Delegation Issuing Service (DIS) Installation Guide

Document History

Version	Date	Comments
0.1	9 September 2005	First draft by Wensheng
0.2	12 September 2005	Updated by Tuan Anh
0.3	14 September 2005	Reviewed by David
1.0	15 September 2005	First public release
1.1	1 July 2006	Updated with Acceptance Tests for issuing attributes
1.2	24 July 2006	Update the installation instructions
1.3	26 July 2006	Updated with Acceptance Tests for revocation attributes
1.4	14 August 2006	Fix some inconsistencies after validating by Romain
1.5	3 October 2006	Add the SearchRequestor parameter, change the parameters in the configuration file using Permis standard parameters.
1.6	24 October 2006	Add instruction for using policy stored in a file
1.7	24 October 2006	Add support for java 1.5
1.8	22 February 2007	Add support for Tomat 5.5
1.9	23 April 2007	Add support for Apache 2.2
2.0	12 August 2008	Using PERMIS v5 policies and Apache http server on Windows (by Linying)
2.1	25 March 2009	Updated the authentication of tomcat, DIS installation instruction, instruction for apache2.2 (by Kaniz)
2.2	17 February 2010	Revised and updated by Kaniz

2.3	16 April 2010	Updated by Mark
2.5	26 October 2010	Install guide rewritten and separated into its own document
2.5.2	2 November 2010	Java installation explained more fully. Tomcat native library installation corrected. Default DIS configuration file inserted.
2.5.4	5 November 2010	Addressed feedback from Kaniz
2.5.5	26 May 2011	Added alternative <FilesMatch> for PHP files. Updated screenshots for DIS Web UI. Refreshed formatting. Added table of contents. Added missing information from old document version.
2.5.6	08 July 2011	Revised and updated software versions used: Firefox 5.0, Tomcat 5.5.33, JDK 6u26, APR 1.4.5, openLDAP 2.4.26, BerkeleyDB 5.2.28, Axis2-1.5.5. DIS server installation guide separated into its own document.
2.5.7	23 August 2011	Addressed received feedback. Updated to Tomcat 7.

Contents

Introduction

The Delegation Issuing Service (DIS) is a web service for issuing attribute certificates on behalf of privilege holders who wish to delegate their privileges to their peers and subordinates. The DIS web service is accessed by DIS web service client applications. A DIS client may be an application (e.g. when an application wants to delegate privileges to another application) or may be directly accessed by a human user via an appropriate user interface, or may be a trusted server acting as a proxy between human users and the DIS (e.g. the PHP client that we provide in the PERMIS website).

In our implementation of the DIS, the DIS web service server is a Java component, which is based on the Tomcat application server and the Apache AXIS2 soap engine, and it can be invoked through SOAP calls.

This document describes the steps necessary for deploying the DIS Web Service based on the package distributed in the PERMIS web site.

System Requirements

- A recent Java Runtime Environment (JRE). JRE 5 or above is recommended and is available from http://java.sun.com/
- A recent Java Servlet Container. Tomcat 7 is recommended and is available from http://tomcat.apache.org/
- A SOAP engine. Apache Axis2 is recommended and is available from http://ws.apache.org/axis2/
- A LDAP server that the DIS can write to. OpenLDAP 2.4 is recommended and is available from http://www.openldap.org/

After installing the software, users should go through the DIS Acceptance Tests document to confirm that their installation of the Delegation Issuing Service is working as expected. It is paramount that the tests are executed in the order described in the document. A publicly available demo of the Delegation Issuing Service is available at https://sec.cs.kent.ac.uk/dis.html. Trying out this demo will give users a better understanding of what they are trying to build.

Software Contents

The software is available from http://sec.cs.kent.ac.uk/permis/downloads/Level3/DIS.shtml.

Inside `disService_5_1_3.zip` there is a directory called `serverSide`, which contains the following files:

- **dis_5_1_3.aar**

  This file is an archive following the AXIS2 web service package format, which contains the Java classes for the DIS Web Service. The archive also contains the DIS configuration file, `issrg/dis/dis.cfg`, which needs to be modified before deploying the web service.

- **dis-policy.xml**

  An example DIS policy.

- **log4j.config**

  An example log4j configuration file. This is used for debugging purposes.

- **data.ldif**

  Example LDAP data. It contains users and role attribute certificates. This is used in the example installation below.

- **disac.ace**

  This is the attribute certificate (AC) of the DIS. It is already included in the `data.ldif` file. The AC will be used by DIS to sign the delegation ACs.

The `serverSide` directory also contains a subdirectory called `disServerKeystore`, which contains the following files, used in the example installation below:

- **dis-cert.pem**

  The DIS server certificate. Used for enabling SSL.

- **dis-cert.key**

  The DIS server private key. Used for enabling SSL.

- **cacert.pem**

  The CA certificate. Validates dis-cert and soa-cert.

- **dis-cert.p12**

  The DIS private key and public key certificate. Used by the DIS to sign attribute certificates. The file's password is "dis123".

- **soa-cert.p12**

  The SOA private key and public key certificate. It will be used to sign the DIS's AC. The file's password is "l3tM3InNow".

