

SECURE Documentation: An Active Directory Compatible Implementation of the eduPerson Schema Extensions

This document details a procedure for introducing an Active Directory (AD) compatible implementation of the eduPerson schema extensions into your AD based Directory Infrastructure. There are four sections to this document. The first provides some information about the eduPerson class and describes the environment where the deployment took place. The second section compares and contrasts the various implementation possibilities and outlines the changes that need to be made to your domain(s) and forest in order to bring Active Directory to a point where the schema extensions can be introduced. The third section details why the “reference” eduPerson attribute definitions needed to be changed in order to work with AD and how the resulting implementation differs from the original. The final section details the procedure for injecting the eduPerson class into the AD schema. But first some warning notes.

1. The resulting eduPerson class does not exactly implement the class as it is defined by [eduCause](#). The primary reason for us implementing this class was to support the operation of Shibboleth. It has been our experience that this implementation has given us the behaviour we wanted for this purpose. It would be in the interest of any other institution to test to see that they are getting the functionality they expect with the applications they want to use it with.
2. Implementing these classes involves making some fundamental changes to your Active Directory infrastructure. For the most complete implementation every server in the Active Directory **forest** needs to be upgraded to Windows Server 2003 and the functional level of the **forest** needs to be upgraded to Windows 2003. Also, once the attributes have been introduced into the schema, they cannot be removed.
3. Active Directory has been compared to a pencil without an eraser. Many of the changes you make to AD are in one direction only. Once made, many changes cannot be un-made - indeed many changes can't even be altered. Given this, it needs to be emphasised from the outset how important it is to test any of the changes you want to make to your directory infrastructure in a test environment.

1.0 Background to eduPerson

The eduPerson object class was developed by an EDUCAUSE/Internet2 working group. The aim of the group was to provide a set of directory attributes and definitions which can represent individuals in higher education.

These attribute definitions are added to the schema of an adopting institutions directory by importing an LDIF file containing the definitions of each of eduPerson class and its attributes. The problem from the point of view of an institution using a Microsoft based directory infrastructure is that AD supports an earlier set of syntax rules defining the type of data each attribute can hold. As it is not possible to add your own syntax rules to Active Directory, the LDIF file was re-written to be compatible with AD.

These extensions were developed and have been implemented at the London School of Economics (LSE). Our directory infrastructure is centralised and our directory topology is reasonably straight forward. We have a single, central directory service - which is Active Directory. We have a single AD forest containing three domains. All of our domains are single site domains. Our primary root domain contains about 30,000 user and 4000 computer objects. We use the Kerberos KDC that is contained within AD, our DNS is AD integrated, as will our Certificate Authority be. By the completion of this project all of the servers in our AD forest will be running Windows Server 2003 and the forest will be in Windows 2003 native mode. The vast majority of our client computers are Windows based and are members of one of the AD domains. The ownership and management of the domains is also centralised. Two domains are under the ownership of the IT Services department, the other is 'owned' by one of the Schools research units, which has a similar commitment to software keeping up to date.

2.0 Introducing the eduPerson Schema into Active Directory

There are three routes through which the eduPerson class could be introduced into Active Directory. All of the routes result in a partial implementation, the longer routes allow you more flexibility, but at the cost of more work in upgrading Domain Controllers in domains other than the one you want to support the eduPerson class in. Each of the three routes is outlined below along with the steps that are needed in each route to bring the directory to the point where the schema extensions can be installed. These tasks generally need to be conducted by someone with enterprise administrator privileges. The notes below are aimed at someone who has the experience to act in that capacity. All of these routes assume you are running at least an environment based on Windows 2000. If you are running Windows NT, you will need to upgrade to Windows 2000 or Windows Server 2003 before you can support the eduPerson class.

