Delegation Issuing Service (DIS) Acceptance Tests

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	27 October 2010	Acceptance tests separated into its own document
2.6	13 July 2011	Revised and updated against latest version of DIS

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).

This document describes a set of acceptance tests for the Delegation Issuing Service (DIS). The result of the testing process is totally dependent on your delegation policy stored in the policy certificate, the DIS's role attribute certificate, the requests from users to the DIS, the sequence of the requests and the role attribute certificates stored in the user entries in the LDAP (so you should ensure that the testing entries do not contain any role ACs before the acceptance testing process starts).

In the following acceptance tests, it is assumed that the DIS Web service has been installed according with the DIS-Service-Installation guide. The tests consider that the DIS Web service is using the example delegation policy provided with the DISService package (included in Annex A), and that the LDAP directory is populated with the contents of the `data1.ldif` file provided with the DISService package.

Understanding the request's parameters to the DIS and the reply's parameters from the DIS

Instead of showing you the Apache client screenshots, we present the parameters of each SOAP request and reply from the DIS. The acceptance tests can be conducted using a SOAP tool of your choice, and you will need to prepare your requests according to the requests' parameters and compare the results' parameters with the results' parameters replied from the DIS.

These tests involve calls to the `delegateByProxy` method of the DIS Web service. In the following, we describe the request/result parameters.

Request

Look at the example SOAP request below.

```
Requester: cn=soa,ou=admin,o=permisv5,c=gb
Holder:   cn=admin1,ou=admin,o=permisv5,c=gb
RoleType:  permisRole  RoleValues:  Admin, Professor
From:  2003.01.01 To:  2011.01.01
Assertion:  cannot
Depth:  0
```

The request parameters are as follows:

- “Requester” is the user that requests the DIS to issue ACs (corresponding to the Delegator).
- “Holder” is the person (or entry) that the requester wants to delegate an AC to.
- “RoleType” is the type of the attributes. In the example provided, “permisRole” is the only valid RoleType.
- “RoleValues” are the values of the attributes. There are five possible values for the “RoleValues” in our test cases: Admin, Professor, Researcher, Staff and Student. These are defined in the dis-policy.xml file used in our example installation.
- “From” and “To” form the validity time of the requested AC.
- “Assertion” receives one value in the set of two values: {can, cannot}. When you click on the “ALLOW the Holder to execute these roles” radio button, the assertion value is “can” represents that the Holder is allowed to execute these roles, while “cannot” represents that the “Holder is not allowed to execute these roles.
- “Depth” is the delegation depth for the requested AC. It expects an integer value, where 0 means unlimited, -1 means no further delegations, and positive numbers represent the number of further delegations.

Reply's parameters from the DIS

Look at the “Reply” below.

Reply:

```
Accepted|CN=admin1,OU=admin,O=permisv5,C=gb|permisRole:Admin| Jan 01
2004| Jan 01 2010|Holder can not assert privileges|0
```

The “Reply” parameters are as follows:

- “Accepted” means the requested AC has been issued.
- “CN=admin1,OU=admin,O=permisv5,C=gb” is the holder's DN (The delegate).

- “permisRole:Admin” is the attribute value (or list of attribute values) of the attribute in the issued AC.
- “Jan 01 2004| Jan 01 2010” is the validity time (from and to) of the issued AC.
- “Holder can not assert privileges” is the expression of the “cannot” value of the “Assertion” variable.
- “0” means the delegation depth is unlimited (value -1 means “No delegation”, value 0 means “unlimited delegation”, value 1 means “one step delegation” etc.)

Actually, both the request and reply messages also contain hh:mm:ss for time instances (for example, the “From” time instance may have the value “2004:01:01 00:00:00”). Since we do not provide the time, the default value is set to 00:00:00 GMT by DIS, unless they are constrained by the Delegation Policy.