Example Installation of the DIS Web Service

This example installation is based on a fresh default installation of Ubuntu 11.04 (Natty Narwhal) and has been tested with the following software versions:

- JDK 6 Update 26
- Tomcat 7.0.19
- Apache Portable Runtime 1.4.5
- Apache Axis2 1.5.5 (**Later versions are *not* currently compatible with the DIS!**)
- OpenLDAP 2.4.28 using Berkeley DB 5.2.28

A general knowledge of UNIX commands is required to follow this example installation.

Step 0: Create a directory for holding the installation.

To keep things simple, we will install everything to a single directory in your home directory. Let's start by making this directory and copying the server-side DIS files into it. Execute the following commands:

```
1. mkdir ~/dis
2. cp disService_5_1_3.zip ~/dis
3. unzip disService_5_1_3.zip
```

You now have a `~/dis/serverSide` directory with the DIS Web Service package.

Step 1: Install JDK

The JDK is required for running Tomcat, compiling Tomcat Native and running the DIS Web Service.

Download Linux version of JDK 6 from http://java.sun.com/ to `~/dis`. Execute the following commands in the directory:

```
1. chmod a+x jdk-6u26-linux-i586.bin
2. ./jdk-6u26-linux-i586.bin
```

This installs JDK to `~/dis`. We now need to set the `JAVA_HOME` environment variable to the path of the JDK installation. Append the following to `.bashrc` in your home directory:

```
export JAVA_HOME=~/dis/jdk1.6.0_26
```

Also add `JAVA_HOME/bin` to your path, by adding the following beneath the above line:

```
export PATH=$JAVA_HOME/bin:$PATH
```

Step 2: Install Tomcat

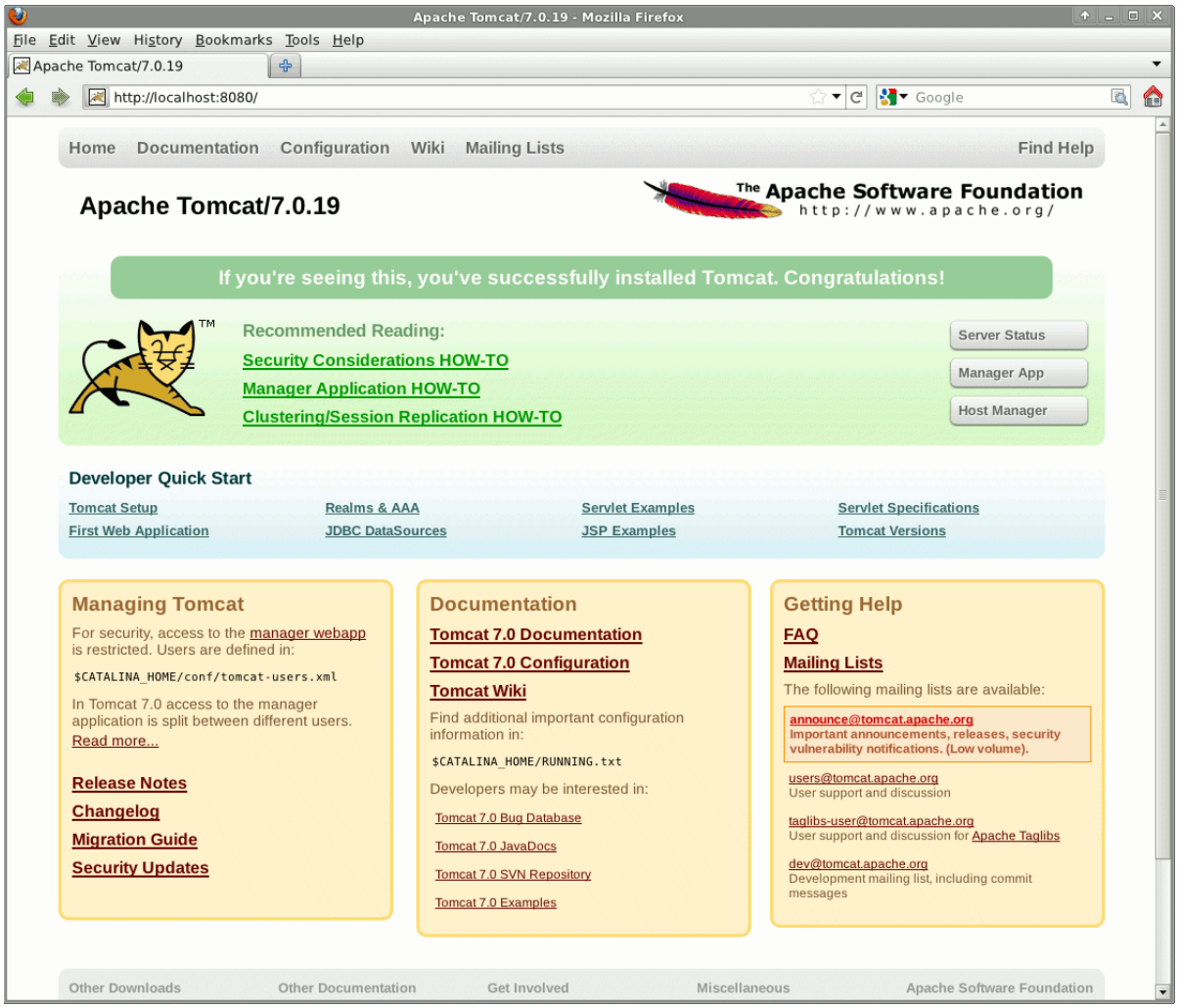
Tomcat is required for hosting Apache Axis2, which in turn is required for hosting the DIS Web Service.