2.1 A Windows 2000 domain with Microsoft's inetOrgPerson class schema extensions installed

This route provides the least complete implementation of the eduPerson class but involves the least amount of change to your existing infrastructure. The domain you want to support the eduPerson class in must be running Windows 2000 and have Microsoft's inetOrgPerson schema extensions installed. The installation of the inetOrgPerson class is a necessary prerequisite because

many of the attributes in the eduPerson specification are inetOrgPerson attributes. However, following this route would not give you a complete implementation of the eduPerson class, nor would this allow you to support future changes in the definition of the eduPerson class. The reasons are listed below:

1. The inetOrgPerson kit for Windows 2000 is an incomplete implementation of the inetOrgPerson class as it is defined in RFC2798. The inetOrgPerson which you introduce into your schema has a different inheritance and some of the attributes have a different specification than those defined in RFC2798 “due to legacy issues”. The documentation included with the inetOrgPerson kit (which can be downloaded from the Microsoft website) goes into (slightly) more detail.
2. Even with the inetOrgPerson class installed there is one attribute (labeledURI) which is only implemented in Windows Server 2003. However, this attribute is not a “required” attribute and it may be possible to dispense with it. Indeed many other directory services do not include it by default either. See eduPerson class specification for further details.
3. The eduPerson class you introduce does not exactly implement the eduPerson class as it is defined by [eduCause](#). The difference is due to Active Directory not supporting the syntax rules used by the designers (section three gives more details). However, as a result of testing by the users of these attributes, there appears to be no functional difference in what we have implemented.
4. A feature of the version of Active Directory that ships with Windows 2000 is that once attributes are introduced into the schema their definition cannot be changed (for example, an attribute that is defined as a 'Printable String' cannot later be changed to a 'Unicode string'). The designers of the eduPerson class have in the past changed the definition of attributes, implementing the eduPerson class through this route would inhibit your ability to change attribute and class definitions once they are in place.
5. It is a feature of Active Directory that once introduced a class or an attribute cannot be deleted (it can be disabled, but not deleted). Once implemented the eduPerson class definition is going to be in your Active Directory for the long term.

Given the number of discrepancies and the fundamental nature of them **we would not recommend proceeding down this route**. However, if you did decide to, these are the steps you would need to take in order to be able to introduce the eduPerson class into your directory.

It is strongly recommended that you run through these procedures in a isolated test lab before you make any changes to your live domain and forest. At the very least you should go through the process of making schema modifications on test Active Directory server that has no connection with your production system. A better understanding of any possible side effects would

be found by constructing a standalone network that reproduces the key features of your production environment.

1. Ensure you have end to end replication in your Active Directory forest.
2. Download the InetOrgPerson class kit from the Microsoft website (search for W2K_IOP_kit.exe). Thoroughly familiarise yourself with documentation and Idif files therein.
3. Follow the steps in the document to introduce the inetOrgPerson class into your schema.
4. Go to section three and read about the eduPerson class, then go to section four and follow the procedure for introducing the class into your schema.

2.2 The domain you want to use the eduPerson class in upgraded to Windows Server 2003

This route implements the eduPerson class with a compliant set of inetOrgPerson attributes. It does involve a substantial amount of change to your directory service infrastructure, but does not involve upgrading all the domains in your forest. However, this route does affect your options in the future. The known limitations are listed below:

1. The eduPerson class you introduce does not exactly implement the eduPerson class as it is defined by [eduCause](#). The difference is due to AD not supporting the syntax rules used by the designers ([section 3.0](#) gives more details). However, as a result of testing by the users of these attributes, there appears to be no difference in the functionality of what we have implemented.
2. Even when you are running Windows Server 2003 you do not have the ability to change the definitions of attributes after they have been introduced (for example, an attribute that is defined as a 'Printable string' cannot later be changed to a 'Unicode string'). To have that sort of functionality you need to upgrade the functional level of your Active Directory forest. The designers of the eduPerson class have in the past changed the definition of attributes, implementing the eduPerson class through this route could inhibit your ability to support future versions of the eduPerson class.
3. It is a feature of Active Directory that once introduced a class or an attribute cannot be deleted (it can be disabled, but not deleted). Once implemented the class definition is going to be in your Active Directory for the long term.

These are the steps you would need to take in order to be able to introduce the eduPerson class into your directory. It is strongly recommended that you run through these procedures in a isolated test lab before you make any changes to your live domain and forest. At the very least you should go through the upgrade process and the process of making schema

modifications on a separate standalone test Active Directory server. A better understanding of any possible side effects would be found by constructing a standalone network that reproduces the key features of your live domain.

1. Upgrade the domain you want to support the eduPerson class in to Windows Server 2003. This will involve you making changes to the forest, obviously you need to have end to end replication before you can do this. Some consequences of upgrading your domain are:
 - Once upgraded your Domain Controllers will not support the netbuei protocol.
 - The default group policy setting is to force secure SMB signing with clients. If you have users using Windows 9x or older copies of NT this could stop them from being able to log into the domain. Though inadvisable, this setting could be disabled if it were really necessary.

