

Tomcat 4.x 인증서 설치 매뉴얼

Version 1.0

March 4, 2008

▶ 개인키 및 CSR생성

1. Tomcat은 Keytool을 통하여 개인키 및 CSR이 생성된다.
 개인키 및 인증서 신청 시 필요한 CSR을 생성하기 위하여 Keytool 명령어를 아래와 같이 실행한다.
keytool -genkey keyalg rsa -alias 앨리어스명 -keystore 키스토어명

```
C:\Program Files\Java\jdk1.6.0_01\bin>keytool -genkey -keyalg rsa -alias tomcat
-keystore crosscert
keystore 암호를 입력하십시오:
```

2. 1번 항목의 명령어를 입력하면 키스토어 패스워드 입력 후 DN값 입력 항목이 나오게 된다.

```
C:\Program Files\Java\jdk1.6.0_01\bin>keytool -genkey -keyalg rsa -alias tomcat
-keystore crosscert
keystore 암호를 입력하십시오:
새 암호를 다시 입력하십시오:
이름과 성을 입력하십시오.
[Unknown]: csteam.crosscert.com → 해당 URL 입력 (ex. www.crosscert.com)
조직 단위 이름을 입력하십시오.
[Unknown]: csteam → 부서명 입력 (ex. VCS Team)
조직 이름을 입력하십시오.
[Unknown]: Crosscert Inc. → 회사명 입력 (ex. Crosscert)
구/군/시 이름을 입력하십시오?
[Unknown]: Seocho-gu → 구/군/시 입력 (ex. Seocho-gu)
시/도 이름을 입력하십시오.
[Unknown]: Seoul → 시/도 입력 (ex. Seoul)
이 조직의 두 자리 국가 코드를 입력하십시오.
[Unknown]: KR → 국가코드 입력 (ex. KR)
CN=csteam.crosscert.com, OU=csteam, O=Crosscert Inc., L=Seocho-gu, ST=Seoul, C=KR
이<가> 맞습니까?
[아니오]: y

<tomcat>에 대한 키 암호를 입력하십시오.
<keystore 암호와 같은 경우 Enter를 누르십시오>:

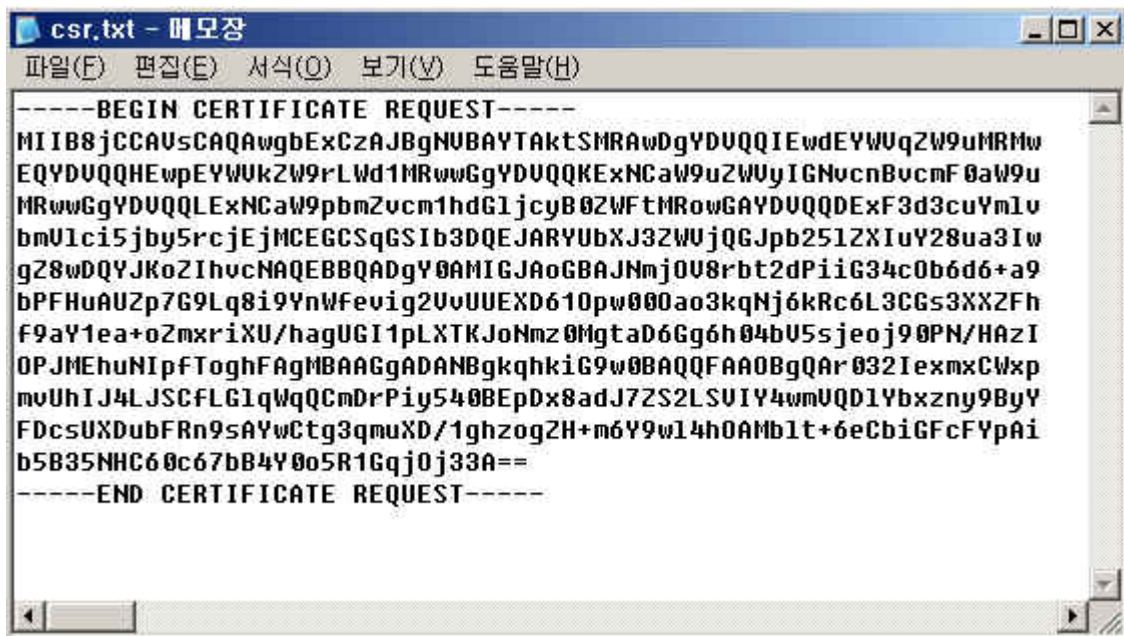
C:\Program Files\Java\jdk1.6.0_01\bin>
```

