Installing Java

Various things need to be added to a standard JDK (or JRE) install for all our web apps to work correctly.

JAI

If Java Advanced Imaging version 1.1.2 update 1 or later is not installed then the FOP processor (for starters) will fail.

Aladdin Certificate

Anything that talks to the Account System daemon on aladdin will fail unless aladdin's certificate is installed. The certificate may currently be found on aladdin at `/opt/bin/TempAD/aladdin.cert` (here's how to make a new one) and is installed as follows:

1. Determine the machine name as {machine-name} (for example `aladdin`)
2. Determine the path to the certificate as {certificate-path}
3. Locate the root certificate file in the Java install of interest. This will probably be `{JAVA_HOME}/jre/lib/security/cacerts` - call this {cacerts-path}
4. You should change the password of the cacerts keystore to be the same as the root password for the machine (will do for the time being) using `keytool -storepasswd -keystore {cacerts-path}` - the initial password is `changeit`
5. You can list the contents of the cacerts file with `keytool -list -keystore {cacerts-path}` which will prompt for the password. Note that the certificates don't seem to be listed in any particular order
6. Import the certificate with `keytool -import -file {certificate-path} -alias {machine-name} -keystore {cacerts-path}` giving the password when requested and entering yes to accept the certificate

And if this doesn't work?

If you're looking to use a java keystore and some helpful chap has created a private key with openssl and supplied you with public.crt and private.key you're going to have problems. Many, many problems. Panic not; help is at hand.

1. Make sure openssl is installed, and you're using it. Be warned that Sun's implementation is installed on our machines. Use the correct version.
2. Create a PKCS12 keystore from the files you have been handed (cabundle.crt can be found on sole or cotter in /opt/www/apache2/conf/ssl.crt/):

```
/opt/openssl/bin/openssl pkcs12 -export -chain -in $public.crt -inkey
$private.key -out $keystorename -CAfile $cabundle.crt -name $aliasofcert
```

And you're done. If you want to check inside your store, use:

```
keytool -list -keystore $keystorename -storetype pkcs12
```

Note you need to define the storetype for this keystore. This is NOT the default type for a keystore so you probably want to convert it to a java keystore which is more useful.

3. Get the Jetty jar from somewhere e.g. the webdev-common/libs directory in subversion

4. Convert from a PKCS12 keystore to a java keystore:

```
java -classpath jetty-6.1.2rc3.jar org.mortbay.jetty.security.
PKCS12Import $oldkeystorename $newkeystorename
```

5. Use your newly made keystore:

For example, a tomcat SSL-enabled connector:

```
<Connector port="8443"
   maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
   enableLookups="false" acceptCount="100" keyAlias="tomcat"
keystoreFile="/opt/www/tomcatkeys"
   disableUploadTimeout="true" scheme="https" secure="true" clientAuth="
false" sslProtocol="TLS" />
```