Download the core Tomcat 7 binary from http://tomcat.apache.org/ to `~/dis`. Extract the downloaded file to the same location. This path will be referred to as `$CATALINA_HOME` from now on.

We will now test Tomcat works. Start Tomcat by executing:

```
$CATALINA_HOME/bin/startup.sh
```

Go to http://localhost:8080 in a web browser. The default Tomcat home page should be displayed.



Stop Tomcat by executing:

```
$CATALINA_HOME/bin/shutdown.sh
```

Step 3: Enable SSL mutual authentication on Tomcat

SSL mutual authentication is required by the DIS Web Service so it knows delegation requests are genuine, i.e. not spoofed.

We require Apache Portable Runtime for enabling Tomcat's native SSL. Download Apache Portable Runtime from http://apr.apache.org/ to `~/dis`. Extract the downloaded file to the same location. Execute the following commands there:

```
1. ./configure --prefix=/home/user/dis/apr
2. make
3. make install
```

Apache Portable Runtime should have now been installed to `~/dis/apr`.

We will now install the Tomcat native libraries. Execute the following commands:

```
1. sudo apt-get install libssl-dev
2. cd $CATALINA_HOME/bin
3. tar xvf tomcat-native.tar.gz
4. cd tomcat-native-1.1.20-src
5. cd jni/native
6. ./configure --with-apr=/home/user/dis/apr --with-ssl
7. make
8. sudo make install
```

The libraries should now be installed to `/usr/local/apr/lib`. This needs to be your Java library path so Tomcat can find Apache Portable Runtime.

Set your `CATALINA_OPTS` environment variable to:

```
"-Djava.library.path=/usr/local/apr/lib"
```

This can be done by editing the $CATALINA_HOME/bin/catalina.sh file. Adding the following line to the beginning of the catalina.sh file (right after the comments with the list of environment variables):

```
CATALINA_OPTS="-Djava.library.path=/usr/local/apr/lib"
```

Now, we must enable server authentication.

In `$CATALINA_HOME/conf/server.xml` look for

```
<Listener className="org.apache.catalina.core.AprLifecycleListener" />
```

and check whether SSLEngine is on, if not, change it to

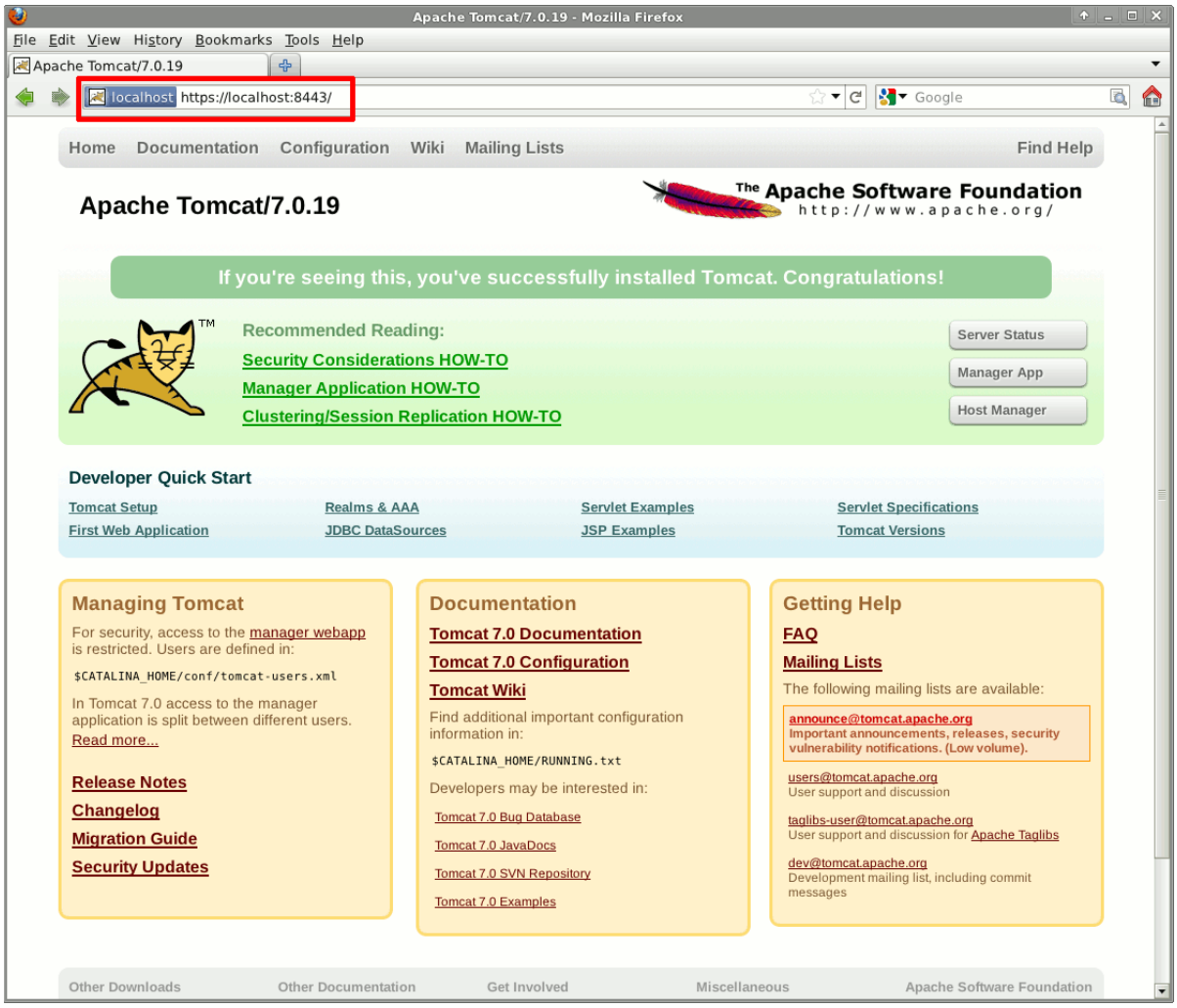
```
<Listener className="org.apache.catalina.core.AprLifecycleListener"
SSLEngine="on" />
```