There is a substantial amount of documentation on how to upgrade a Windows 2000 domain to Windows Server 2003 on the Microsoft website. Knowledge Base Article 325379, '[How to upgrade Windows 2000 domain controllers to Windows Server 2003](#)', is a good place to start. You should allocate several months to ensure that this is done properly.

2. Go to section three and read about the eduPerson class, then go to section four and follow the procedure for introducing the class into your schema.

2.3 The Active Directory Forest is upgraded to Windows Server 2003

This route sees every Domain Controller in the Active Directory forest is upgraded to Windows Server 2003, and the functional level of the forest is raised to Windows 2003. This route will give you as complete support for the eduPerson class as you are going to get. If you have a large forest, the upgrading of every domain controller in every domain will be a substantial undertaking. The benefit of following this route will be to allow you to support the modification of class attributes after they have been added to the schema. It is however a feature of Active Directory that once introduced a class or an attribute cannot be deleted (it can be disabled, but not deleted) - once introduced, the class definition is going to be in your Active Directory for the long term. Also, once the functional level of a domain or forest has been raised it cannot be changed back again. When the functional level is raised to Windows 2003 in a domain, any new domain controller you add can only be running Windows Server 2003.

These are the steps you need to take to introduce the eduPerson class into your directory. It is strongly recommended that you run through these procedures in a isolated test lab before you make any changes to your live domain and forest. At the very least you should go through the upgrade process and the process of making schema modifications on a separate

standalone test Active Directory server. A better understanding of any possible side effects would be found by constructing a standalone network that reproduces the key features of your live domain.

1. Follow the procedure in 2.2 above to upgrade one domain to Windows Server 2003.
2. Follow the same procedure for all the other domains in your forest.
3. Raise the functional level of your forest. Follow the procedure outlined in Microsoft's Knowledge Base Article 322692, '[How to raise domain and forest functional levels in Windows Server 2003](#)'.
4. Go to Section 2 and follow the procedure for introducing the eduPerson class into your schema.

3.0 The eduPerson Class

This section details why the eduPerson class used with Active Directory differs from the class maintained by eduCause, and documents where the differences lie. The [eduPerson Object Class](#) constitutes the 'reference' version.

The eduPerson specification document (as at December 2003) uses 43 attributes to describe an individual within a higher education establishment. These 42 attributes are drawn from four different object classes; 9 from the eduPerson object class, 19 attributes from the inetOrgPerson class, 9 attributes from the orgPerson class and 5 attributes from the person class - one other attribute is defined but it is suggested that its use be avoided.

The version of Active Directory supplied with Windows Server 2003 contains compliant definitions for the person, orgPerson and inetOrgPerson classes, so the implementation task is one of introducing the nine attributes in the eduPerson class into the AD schema. The 'reference' version of the eduPerson class is supplied as an ldif file, sites import this to modify their schema. There are two problems with using the ldif file supplied by eduCause with Active Directory. The first is that the attributes for the eduPerson class are defined using an LDAP syntax defined in [RFC2252](#), 'Lightweight directory access protocol (v3): attribute syntax definitions'. Active Directory does not support this attribute syntax, instead it uses an older and more limited X500 syntax. The other problem is that RFC2252 describes a set of equality matching rules for attributes, it is not possible to alter the matching rules beyond the default behaviour of the attribute using the X500 syntax. The ldif file that is supplied here is a rewrite of eduCause's 'reference' attributes using the older attribute definitions supported by Active Directory. The resulting ldif file therefore does not implement the class exactly as defined by eduCause. Nevertheless, in our tests we found that the attribute specification using the X500 syntax gave us the behaviour we wanted.

The section below outlines each of the attributes in the eduPerson class and how they are implemented in the AD specific ldif, and how each attribute differs from its definition in the 'reference' class.