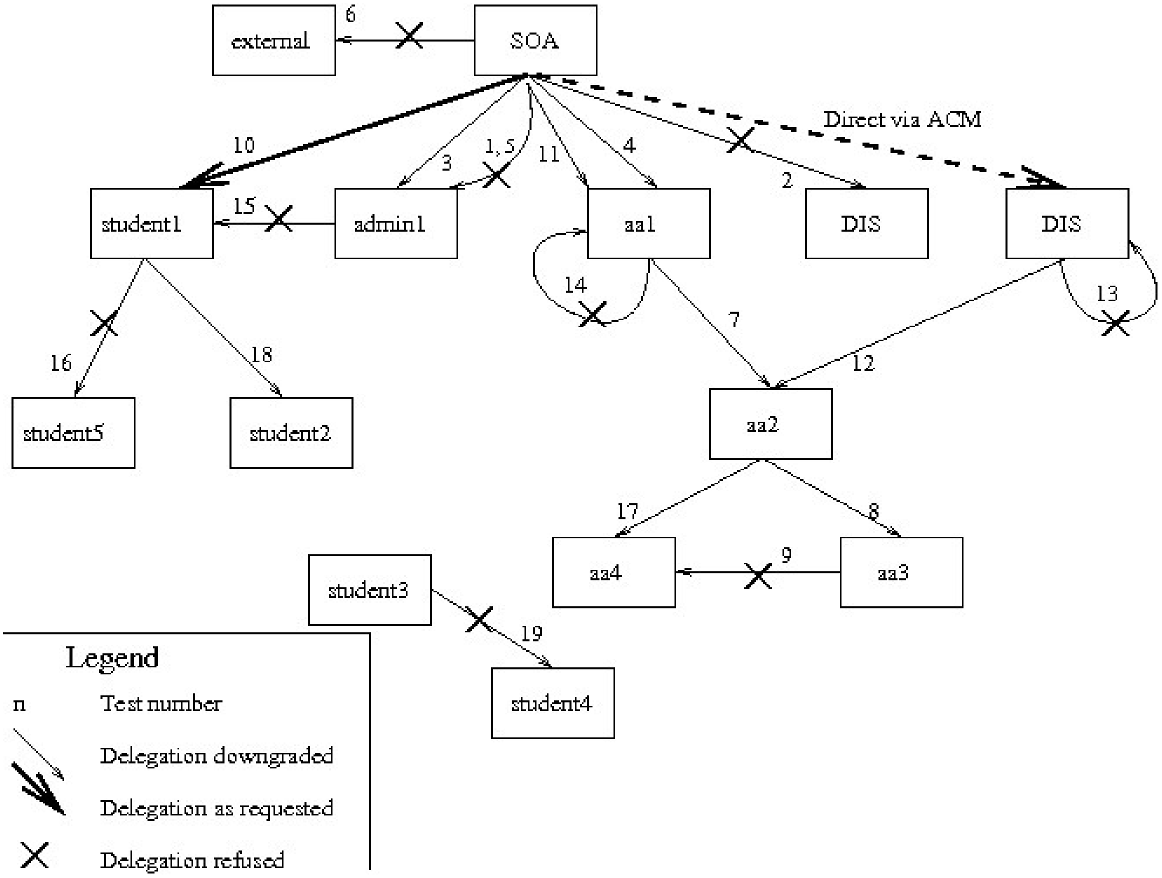
Test cases

Now you can proceed with the test cases.

DO NOT CHANGE THE SEQUENCE OF THE TEST CASES AND THE PARAMETERS OF THE REQUESTS. OTHERWISE, YOU MAY NEED TO RESTART THE TESTING PROCESS FROM BEGINNING. Some former test cases are the preparation for some later test cases.

Also, the date on the computer hosting the DIS web service should be set to anything before 1st June 2004. This is because these particular tests depend on the current date being before this date. If it is not, some tests will result in an incorrectly denied delegation (e.g. if the requested end time is before the current date) or a slightly different delegation (e.g. if the current date is between the requested start and end time).

The following diagram shows the sequence of delegations that are attempted in the following tests. The delegation from SOA to DIS has been stored in the LDAP repository during the installation of the of the DIS web service.