In the same file ($CATALINA_HOME/conf/server.xml), add to the content of the `<Service name="Catalina">` element:

```
<Connector protocol="org.apache.coyote.http11.Http11AprProtocol"
           port="8443" maxThreads="150"
           scheme="https" secure="true"
           SSLEnabled="true"
           SSLCertificateKeyFile="/home/user/dis/serverSide/disServerKeystore/dis-cert.key"
           SSLCertificateFile="/home/user/dis/serverSide/disServerKeystore/dis-cert.pem" />
```

The `SSLCertificateKeyFile` and `SSLCertificateFile` attributes must be adjusted accordingly. This enables server authentication.

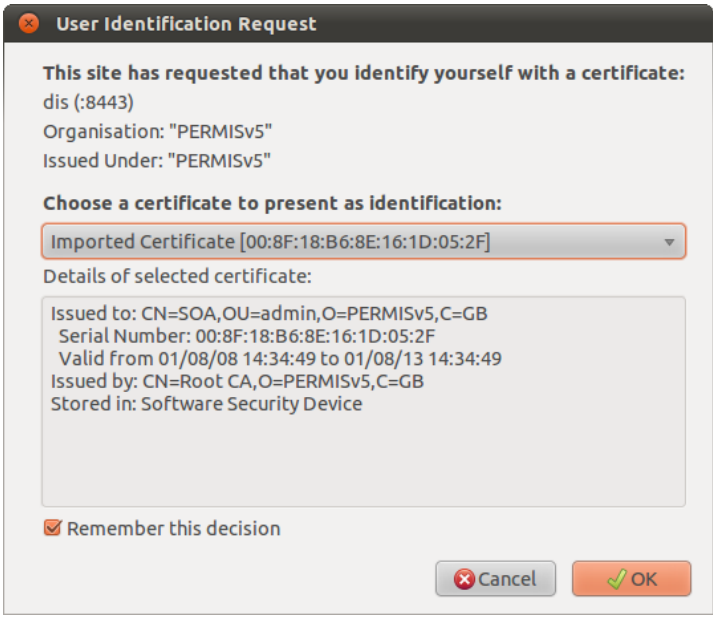
We will now test server authentication. Restart Tomcat and go to https://localhost:8443 in a web browser. After accepting the server's certificate, the default Tomcat home page should be displayed.



To enable client authentication, add the following attributes to the previously added `<Connector>` element (in the `$CATALINA_HOME/conf/server.xml file`), adjusting the `SSLCACertificateFile` accordingly:

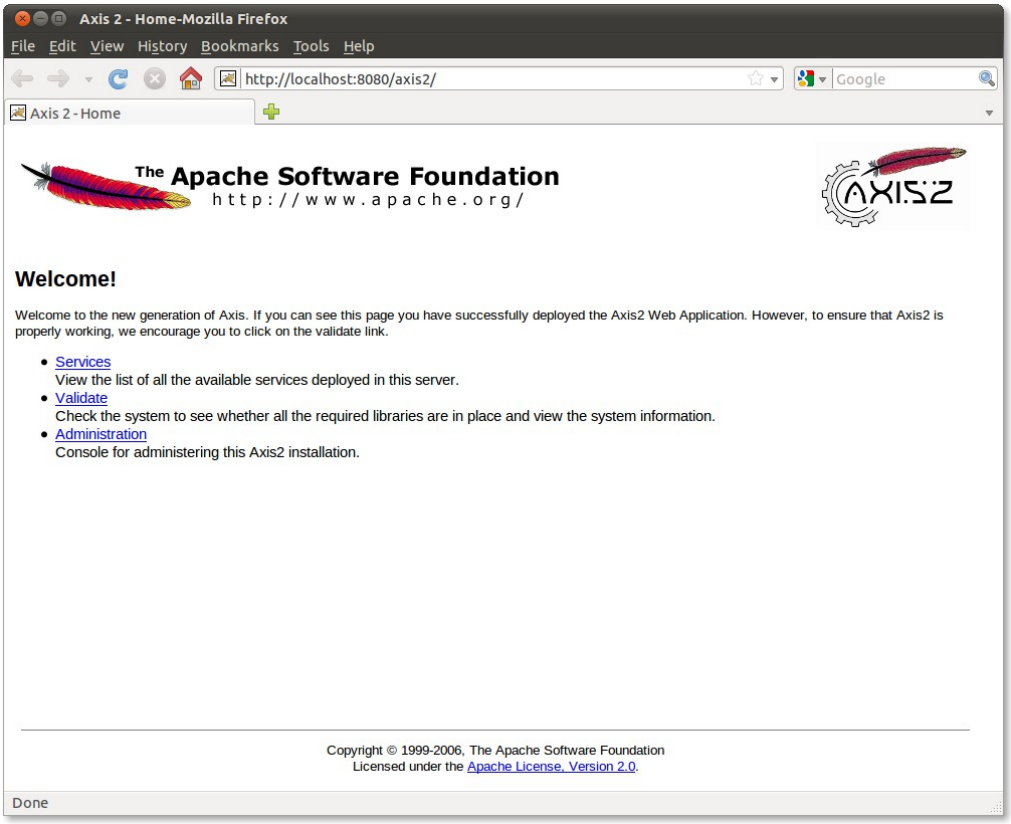
```
SSLVerifyClient="require"
SSLCACertificateFile="/home/user/dis/serverSide/disServerKeystore/cacert.pem"
```

We will now test client authentication. Import `soa-cert.p12` into your web browser. (For Firefox 5.0, Preferences > Advanced > Encryption > View Certificates > Your Certificates > Import > Open soa-cert.p12 > Enter “l3tM3InNow”.) Restart Tomcat and go to https://localhost:8443 in your web browser. After presenting the SOA certificate, the default Tomcat home page should be displayed again.

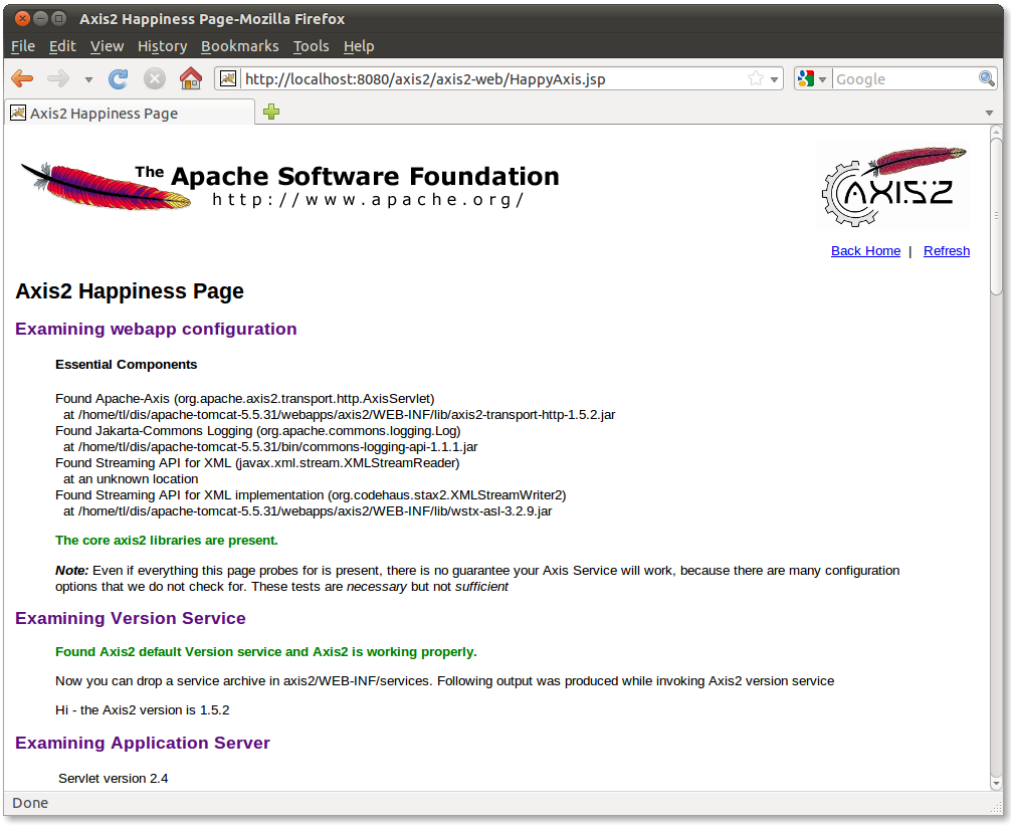


Step 4: Install Apache Axis2

Download the Axis2 1.5.5 WAR Distribution from http://ws.apache.org/axis2/. Copy `axis2.war` from the downloaded file to `$CATALINA_HOME/webapps`. Go to http://localhost:8080/axis2 in a web browser. The Axis2 welcome page should be displayed.