More information about how syntaxes work in Active Directory can be found on the MSDN website. Search for “characteristics of attributes” and “Syntaxes for Active Directory Attributes” at MSDN.

eduPersonAffiliation

This attribute specifies the person's relationship(s) to the institution in broad categories such as student, faculty, staff, alum, etc. The RFC2252 definition of the attribute in the 'reference' Idif is:

```
attributetypes: ( 1.3.6.1.4.1.5923.1.1.1.1
                NAME 'eduPersonAffiliation'
                DESC 'eduPerson per Internet2 and EDUCAUSE'
                EQUALITY caseIgnoreMatch
                SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )
```

1.3.6.1.4.1.1466.115.121.1.15 is the RFC2252 OID of the ldap type LDAPTYPE_DIRECTORYSTRING. The equivalent X500 syntax type is *Directory String* (aka *Unicode String*) known within ADSI as ADSTYPE_CASE_IGNORE_STRING. The OID of the X500 syntax is 2.5.5.12, oMSyntax is 64.

The eduPersonAffiliation attribute appears in the modified Idif file as:

```
dn: CN=eduPersonAffiliation,CN=Schema,CN=Configuration,DC=yourPlace,DC=edu
changetype: add
objectClass: attributeSchema
name: eduPersonAffiliation
description: eduPerson per Internet2 and EDUCAUSE
attributeID: 1.3.6.1.4.1.5923.1.1.1.1
attributeSyntax: 2.5.5.12
oMSyntax: 64
systemOnly: FALSE
isSingleValued:FALSE
```

eduPersonEntitlement

This attribute is a URI (either URN or URL) that indicates a set of rights to specific resources. The RFC2252 definition of the attribute in the 'reference' Idif is:

```
attributetypes: ( 1.3.6.1.4.1.5923.1.1.1.7
                NAME 'eduPersonEntitlement'
```

```
DESC 'eduPerson per Internet2 and EDUCAUSE'  
EQUALITY caseIgnoreMatch  
SYNTAX '1.3.6.1.4.1.1466.115.121.1.15')
```

1.3.6.1.4.1.1466.115.121.1.15 is the OID of the ldap type LDAPTYPE_DIRECTORYSTRING. The equivalent X500 syntax type is *Directory String* (aka *Unicode string*) known within ADSI as ADSTYPE_CASE_IGNORE_STRING. The OID of the X500 syntax is 2.5.5.12, oMSyntax is 64.

The eduPersonEntitlement attribute appears in the modified ldif file as:

```
dn: CN=eduPersonEntitlement,CN=Schema,CN=Configuration,DC=yourPlace,DC=ac,DC=uk  
changetype: add  
objectClass: attributeSchema  
name: eduPersonEntitlement  
description: eduPerson per Internet2 and EDUCAUSE  
attributeID: 1.3.6.1.4.1.5923.1.1.1.7  
attributeSyntax: 2.5.5.12  
oMSyntax: 64  
systemOnly: FALSE  
isSingleValued:FALSE
```

eduPersonNickname

This attribute specifies the person's nickname, or the informal name by which they are accustomed to be hailed. The RFC2252 definition of the attribute in the 'reference' ldif is

```
attributetypes: ( 1.3.6.1.4.1.5923.1.1.1.2  
NAME 'eduPersonNickname'  
DESC 'eduPerson per Internet2 and EDUCAUSE'  
EQUALITY caseIgnoreMatch  
SYNTAX '1.3.6.1.4.1.1466.115.121.1.15')
```

1.3.6.1.4.1.1466.115.121.1.15 is the OID of the ldap type LDAPTYPE_DIRECTORYSTRING. The equivalent X500 syntax type is *Directory String* (aka *Unicode String*) known within ADSI as ADSTYPE_CASE_IGNORE_STRING. The OID of the X500 syntax is 2.5.5.12, oMSyntax is 64.

The eduPersonNickname attribute appears in the modified ldif file as:

```
dn: CN=eduPersonNickname,CN=Schema,CN=Configuration,DC=lse,DC=ac,DC=uk
changetype: add
objectClass: attributeSchema
name: eduPersonNickname
description: eduPerson per Internet2 and EDUCAUSE
attributelD: 1.3.6.1.4.1.5923.1.1.1.2
attributeSyntax: 2.5.5.12
oMSyntax: 64
systemOnly: FALSE
isSingleValued:FALSE
```

eduPersonOrgDN

This attribute is the distinguished name (DN) of the of the directory entry representing the institution with which the person is associated. The RFC2252 definition of the attribute in the 'reference' Idif is:

```
attributetypes: ( 1.3.6.1.4.1.5923.1.1.1.3
    NAME 'eduPersonOrgDN'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY distinguishedNameMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.12' SINGLE-VALUE )
```

1.3.6.1.4.1.1466.115.121.1.12 is the OID of the ldap type LDAPATYPE_DN. The equivalent X500 syntax type is *DN* (aka *distinguished name* or *DN String*). Known within ADSI as ADSI_DN_STRING. The OID of the X500 syntax is 2.5.5.1, oMSyntax is 127.