Test number: 1

Description: to confirm that the DIS cannot be used to delegate any privilege ACs until it has been directly given a privilege AC by the SOA
In order to run this test, the attributeCertificateAttribute of the dis must be removed from its respective LDAP entry. Using the LDAP browser of your preference, locate the entry with the following DN: “cn=dis,ou=admin,o=PERMISv5,c=gb”, and remove the `attributeCertificateAttribute` value.
Then, invoke the `delegateByProxy` method with the following parameters.

```
Requester: cn=soa,ou=admin,o=permisv5,c=gb
Holder:    cn=admin1,ou=admin,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Admin
From:      2004.01.01  To: 2012.01.01
Assertion: can not
Depth:     0
```

Reply:

```
Delegation Issuing Service does not have enough privilege to issue this
certificate
```

Preparation for all the following test cases. Directly issue an AC to the DIS and store it in your LDAP server.

Before running the next test cases, it is necessary to authorise DIS to delegate.

Using the LDAP browser of your choice, import the disace.ace into the attributeCertificateAttribute value of the DIS LDAP entry. Its DN is “cn=dis,ou=admin,o=PERMISv5,c=gb”.

Test number: 2

Description: to confirm that the DIS cannot delegate any additional privilege ACs to itself, even if the requester has the necessary privileges.

```
Requester: cn=soa,ou=admin,o=permisv5,c=gb
Holder:    cn=dis,ou=admin,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Admin
From:      2004.01.01 To: 2012.01.01
Assertion: can not
Depth:     0
```

Reply:

```
Delegation Issuing Service does not have enough privilege to issue this
certificate
```

Test number: 3

Description: to confirm that a delegator can delegate multiple roles in an AC, but the DIS will condense these to the highest role in the role hierarchy (if possible). The “Assertion” variable is “can” and delegation depth is unlimited. In this example, the From date has been constrained to comply with the Delegation Policy.

```
Requester: cn=soa,ou=admin,o=permisv5,c=gb
Holder:    cn=admin1,ou=admin,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Admin, Professor
From:      2003.01.01 To: 2012.01.01
Assertion: can
Depth:     0
```

Reply:

```
Accepted|CN=admin1,OU=admin,O=permisv5,C=gb|permisRole:Admin| Jan 01
2004| Jan 01 2010|Holder can assert privileges|0
```

Test number: 4

Description: to confirm that a delegator can issue multiple roles in an AC, and in this case the DIS is unable to condense them. The “Assertion” variable is “can” and delegation depth is 2. Note that the validity date had been altered to conform to the delegation policy.

```
Requester: cn=soa,ou=admin,o=permisv5,c=gb
Holder:    cn=aa1,ou=staff,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Professor, Researcher
From:      2001.01.01  To: 2012.01.01
Assertion: can
Depth:     2
```

Reply:

```
Accepted|CN=aa1,OU=staff,O=permisv5,C=gb|permisRole:Professor,
Researcher| Jun 01 2004| Jan 01 2006|Holder can assert privileges|2
```

Test number: 5

Description: to confirm that the DIS cannot issue any ACs with an unknown attribute type or attribute value.

```
Requester: cn=soa,ou=admin,o=permisv5,c=gb
Holder:    cn=admin1,ou=admin,o=permisv5,c=gb
RoleType:  perRole  RoleValues: Staff
From:      2005.01.01  To: 2012.01.01
Assertion: can
Depth:     2
```

Reply:

```
Role type or role value is not supported in policy
```

Test number: 6

Description: to confirm that a delegator cannot issue any AC to an entry that is not in any subject domain.

```
Requester: cn=soa,ou=admin,o=permisv5,c=gb
Holder:    cn=external, o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Staff
From:      2005.01.01  To: 2012.01.01
Assertion: can
Depth:     2
```

Reply:

```
Issuer does not have enough privilege or can not downgrade privilege or
wrong request
```

Test number: 7

Description: to confirm that a user can delegate a subset of their privileges to another user in the same group, but not with a greater delegation depth than they have been given (unlimited requested, but DIS limits it to 1 due to previous delegation in Test 4).

```
Requester: cn=aa1,ou=staff,o=permisv5,c=gb
Holder:    cn=aa2,ou=staff,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Professor
From:      2004.06.01  To: 2012.08.27
Assertion: can
Depth:     0
```