3. CSR을 파일형태로 생성하기 위해 아래의 명령어를 실행한다.

`keytool -certreq -alias 앨리어스명 -keyalg rsa -file csr.txt -keystore 키스토어명`

```
C:\Program Files\Java\jdk1.6.0_01\bin>keytool -certreq -alias tomcat -keyalg rsa
-file csr.txt -keystore crosscert
keystore 암호를 입력하십시오:
C:\Program Files\Java\jdk1.6.0_01\bin>
```

4. CSR생성 시 설정했던 저장위치로 이동하여 CSR파일(예 : csr.txt)을 메모장이나 워드패드로 열어보면 **-----BEGIN NEW CERTIFICATE REQUEST----- 부터 -----END NEW CERTIFICATE REQUEST-----** 를 확인할 수 있다. 인증서 신청 시 이 내용을 이용하여 인증서 신청을 진행한다.



▶ 인증서 설치

1. 인증서는 이메일을 통하여 전송되게 된다. 전달받은 인증서가 담긴 이메일의 내용 중 **-----BEGIN CERTIFICATE-----** 부터 **-----END CERTIFICATE-----** 까지를 복사하여 메모장이나 워드패드에 붙여 넣은 후 임의의 파일명(예 : cert.pem)으로 저장한다.

2. 발급 받은 인증서를 탑재하기 전에 먼저 체인인증서를 키스토어에 탑재하여야 한다.

`keytool -import -alias Intermediate -trustcacerts -file 체인인증서 파일명 -keystore 키스토어명`

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>cd C:\Program Files\Apache Software Foundation\Tomcat 5.5\ssl\test

C:\Program Files\Apache Software Foundation\Tomcat 5.5\ssl\test>keytool -import -alias Intermediate -trustcacerts -file Intermediate.cer -keystore crosscert.keystore 암호를 입력하십시오: 123456
인증이 keystore에 추가되었습니다.

C:\Program Files\Apache Software Foundation\Tomcat 5.5\ssl\test>_
    
```

3. 발급 받은 인증서를 키스토어에 탑재한다.

`keytool -import -alias 엘리어스명 -trustcacerts -file 인증서 파일명 -keystore 키스토어명`

```

C:\WINDOWS\system32\cmd.exe - keytool -import -alias crosscert -trustcacerts -f...
(C) Copyright 1985-2003 Microsoft Corp.

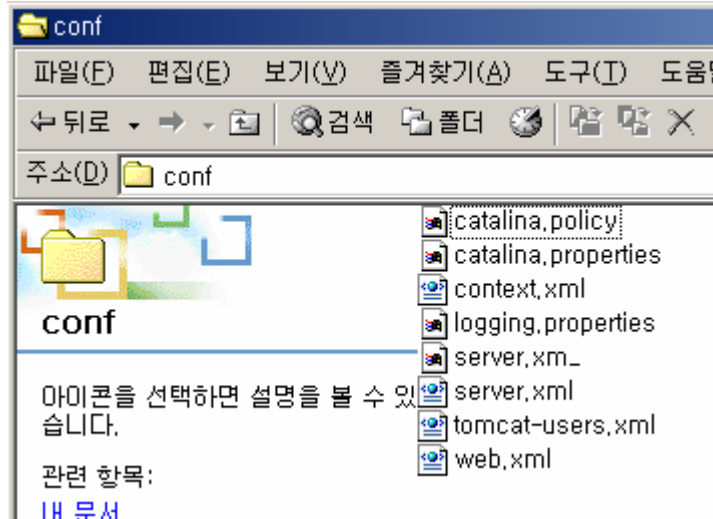
C:\Documents and Settings\Administrator>cd C:\Program Files\Apache Software Foundation\Tomcat 5.5\ssl\test

C:\Program Files\Apache Software Foundation\Tomcat 5.5\ssl\test>keytool -import -alias Intermediate -trustcacerts -file Intermediate.cer -keystore crosscert.keystore 암호를 입력하십시오: 123456
인증이 keystore에 추가되었습니다.