The eduPersonOrgDN attribute appears in the modified Idif file as:

```
dn: CN=eduPersonOrgDN,CN=Schema,CN=Configuration,DC=lse,DC=ac,DC=uk
changetype: add
objectClass: attributeSchema
name: eduPersonOrgDN
description: eduPerson per Internet2 and EDUCAUSE
attributelD: 1.3.6.1.4.1.5923.1.1.1.3
attributeSyntax: 2.5.5.1
oMSyntax: 127
systemOnly: FALSE
isSingleValued:TRUE
```

eduPersonOrgUnitDN

This attribute is defined as the distinguished name(s) (DN) of the directory entries representing the person's Organizational Unit(s). The RFC2252 definition of the attribute in the 'reference' Idif is:

```
attributetypes: ( 1.3.6.1.4.1.5923.1.1.1.4
    NAME 'eduPersonOrgUnitDN'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY distinguishedNameMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.12' )
```

1.3.6.1.4.1.1466.115.121.1.12 is the OID of the ldap type LDAPTYPE_DN. The equivalent X500 syntax type is *DN* (aka *distinguished name* or *DN String*). Known within ADSI as ADSI_DN_STRING. The OID of the X500 syntax is 2.5.5.1, oMSyntax is 127.

The eduPersonOrgUnitDN attribute appears in the modified Idif file as:

```
dn: CN=eduPersonOrgUnitDN,CN=Schema,CN=Configuration,DC=lse,DC=ac,DC=uk
changetype: add
objectClass: attributeSchema
name: eduPersonOrgUnitDN
description: eduPerson per Internet2 and EDUCAUSE
attributeID: 1.3.6.1.4.1.5923.1.1.1.4
attributeSyntax: 2.5.5.1
oMSyntax: 127
systemOnly: FALSE
isSingleValued:FALSE
```

eduPersonPrimaryAffiliation

This attribute specifies the person's PRIMARY relationship to the institution in broad categories such as student, faculty, staff, alum, etc. The RFC2252 definition of the attribute in the 'reference' Idif is:

```
attributetypes: ( 1.3.6.1.4.1.5923.1.1.1.5
    NAME 'eduPersonPrimaryAffiliation'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY caseIgnoreMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' SINGLE-VALUE )
```

1.3.6.1.4.1.1466.115.121.1.15 is the OID of the ldap type LDAPTYPE_DIRECTORYSTRING. The equivalent X500 syntax type is *Directory String* (aka *Unicode String*) known within ADSI as ADSTYPE_CASE_IGNORE_STRING. The OID of the X500 syntax is 2.5.5.12, oMSyntax is 64.

The eduPersonPrimaryAffiliation attribute appears in the modified Idif file as:

```
dn: CN=eduPersonPrimaryAffiliation,CN=Schema,CN=Configuration,DC=yourPlace,DC=edu
changetype: add
objectClass: attributeSchema
name: eduPersonPrimaryAffiliation
description: eduPerson per Internet2 and EDUCAUSE
attributelD: 1.3.6.1.4.1.5923.1.1.1.5
attributeSyntax: 2.5.5.12
oMSyntax: 64
systemOnly: FALSE
isSingleValued:TRUE
```

eduPersonPrimaryOrgUnitDN

This attribute is the distinguished name (DN) of the directory entry representing the person's primary Organizational Unit(s). The RFC2252 definition of the attribute in the 'reference' Idif is:

```
attributetypes: ( 1.3.6.1.4.1.5923.1.1.1.8
    NAME 'eduPersonPrimaryOrgUnitDN'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY distinguishedNameMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.12' SINGLE-VALUE)
```