Reply:

```
Accepted|CN=aa2,OU=staff,O=permisv5,C=gb|permisRole:Professor| Jun 01
2004| Jan 01 2006|Holder can assert privileges|1
```

Test number: 8

Description: to confirm that the DIS enforces delegation policy, enforcing delegation depth. Note that the delegate is no longer allowed to delegate further.

```
Requester: cn=aa2,ou=staff,o=permisv5,c=gb
Holder:    cn=aa3,ou=staff,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Professor
From:      2004.06.01  To: 2012.08.27
Assertion: can
Depth:     0
```

Reply:

```
Accepted|CN=aa3,OU=staff,O=permisv5,C=gb|permisRole:Professor| Jun 01
2004| Jan 01 2006|Holder can assert privileges|-1
```

Test number: 9

Description: to confirm that the DIS enforces delegation policy, enforcing delegation depth and stopping a user delegating when he is not allowed to.

```
Requester: cn=aa3,ou=staff,o=permisv5,c=gb
Holder:    cn=aa4,ou=staff,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Staff
From:      2004.06.01  To: 2012.08.27
Assertion: can
Depth:     0
```

Reply:

```
Issuer does not have enough privilege or can not downgrade privilege or
wrong request
```

Test number: 10

Description: to confirm that the SOA can issue any AC to any entry in group 3 (student domain), delegation depth is 2 and maximum “From” and “To” time instances.

```
Requester: cn=soa,ou=admin,o=permisv5,c=gb
Holder:    cn=student1,ou=student,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Student
From:      2004.06.10  To: 2012.08.27
Assertion: can
Depth:     2
```

Reply:

```
Accepted|CN=student1,OU=student,O=permisv5,C=gb|permisRole:Student| Jun
10 2004| Aug 27 2007|Holder can assert privileges|2
```

Test number: 11

Description: to confirm that the DIS will enforce the Delegation Policy and not assign attributes (roles) to users who are not allowed them, or for longer time periods than allowed. In this test one attribute value (Admin) is downgraded (to Professor) because it cannot be assigned to any staff in group 2, even though the delegator has the role. Also both the “From” and “To” time instances are constrained.

```
Requester: cn=soa,ou=admin,o=permisv5,c=gb
Holder:    cn=aa1,ou=staff,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Researcher, Admin
From:      2003.06.01  To: 2012.08.27
Assertion: can
Depth:     2
```

Reply:

```
Accepted|CN=aa1,OU=staff,O=permisv5,C=gb|permisRole: Professor,
Researcher | Jun 1 2004| Aug 27 2012|Holder can assert privileges|2
```

Test number: 12

Description: to confirm that the DIS can issue an AC to a subject, when the requester is itself.

```
Requester: cn=dis,ou=admin,o=permisv5,c=gb
Holder:    cn=aa2,ou=staff,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Researcher
From:      2004.01.01  To: 2012.01.01
Assertion: cannot
Depth:     0
```

Reply:

```
Accepted|CN=aa2,OU=staff,O=permisv5,C=gb|permisRole:Researcher| Jun 01
2004| Aug 27 2007|Holder can not assert privileges|0
```

Test number: 13

Description: to confirm that the DIS can not issue an AC to itself when the requester is also itself.

```
Requester: cn=dis,ou=admin,o=permisv5,c=gb
Holder:    cn=dis,ou=admin,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Admin, Professor
From:      2006.01.01  To: 2012.01.01
Assertion: cannot
Depth:     0
```

Reply:

```
Issuer does not have enough privilege or can not downgrade privilege or
wrong request
```

Test number: 14

Description: to confirm that a holder cannot delegate an AC to himself.

```
Requester: cn=aa1,ou=staff,o=permisv5,c=gb
Holder:    cn=aa1,ou=staff,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Student, Researcher
From:      2004.06.01  To: 2011.08.27
Assertion: can
Depth:     2
```

Reply:

```
Issuer does not have enough privilege or can not downgrade privilege or
wrong request
```

Test number: 15

Description: to confirm that subjects cannot delegate ACs to holders that are in a different subject domain.

```
Requester: cn=admin1,ou=admin,o=permisv5,c=gb
Holder:    cn=student1,ou=student,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Student
From:      2004.06.01  To: 2011.08.27
Assertion: can
Depth:     2
```

