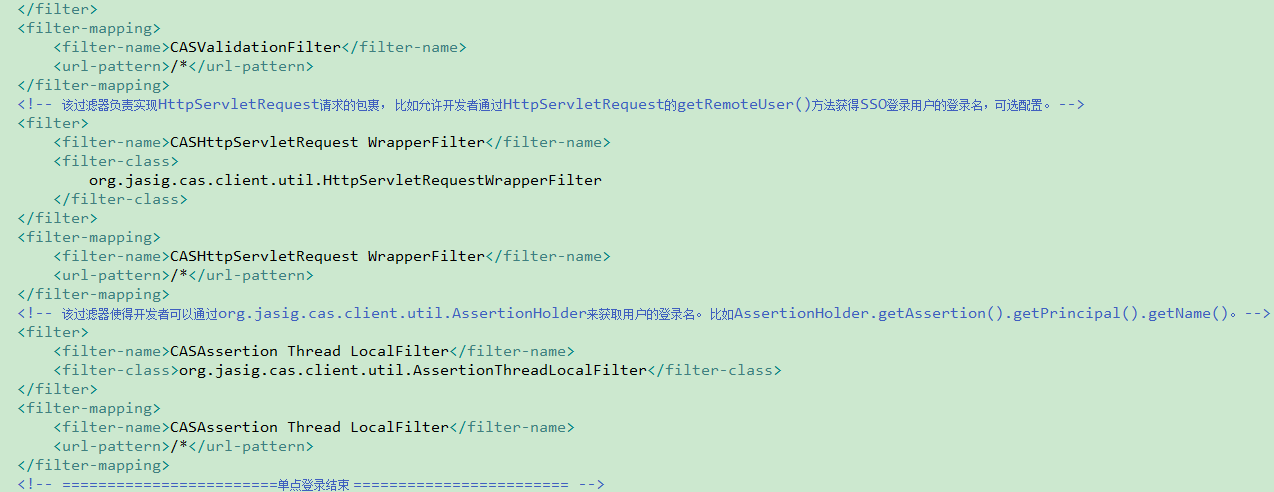


## 6.客户端应用配置

1、应用程序WEB-INF\lib目录下加入cas-client-core-3.2.1.jar包并添加到buildpath中；

2、编辑应用程序中的web.xml文件在最末端加入如下内容：





注意：

1.serverName属性值为客户端实际IP地址，可以为域名但绝不能为localhost

2.客户端程序单点退出功能，需要访问https://sso.demo.com/cas/logout

## 7.补充说明

上面讲的是CAS单独部署的情况，也就是与其它应用分开部署。但有些情况是CAS与其它应用部署到同一台机器同一个Tomcat（关键问题是使用同一个JDK），这种情况下，服务端Tomcat配置中的<Connector truststoreFile属性就要指定为jdk下的cacerts文件路径或者直接注释掉（默认会找$JAVAHOME\jre\lib\security\下的cacerts文件）。这样服务端和客户端是同一个，也就不必再根据服务端证书生成客户端密钥库文件了。

## 8.常见配置错误

1. CAS服务端Tomcat启动后报错：Error initializing endpoint java.io.IOException: Cannot recover key

是由于生成服务端密钥文件时所指定的keypass与提示输入的“keystore密码”不一致。

2.javax.servlet.ServletException: org.jasig.cas.client.validation.TicketValidationException:票根'ST-2-hozuLnLtIVGeaD5yju0Y-cas'不识别

一般是由于客户端应用web.xml中配置的serverName属性值为localhost或CAS服务端cas-servlet.xml配置文件中的<bean logoutController/>没加p:followServiceRedirects="true"退出后重定向属性。

3.SSLHandshakeException: java.security.cert.CertificateException: Nosubject alternative names present

是由于客户端应用web.xml配置中的casServerLoginUrl和casServerUrlPrefix两个URL属性的域名与证书中定义的不一致。