Click the “Validate” link. “The core axis2 libraries are present.” and “Found Axis2 default Version service and Axis2 is working properly.” should be displayed.



Step 5: Activate SSL connections in Axis2

We need to configure Axis2 to accept SSL connections, as explained by its documentation.

Edit the axis2 configuration file (axis2.xml), located at `/home/user/dis/apache-tomcat-5.5.33/webapps/axis2/WEB-INF/conf/axis2.xml`.

Locate the element

```
<transportReceiver name="http"
            class="org.apache.axis2.transport.http.AxisServletListener"/>
```

and replace with

```
<transportReceiver name="http"
         class="org.apache.axis2.transport.http.AxisServletListener">
    <parameter name="port">8080</parameter>
</transportReceiver>
<transportReceiver name="https"
         class="org.apache.axis2.transport.http.AxisServletListener">
    <parameter name="port">8443</parameter>
</transportReceiver>
```

Restart tomcat.

Step 6: Configure the DIS Web Service

Edit the DIS configuration file `dis.cfg` in `dis_x_x_x.aar` accordingly. This file must be extracted from the `dis_x_x_x.aar` archive, and edited accordingly.

Extract the contents of the DIS Web Service package.

```
unzip dis_x_x_x.aar -d dis
```

Edit the `~/dis/serverSide/dis/issrg/dis/dis.cfg` file. The default configuration file is shown below. Lines that start with a '#' are comments, and are ignored by the DIS.

```
#PolicyIssuer cn=SOA,ou=Admin,o=PERMISv5, c=gb
#PolicyIdentifier 1.2.826.0.1.3344810.6.0.0.n
```

```
#PolicyLocation ldap://localhost/c=gb
PolicyLocation file://home/user/dis/serverSide/dis-policy.xml
#PolicyLocationUsername cn=Manager,c=gb
#PolicyLocationPW secret
RootPKC /home/user/dis/serverSide/disServerKeystore/cacert.cer
CredentialLocation ldap://localhost/c=gb
CredentialLocationUsername cn=Manager,c=gb
CredentialLocationPW secret
downgradeable 1
#SigningKeyFile will be used by DIS to sign the certificate
SigningKeyFile /home/user/dis/serverSide/disServerKeystore/dis-cert.p12
SigningKeyPW dis123
LDAP_AC_Attribute attributeCertificateAttribute
LDAP_PKC_Attribute userCertificate;binary
#There are 6 debug levels:trace, debug, info, warn, error and fatal
#DebugLevel debug
log4jlocation /home/user/dis/serverSide/log4j.config
#TrustedProxy will contain the trusted proxies. It can have zero or more
trusted proxies.
TrustedProxy cn=httpd,ou=admin,o=PERMISv5,c=GB
TrustedProxy CN=SOA,OU=admin,O=PERMISv5,C=GB
#SearchRequestor indicates who can search and view the ACs.
#There are two options for this parameter: anyone, revokers
SearchRequestor anyone
```

Each parameter is defined below. For this example installation `PolicyLocation`, `RootPKC`, `SigningKeyFile`, and `log4jlocation` are the only parameters that need to be adjusted. Your `log4j.config` file's `log4j.appender.A1.File` parameter should be adjusted too.

- **PolicyIssuer**

  The issuer of the DIS's policy. This parameter is not required for XML policy files.

- **PolicyIdentifier**

  The identifier of the DIS's policy. This parameter is not required for XML policy files. The policy ID (PolicyIdentifier) is 1.2.826.0.1.3344810.6.0.0.*n*, where *n* can be anything you want. In this demonstration package, *n* is 1. If you do not have access to your own object identifier branch, then you can request us to issue you one under our branch 1.2.826.0.1.3344810.6. Alternatively you can now use any URN as the policy object ID.

- **PolicyLocation**

  The location of the DIS's policy. If the policy is stored in an LDAP server, this parameter's value should be the server's URL. If the policy is stored in a file, the

value should be the file's URL in the (non-standard) format "`file://path/to/policy.ext`". Currently, two types of policy files are supported, which are X.509 AC policy files (extension "ace") and XML policy files (extension "xml").

- **PolicyLocationUsername**

  The user of the LDAP server to connect as. Read access is sufficient. This parameter is not required for file policies.

- **PolicyLocationPW**

  The password of the above user. This parameter is not required for file policies.

- **RootPKC**

  The path of the root certificate.

- **CredentialLocation**

  The URL of the LDAP server that will store the delegated role attribute certificates.

- **CredentialLocationUsername**

  The username to be used to connect to the LDAP server. Full write access is required by this username.

- **CredentialLocationPW**

  The password of the above username.

- **downgradeable**

  The value 1 means true and 0 means false. If downgradable is true, a delegation request can be downgraded by the DIS if the request exceeds the limits of the delegation policy. If downgradable is false, the DIS will only accept delegation requests that are strictly within the limits of the delegation policy.

- **SigningKeyFile**

  The path of the DIS's private key for signing delegated role attribute certificates.