Reply:

```
Issuer does not have enough privilege or can not downgrade privilege or
wrong request
```

Test number: 16

Description: to confirm that the DIS can not issue an AC when the holder is excluded from a subject domain.

```
Requester: cn=student1,ou=student,o=permisv5,c=gb
Holder:    cn=student5,ou=student,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Student
From:      2005.01.01  To: 2006.05.01
Assertion: can
Depth:     0
```

Reply:

```
Issuer does not have enough privilege or can not downgrade privilege or
wrong request
```

Test number: 17

Description: to confirm that when a requester has two separate roles, delegated from different superiors, he cannot merge them into a single combined delegated credential. (Two separate delegations should be performed in order to stop the chains from cross linking.) Only the most superior role will be delegated in this request. When the two roles have the same superiority (Professor and Researcher) then the one with longer validity period will be used for delegation.

```
Requester: cn=aa2,ou=staff,o=permisv5,c=gb
Holder:    cn=aa4,ou=staff,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Professor, Researcher
From:      2005.01.01  To: 2011.08.27
Assertion: can
Depth:     0
```

Reply:

```
Accepted|CN=aa4,OU=staff,O=permisv5,C=gb|permisRole:Researcher| Jan 01
2005| Aug 27 2007|Holder can assert privileges|0
```

Test number: 18

Description: to confirm that a subject can delegate to another subject in his domain, providing it is in accordance with the Delegation Policy. In this case, the “From” time and the delegation depth are downgraded.

```
Requester: cn=student1,ou=student,o=permisv5,c=gb
Holder:    cn=student2,ou=student,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Student
From:      2004.03.01  To: 2007.08.27
Assertion: can
Depth:     2
```

Reply:

```
Accepted|CN=student2,OU=student,O=permisv5,C=gb|permisRole:Student| Jun
10  2004| Aug 27 2007|Holder can assert privileges|1
```

Test number: 19

Description: to confirm that a requester cannot delegate a privilege that he does not hold.

```
Requester: cn=student3,ou=student,o=permisv5,c=gb
Holder:    cn=student4,ou=student,o=permisv5,c=gb
RoleType:  permisRole  RoleValues: Student
From:      2005.01.01  To: 2006.05.01
Assertion: can
Depth:     0
```

Reply:

```
Issuer does not have enough privilege or can not downgrade privilege or
wrong request
```

Revocation Test Cases

The following test cases exercises the `searchForMe` and `revokeForMe` methods of DIS. The searchForMe method returns a String with the following format: `issuerDN|HolderDN|serialNumber|roleValues|From|To|Depth|assertion|issued on behalf of||Validity`.

Test number: 20

Description: to confirm that a user that is out of all the subject domains can search and view any issued attributes.

```
SearchForMe parameters:
Requester: cn=external, o=permisv5, c=gb
Holder: cn=aa1, ou=staff, o=permisv5, c=gb
```

ACs returned

```
Holder: cn=aa1, ou=staff, o=permisv5, c=gb
Issuer: cn=dis, ou=admin, o=permisv5, c=gb
Issued on behalf of : cn=soa,ou=admin, o=permisv5, c=gb
Role(s): Professor, Researcher
```

and another AC:

```
Holder: cn=aa1, ou=staff, o=permisv5, c=gb
Issuer: cn=dis, ou=admin, o=permisv5, c=gb
Issued on behalf of : cn=soa,ou=admin, o=permisv5, c=gb
Role(s): Professor, Researcher
```

Test number: 21

Description: an entry can search and view attributes of another entry which is in a different subject domain but can not revoke any of these attributes.

```
Search attributes:
Requester: cn=aa1, ou=staff, o=permisv5, c=gb
Holder: cn=student2, ou=student, o=permisv5, c=gb
```

ACs returned

```
Holder: cn=student2, ou=student, o=permisv5, c=gb
Issuer: cn=dis, ou=admin, o=permisv5, c=gb
Issued on behalf of : cn=student1, ou=student, o=permisv5, c=gb
Role(s): Student
```

