Installation - Java

1. Using JDK 1.4.0

After SUN released the Java (TM) 2 Platform Standard Edition v1.4.0 , the xml-security package stopped working. This is a known problem: SUN packaged a beta of Xalan into the JDK 1.4.0, but the xml-security package requires a stable version of Xalan (v2.2.0 or later). To fix the problem, you have to put the xalan.jar into a special directory in your JDK: `j2sdk1.4.0/jre/lib/endorsed/xalan.jar` . If you installed an out-of-the-box JDK1.4 (e.g. on Windows 2000), the "endorsed" directory does not exist: you'll have to create it by hand.

Warning:
Putting this JAR to another location like `lib/ext` WILL NOT WORK.

For more on that, you can also check the Unofficial JAXP FAQ .

2. Prerequisites

Make sure you get the Jakarta Ant Tool from http://jakarta.apache.org/ant/

3. Getting the source

You can download the sources via WWW from the download page at http://xml.apache.org/security/download.html

This project's CVS repository can be checked out through anonymous (pserver) CVS with the following instruction set. The module you wish to check out must be specified as the modulename. When prompted for a password for anonymous, simply enter "anoncvs" without quotes:

```
cvs -d :pserver:anoncvs@cvs.apache.org:/home/cvspublic login
        password: anoncvs
        cvs -d :pserver:anoncvs@cvs.apache.org:/home/cvspublic checkout
        xml-security
```

A HTTP interface to browse the sources online is available via

.cgi/xml-security/

4. Compiling the source

At the command prompt type `ant test`. If you want to use jikes instead of your default java compiler locate the `build.xml` file and replace the line

```
<property name="build.compiler" value="classic"/>
```

with

```
<property name="build.compiler" value="jikes"/>
```

5. Testing the distibution

The first way to ensure that everything is in place is to run the unit tests. This is simply done by typing `ant test`. This starts the included JUnit test cases. Actually, we do not have complete test coverage, but as a first start, it works.

6. Playing around with the examples

To see how the distribution works, simply run `ant mega-sample` to let ant execute several examples from the `src_samples/` directory.

7. Files in the source package release

build.xml	Top level Ant Makefile -- read README file before building
LICENSE.txt	License for the software
README	Build instructions
Readme.html	Web page redirect required for building documentation
STATUS	Current source code status information
data/	Directory containing sample data files and test vectors for the unit tests
doc/xml/	Directory containing documentation, in XML form
src/	Directory containing source code for the core library
src_samples/	Directory containing source code for samples

src_unitTests/	Directory containing source code for unit tests

8. Where is my JCE?

There is *no* JCE bundled together with this distribution. Living in Germany, I had no problem to include the JCE in this software package but then I realized that the Apache Project is hosted in the US where some export restrictions apply to the cryptographic primitives.

Well, how do we solve this problem? The nice guys from the Bouncy Castle where so helpful to supply the JAR that you need to create HMAC integrity checks on their web site. When you use the ant makefile `build.xml` and simply say `ant compile` or `ant get-jce`, `ant` tries to fetch this JAR from the australian server. After that step, the compilation works completely.

The ant make tools initiates an automated download of the BouncyCastle JCE. The file is downloaded into the libs/ directory and a "bc-" is prepended to the filename. This is done because we want the provider name (bc means BouncyCastle) being visible in the JAR's filename.

If you are a little paranoid like all security people and don't want ant to make automated downloads or your firewall doesn't permit it (preventing programs "phoning home"), look in the `./build.xml` file for the properties called

```
        <property name="jce.download.file" value="jce-jdk13-114.jar />
        <property name="jce.download"
                  value="http://www.bouncycastle.org/download/${jce.download.file}" />
        <property name="lib.jce" value="${libs}/bc-${jce.download.file}" />
```

Here you can see that the file jce-jdk13-114.jar is downloaded and stored in the location `./libs/bc-jce-jdk13-114.jar`

If you do this by hand (pointing you favourite web browser and download it yourself), simply put a "`bc-`" in front of the filename and put it to `./libs/`, then ant won't try to make a download.