- **SigningKeyPW**

  The password of the above key file.

- **LDAP_AC_Attribute**

  The LDAP attribute type used to store attribute certificates.

- **LDAP_PKC_Attribute**

  The LDAP attribute type used to store users' public key certificates.

- **DebugLevel**

  The value can be either trace, debug, info, warn, error or fatal. This parameter tells the DIS how much debug information to output.

- **log4jlocation**

  The path of the log4j configuration file to use for logging. Use in combination with the DebugLevel parameter.

- **TrustedProxy**

  The DN of an entity trusted to act as a user proxy to the DIS. The DN must be the entity's subject DN from its public key certificate. The TrustedProxy parameter can be repeated as many times as necessary, for multiple trusted proxies. The TrustedProxy parameter will normally be set to the DN of the Apache HTTP Server hosting the DIS Web User Interface.

- **SearchRequestor**

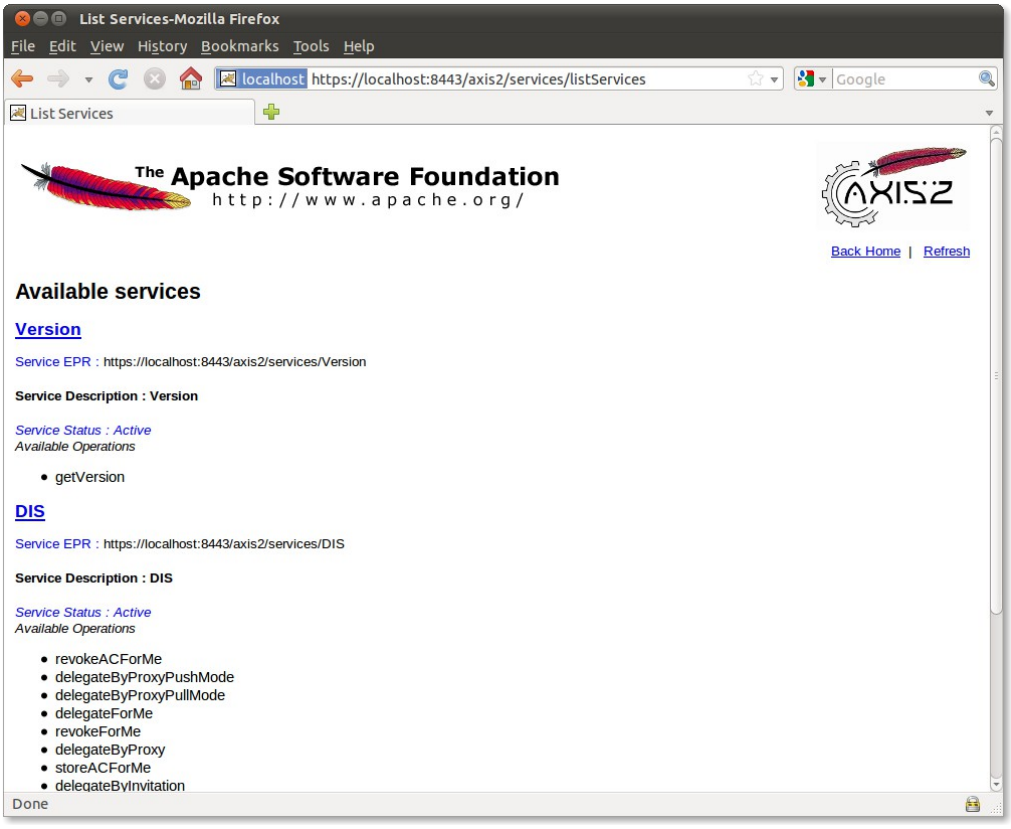
  The value can be either “anyone” or “revokers”. If the value is “anyone”, any user can browse the attribute certificates of all users. If the value is “revokers”, a user can only browse the attribute certificates that they can revoke.

Step 7: Deploy the DIS Web Service

Copy the `dis` directory with the Web Service package into `$CATALINA_HOME/webapps/axis2/WEB-INF/services`.

```
cp -r /home/user/dis/serverSide/dis  $CATALINA_HOME/webapps/axis2/WEB-INF/services/
```

Now, go to https://localhost:8443/axis2/services/listServices in a web browser. The DIS should be listed as an available service.



Step 8: Install OpenLDAP

OpenLDAP is required for holding users' attribute certificates that contain delegated roles.

Oracle Berkeley DB is a required dependency of OpenLDAP. Download Oracle Berkeley DB 5.2.28 from http://www.oracle.com/technetwork/database/berkeleydb/index.html to `~/dis`. Extract the downloaded file to the same location. Execute the following commands there:

```
1. cd build_unix
2. ../dist/configure --prefix=/home/user/dis/bdb4
3. make
4. make install
```

Download OpenLDAP 2.4.26 from http://www.openldap.org/ to `~/dis`. Extract the downloaded file to the same location. Execute the following commands there:

```
1. env CPPFLAGS="-I/home/user/dis/bdb4/include" LDFLAGS="-
   L/home/user/dis/bdb4/lib" LD_LIBRARY_PATH="/home/user/dis/bdb4/lib"
   ./configure --prefix=/home/user/dis/openldap
2. make depend
3. make
4. make install
```

OpenLDAP should now be installed to `/home/user/dis/openldap`.