(This AC was issued in test case 18)

Now, revoke the AC using the `revokeForMe` method, using the following parameters:

```
Requester: cn=aa1, ou=staff, o=permisv5, c=gb
Holder: cn=student2, ou=student, o=permisv5, c=gb
IssuerDN: cn=dis, ou=admin, o=permisv5, c=gb
serial: <the serial of the search reply>
```

The reply is:

```
You are not allowed to revoke an Attribute that you do not hold or did
not issue
```

Test number: 22

Description: to confirm that a user can revoke an attribute which was issued on his behalf.

```
Search attributes:
Requester: cn=aa1, ou=staff, o=permisv5, c=gb
Holder: cn=aa2, ou=staff, o=permisv5, c=gb
```

ACs returned

```
Holder: cn=aa2, ou=staff, o=permisv5, c=gb
Issuer: cn=dis, ou=admin, o=permisv5, c=gb
Issued on behalf of : cn=aa1, ou=staff, o=permisv5, c=gb
Role(s): Professor
```

(This AC was issued in test case 7)

Now, revoke the AC using the `revokeForMe` method, using the following parameters:

```
Requester: cn=aa1, ou=staff, o=permisv5, c=gb
Holder: cn=aa2, ou=staff, o=permisv5, c=gb
IssuerDN: cn=dis, ou=admin, o=permisv5, c=gb
serial: <the serial of the search reply>
```

The reply is:

```
Requested Attribute is revoked
```

Test number: 23

Description: to confirm that a user can revoke an attribute issued to him by anyone

```
Search attributes:
Requester: cn=student2, ou=student, o=permisv5, c=gb
Holder: cn=student2, ou=student, o=permisv5, c=gb
```

ACs returned

```
Holder: cn=student2, ou=student, o=permisv5, c=gb
Issuer: cn=dis, ou=admin, o=permisv5, c=gb
Issued on behalf of : cn=student1, ou=student, o=permisv5, c=gb
Role(s): Student
```

(This AC was issued in test case 18)

Now, revoke the AC using the `revokeForMe` method, using the following parameters:

```
Requester: cn=student2,ou=student,o=permisv5,c=gb
Holder: cn=student2,ou=student,o=permisv5,c=gb
IssuerDN: cn=dis,ou=admin,o=permisv5,c=gb
serial: <the serial of the search reply>
```

The reply is:

```
Requested Attribute is revoked
```

Test number: 24

Description: to confirm that an issuer can revoke any attribute issued by it

```
Search attributes:
Requester: cn=dis, ou=admin, o=permisv5, c=gb
Holder: cn=admin1, ou=admin, o=permisv5, c=gb
```

ACs returned

```
Holder: cn=admin1, ou=admin o=permisv5, c=gb
Issuer: cn=dis, ou=admin, o=permis, c=gb
Issued on behalf of : cn=SOA, ou=admin, o=permisv5, c=gb
Role(s): Admin
```

(This AC was issued in test case 3)

Now, revoke the AC using the `revokeForMe` method, using the following parameters:

```
Requester: cn=dis,ou=admin,o=permisv5,c=gb
Holder: cn=admin1,ou=admin,o=permisv5,c=gb
IssuerDN: cn=dis,ou=admin,o=permisv5,c=gb
serial: <the serial of the search reply>
```

The reply is:

```
Requested Attribute is revoked
```

Test number: 25

Description: to confirm that an attribute authority can revoke any attribute that it did not issue itself, if it has enough privileges to have issued that attribute.
Note. This feature is enabled so as to allow a manager to revoke the attributes of a staff member when the issuing manager is not available.

```
Search attributes:
Requester: cn=aa1,ou=staff,o=permisv5,c=gb
Holder: cn=aa3,ou=staff,o=permisv5,c=gb
```

ACs returned

```
Holder: cn=aa3, ou=staff o=permisv5, c=gb
Issuer: cn=dis, ou=admin, o=permisv5, c=gb
Issued on behalf of : cn=aa2, ou=staff, o=permisv5, c=gb
Role(s): Professor
```

(This AC was issued in test case 8)