C:\Program Files\Apache Software Foundation\Tomcat 5.5\ssl\test>keytool -import -alias crosscert -trustcacerts -file cert.pem -keystore crosscert.keystore 암호를 입력하십시오: 123456
수신자: CN=www.sandvil.com, OU=Member, VeriSign Trust Network, OU=Authenticated by KEC&, Inc.", OU=Terms of use at www.crosscert.com/rpa (c) 04, OU=WEB3, O=Ganevil, L=Guro, ST=Seoul, C=KR
발급자: CN=VeriSign Class 3 Secure Server CA, OU=Terms of use at https://www.verisign.com/rpa (c)05, OU=VeriSign Trust Network, O="VeriSign, Inc.", C=US
일련 번호: 332bf719f8a043bb538e6e400ea63046
개시일: Thu Nov 06 09:00:00 KST 2008 만료일: Sat Nov 07 08:59:59 KST 2009
인증서 지문:
MD5: 3E:07:5C:98:36:1E:94:6C:82:8F:10:72:61:BC:0B:30
SHA1: B9:1B:9B:97:BA:93:97:99:DA:A7:83:8D:30:2F:68:C6:F0:18:15:67
이 인증서를 신뢰하십니까? [아니오]: _
    
```

▶ SSL 적용

1. Tomcat 4.x의 환경설정 디렉토리로 이동한다.



2. server.xml파일을 메모장과 같은 프로그램을 통해 엽니다.

```

server.xml - 메모장
파일(F) 편집(E) 서식(O) 도움말(H)
<!-- Note: A "Server" is not itself a "Container", so you may not
      define subcomponents such as "Values" at this level.
      Documentation at /docs/config/server.html
-->
<Server port="8005" shutdown="SHUTDOWN">

  <!-- APR library loader. Documentation at /docs/apr.html -->
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <!-- Initialize Jasper prior to webapps are loaded. Documentation at /docs/jasper-howto.html -->
  <Listener className="org.apache.catalina.core.JasperListener" />
  <!-- JMX Support for the Tomcat server. Documentation at /docs/non-existent.html -->
  <Listener className="org.apache.catalina.mbeans.ServerLifecycleListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />

  <!-- Global JNDI resources
      Documentation at /docs/jndi-resources-howto.html
  -->
  <GlobalNamingResources>
    <!-- Editable user database that can also be used by
          UserDatabaseRealm to authenticate users
    -->
    <Resource name="UserDatabase" auth="Container"
              type="org.apache.catalina.UserDatabase"
              description="User database that can be updated and saved"
              factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
              pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>

  <!-- A "Service" is a collection of one or more "Connectors" that share
        a single "Container" Note: A "Service" is not itself a "Container",
        so you may not define subcomponents such as "Values" at this level.
        Documentation at /docs/config/service.html
  -->
  <Service name="Catalina">

    <!--The connectors can use a shared executor, you can define one or more named thread pools-->
    <!--
    <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
              maxThreads="150" minSpareThreads="4"/>
    -->
    
```

3. Tomcat 4.x의 경우 server.xml파일에서 아래와 같이 설정해 주어야 한다. (SSL의 기본포트 : 443)

```
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
  port="SSL에 사용할 포트" enableLookups="true" scheme="https" secure="true"
  acceptCount="100"
  useURIValidationHack="flase" disableUploadTimeout="true"
  keystoreFile="키스토어 경로"
  keystorePass="키스토어 패스워드"
  clientAuth="false" sslProtocol="TLS" />
```

<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->

```
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
  port="8443" enableLookups="true" scheme="https" secure="true"
  acceptCount="100"
  useURIValidationHack="False" disableUploadTimeout="true"
  keystoreFile="C:\Program Files\Apache Software Foundation\Tomcat 4.1\key\crosscert"
  keystorePass="123456"
  clientAuth="false" sslProtocol="TLS" />
```

4. https로 접속하여 인증서가 정상적으로 탑재되었는지 확인한다.