Step 9: Import sample data into OpenLDAP

OpenLDAP has to be configured to understand attribute certificates.

In `~/dis/openldap/etc/openldap/slapd.conf` change the "suffix" and "rootdn" to "c=gb" and "cn=Manager,c=gb" respectively.

Append the following to `~/dis/openldap/etc/openldap/schema/core.schema`:

```
attributetype (2.5.4.58
        NAME 'attributeCertificateAttribute'
        DESC 'A binary attribute certificate'
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.5 )


attributetype (1.2.826.0.1.3344810.1.1.14
        NAME 'permisRole'
        DESC 'A permisRole to be passed to Shibboleth'
        SUP name )


attributetype (1.2.826.0.1.3344810.1.1.99
        NAME 'delegations'
        DESC 'Delegation metadata'
        EQUALITY caseExactIA5Match
        SUBSTR caseExactIA5SubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )


objectclass (2.5.6.24
        NAME 'pmiUser'
        SUP top AUXILIARY
        DESC 'a pmi entity that can contain X509 ACs'
        MAY (attributeCertificateAttribute $ permisRole $ delegations $
userPassword $ telephoneNumber $ seeAlso $ description $ uid $ userCertificate $
email) )
```

Test the new schema by executing:

```
1. env LD_LIBRARY_PATH="/home/user/dis/bdb4/lib"
~/dis/openldap/sbin/slaptest -u
```

Start OpenLDAP by executing:

```
1. sudo env LD_LIBRARY_PATH="/home/user/dis/bdb4/lib"
~/dis/openldap/libexec/slapd -d 1
```

Import the sample data by executing the following commands:

```
1. sudo apt-get install ldap-utils
2. ldapadd -f ~/dis/serverSide/data.ldif -xv -D "cn=Manager,c=gb" -w
secret
```

Step 10: Test the DIS Web Service works

We are now ready to test the DIS Web Service.

First we will check the DIS initialises properly. Restart Tomcat. After a minute or two, the last of your log4j log file (as specified in your log4j configuration file) should be similar to:

```
2010-05-08 10:08:45,123 [main] DEBUG issrg.dis.DISCore - The DIS is
initialized. Log is enable at level null, policy ID: null obtained from
file://home/user/dis/serverSide/policy.xml
```

If it is, the DIS has initialised properly.

Next, we will use the application soapUI to send a request to the DIS Web Service. SoapUI is available from http://sourceforge.net/projects/soapui/files/soapui/.

For that, SoapUI must be configured for SSL connection and authentication.

To configure SoapUI, go to File -> Preferences -> SSL Settings.

In the field "**KeyStore**", click Browse, and navigate to the soa-cert.p12 file (`/home/user/dis/serverSide/disServerKeystore/soa-cert.p12`).

In the field "**KeyStore Password**", type "l3tM3InNow".

In the "**Client Authentication**" option, mark "**requires client authentication**".

Click Ok.

Now we must load the WSDL of DIS into SoapUI.

Create a new project, using "DIS" as project name.

```
Paste the address of the DIS WSDL in the "Initial WSDL/WADL" field:
https://localhost:8443/axis2/services/DIS?wsdl
```

Click Ok.

Once the project has been created, send a `searchForMe` request with both arguments (`requestorDN` and `holderDN`) as `cn=soa,ou=admin,o=permisv5,c=gb` to the DIS.

The following is an example of request:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsd="http://sec.cs.kent.ac.uk/dis/xsd">
   <soap:Header/>
   <soap:Body>
      <xsd:searchForMe>
         <!--Optional:-->
         <xsd:args0>cn=soa,ou=admin,o=permisv5,c=gb</xsd:args0>
         <!--Optional:-->
         <xsd:args1>cn=soa,ou=admin,o=permisv5,c=gb</xsd:args1>
      </xsd:searchForMe>
   </soap:Body>
</soap:Envelope>
```

The DIS should return the following response.

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-
envelope">
   <soapenv:Body>
      <ns:searchForMeResponse xmlns:ns="http://sec.cs.kent.ac.uk/dis/xsd"
xmlns:ax21="http://dis.issrg/xsd">
         <ns:return>CN=SOA,OU=ADMIN,O=PERMISV5,C=GB|
CN=SOA,OU=ADMIN,O=PERMISV5,C=GB|
754159980139181934436096906845656646782170240000|PMI_XML_POLICY|Thu Jul 31
00:00:00 BST 2008|Wed Jul 31 00:00:00 BST 2013</ns:return>
         <ns:return>CN=SOA,OU=ADMIN,O=PERMISV5,C=GB|
CN=SOA,OU=ADMIN,O=PERMISV5,C=GB|
218007036583948315408630461932214882573323355354|PMI_XML_POLICY|Fri Aug
22 01:00:00 BST 2008|Thu Aug 22 01:00:00 BST 2013</ns:return>
      </ns:searchForMeResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

If you receive a response similar to the one presented above, DIS has been successfully installed and is ready for use.