Now, revoke the AC using the `revokeForMe` method, using the following parameters:

```
Requester: cn=aa1,ou=staff,o=permisv5,c=gb
Holder: cn=aa3,ou=staff,o=permisv5,c=gb
IssuerDN: cn=dis,ou=admin,o=permisv5,c=gb
serial: <the serial of the search reply>
```

The reply is:

```
Requested Attribute is revoked
```

Test number: 26

Description: to confirm that the SOA can revoke any attribute.

```
Search attributes:
Requester: cn=soa,ou=admin,o=permisv5,c=gb
Holder: cn=aa4,ou=staff,o=permisv5,c=gb
```

ACs returned

```
Holder: cn=aa4, ou=staff, o=permisv5, c=gb
Issuer: cn=dis, ou=admin, o=permisv5, c=gb
Issued on behalf of : cn=aa2, ou=staff, o=permisv5, c=gb
Role(s): Researcher
```

(This AC was issued in test case 17)

Now, revoke the AC using the `revokeForMe` method, using the following parameters:

```
Requester: cn=soa,ou=admin,o=permisv5,c=gb
Holder: cn=aa4,ou=staff,o=permisv5,c=gb
IssuerDN: cn=dis,ou=admin,o=permisv5,c=gb
serial: <the serial of the search reply>
```

The reply is:

```
Requested Attribute is revoked
```

Annex A. The Delegation Policy

The PERMIS Policy (X.509 PMI RBAC) is composed of a number of sub-policies and they are briefly presented below.

SubjectPolicy

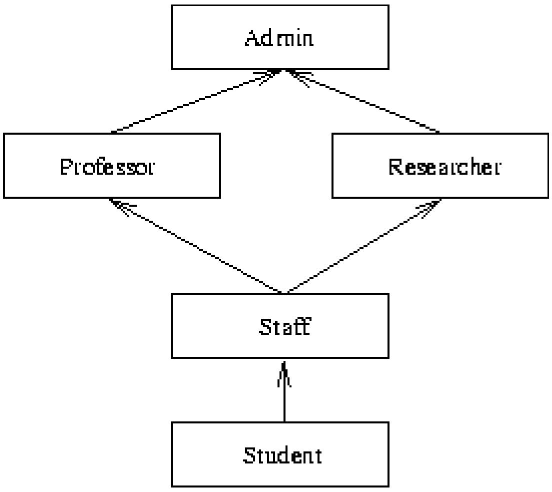
The SubjectPolicy specifies the domains of users who may be granted roles and delegation privileges within the overall PMI policy. Subjects within a subject domain can delegate to other subjects in the same domain, but not to subjects in other domains (i.e. cross domain delegation is not allowed.) Each domain is specified as an LDAP subtree. In the testing policy, we have the following subject domains:

```
Domain ID="student" with LDAPDN="ou=student,o=permisv5,c=gb"

but excluding “cn=student5,ou=student,o=permisv5,c=gb” and including
“cn=dis,ou=admin,o=permisv5,c=gb”

Domain ID="staff" with LDAPDN="ou=staff,o=permisv5,c=gb" and including
“cn=dis, ou=admin,o=permisv5,c=gb”

Domain ID="admin" with LDAPDN="ou=admin,o=permisv5,c=gb"
```

(Note that the DIS needs to part of each subject domain that it is allowed to delegate in)

RoleHierarchyPolicy

The RoleHierarchyPolicy defines the role hierarchies that are supported by a specific RBAC policy. Each role hierarchy is specified as a set of Superior-Subordinates attribute values. Each superior role can have multiple subordinate roles, and each subordinate role may also be a superior. In the testing policy, we have the following role hierarchy:



SOAPolicy

The SOAPolicy lists the LDAPDNs of the Sources of Authority (SoAs) that are trusted to issue roles to the subjects specified in the subject policy. There is one entity that is trusted to issue roles to subjects in our test policy, and that is the policy creator (i.e. he only trusts himself!!).

```
SOA with ID=”SOA” LDAPDN=”cn=SOA, ou=admin, o=permisv5,c=GB”
```

RoleAssignmentPolicy

The RoleAssignmentPolicy specifies which roles can be assigned to which subjects by which SoAs. For each role assignment, we also specify whether the assigned roles can be delegated or not and whether there are any time constraints on the assignment. Following is the brief review of the RoleAssignmentPolicy in our testing policy.

The SOA is trusted to assign:

1. role “Student” to subjects in the “student” domain with validity time from 2004-06-10 to 2007-08-27,
2. roles “Staff”, “Researcher” and “Professor” to subjects in the “staff” domain with validity time from 2004-06-01 to 2007-08-27,
3. role “Admin” to subjects in the “admin” domain with validity time from 2004-01-01 to 2010-01-1.

The SOA has unlimited delegation depth in the testing policy.

Delegation Policy in XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 4 U (http://www.xmlspy.com) by o (u) -->
<!--Sample XML file generated by XMLSPY v5 rel. 4 U
(http://www.xmlspy.com)-->
<!DOCTYPE X.509_PMI_RBAC_Policy>
<X.509_PMI_RBAC_Policy OID="1.2.826.0.1.3344810.6.0.0.1">
  <SubjectPolicy>
    <SubjectDomainSpec ID="student">
      <Include LDAPDN="ou=student,o=permisv5,c=GB">
        <Exclude LDAPDN="cn=student5,ou=student,o=permisv5,c=GB"/>
      </Include>
      <Include LDAPDN="cn=dis,ou=admin,o=permisv5,c=gb"/>
    </SubjectDomainSpec>
    <SubjectDomainSpec ID="staff">
      <Include LDAPDN="ou=staff,o=permisv5,c=GB"/>
      <Include LDAPDN="cn=dis,ou=admin,o=permisv5,c=gb"/>
    </SubjectDomainSpec>
    <SubjectDomainSpec ID="admin">
      <Include LDAPDN="ou=admin,o=permisv5,c=GB"/>
    </SubjectDomainSpec>
  </SubjectPolicy>
  <RoleHierarchyPolicy>
    <RoleSpec Type="permisRole" OID="1.2.826.0.1.3344810.1.1.14">
      <SupRole Value="Student">
      </SupRole>
      <SupRole Value="Staff">
        <SubRole Value="Student"/>
      </SupRole>
      <SupRole Value="Professor">
        <SubRole Value="Staff"/>
      </SupRole>
      <SupRole Value="Researcher">
        <SubRole Value="Staff"/>
      </SupRole>
      <SupRole Value="Admin">
        <SubRole Value="Professor"/>
        <SubRole Value="Researcher"/>
      </SupRole>
    </RoleSpec>
```

```
</RoleHierarchyPolicy>
<SOAPolicy>
  <SOASpec ID="SOA" LDAPDN="cn=SOA, ou=admin, o=permisv5,c=GB"/>
</SOAPolicy>
<RoleAssignmentPolicy>
  <RoleAssignment>
    <SubjectDomain ID="student"/>
    <RoleList>
      <Role Type="permisRole" Value="Student"/>
    </RoleList>
    <Delegate />
    <SOA ID="SOA"/>
    <Validity>
      <Absolute Start="2009-04-01" End="2019-04-01"/>
    </Validity>
  </RoleAssignment>
  <RoleAssignment>
    <SubjectDomain ID="staff"/>
    <RoleList>
      <Role Type="permisRole" Value="Professor"/>
      <Role Type="permisRole" Value="Staff"/>
      <Role Type="permisRole" Value="Researcher"/>
    </RoleList>
    <Delegate />
    <SOA ID="SOA"/>
    <Validity>
      <Absolute Start="2009-02-01" End="2015-02-01"/>
    </Validity>
  </RoleAssignment>
  <RoleAssignment>
    <SubjectDomain ID="admin"/>
    <RoleList>
      <Role Type="permisRole" Value="Admin"/>
    </RoleList>
    <Delegate />
    <SOA ID="SOA"/>
    <Validity>
      <Absolute Start="2009-01-01" End="2013-01-01"/>
    </Validity>
  </RoleAssignment>
  </RoleAssignmentPolicy>
</X.509_PMI_RBAC_Policy>
```