1.3.6.1.4.1.1466.115.121.1.12 is the OID of the ldap type LDAPTYPE_DN. The equivalent X500 syntax type is *DN* (aka *distinguished name* or *DN String*). Known within ADSI as ADSI_DN_STRING. The OID of the X500 syntax is 2.5.5.1, oMSyntax is 127.

```
dn: CN=eduPersonPrimaryOrgUnitDN,CN=Schema,CN=Configuration,DC=lse,DC=ac,DC=uk
changetype: add
objectClass: attributeSchema
name: eduPersonPrimaryOrgUnitDN
description: eduPerson per Internet2 and EDUCAUSE
attributelD: 1.3.6.1.4.1.5923.1.1.1.8
```

attributeSyntax: 2.5.5.1

oMSyntax: 127

systemOnly: FALSE

isSingleValued:TRUE

eduPersonPrincipalName

This attribute is defined as the "NetID" of the person for the purposes of inter-institutional authentication. The RFC2252 definition of the attribute in the 'reference' Idif is:

```
attributetypes: ( 1.3.6.1.4.1.5923.1.1.1.6
    NAME 'eduPersonPrincipalName'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY caseIgnoreMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' SINGLE-VALUE )
```

1.3.6.1.4.1.1466.115.121.1.15 is the OID of the ldap type LDAPTYPE_DIRECTORYSTRING. The equivalent X500 syntax type is Directory String (aka *Unicode String*) known within ADSI as ADSTYPE_CASE_IGNORE_STRING. The OID of the X500 syntax is 2.5.5.12, oMSyntax is 64.

The eduPersonPrincipalName attribute appears in the modified Idif file as:

```
dn: CN=eduPersonPrincipalName,CN=Schema,CN=Configuration,DC=lse,DC=ac,DC=uk
changetype: add
objectClass: attributeSchema
name: eduPersonPrincipalName
description: eduPerson per Internet2 and EDUCAUSE
attributeID: 1.3.6.1.4.1.5923.1.1.1.6
attributeSyntax: 2.5.5.12
oMSyntax: 64
systemOnly: FALSE
isSingleValued:TRUE
```

eduPersonScopedAffiliation

Specifies the person's affiliation within a particular security domain in broad categories such as student, faculty, staff, alum, etc. The RFC2252 definition of the attribute in the 'reference' Idif is:

```
attributetypes: ( 1.3.6.1.4.1.5923.1.1.1.9
    NAME 'eduPersonScopedAffiliation'
    DESC 'eduPerson per Internet2 and EDUCAUSE'
    EQUALITY caseIgnoreMatch
    SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )
```

1.3.6.1.4.1.1466.115.121.1.15 is the OID of the ldap type LDAPTYPE_DIRECTORYSTRING. The equivalent X500 syntax type is *Directory String* (aka *Unicode String*) known within ADSI as ADSTYPE_CASE_IGNORE_STRING. The OID of the X500 syntax is 2.5.5.12, oMSyntax is 64.

The eduPersonScopedAffiliation attribute appears in the modified ldif file as:

```
dn: CN=eduPersonScopedAffiliation,CN=Schema,CN=Configuration,DC=lse,DC=ac,DC=uk
changetype: add
objectClass: attributeSchema
name: eduPersonScopedAffiliation
description: eduPerson per Internet2 and EDUCAUSE
attributeID: 1.3.6.1.4.1.5923.1.1.1.9
attributeSyntax: 2.5.5.12
oMSyntax: 64
systemOnly: FALSE
isSingleValued:TRUE
```

4.0 Introducing the eduPerson class into the Active Directory schema

Once your sure your environment is ready, you need to introduce the modifications to your schema. The process, procedures and implications of extending the schema are detailed in the Active Directory Programmers Reference which can be found on Microsoft's [MSDN](#) web site (search for "Extending the Schema"). The procedure below is based on that.

It is strongly recommended that you run through this procedure in an isolated test lab before you make any changes to your live domain and forest. At the very least you should go through this procedure on a standalone test Active Directory server that is not connected to your forest.

It is assumed that you are introducing the schema changes into a Windows 2003 environment. If you are working with a Windows 2000 environment you need to make an additional step.

1. You need to be a member of the Schema Administrators group before you have sufficient permissions to modify the schema. Add yourself to this group.
2. Locate the domain controller that holds the Schema Master FSMO role, you should make the following changes from this server.
3. On the schema master, register schmmgmt.dll (the schema management DLL). Namely:
 - Start, Run, regsrv32 schmmgmt.dll;
 - Then add the schema management snap-in to an MMC: Start, Run, MMC;
 - Then *Console | Add/Remove Snap-in* and add the snap-in called Active Directory Schema.

If you are making the modifications to a Windows 2000 domain you need to enable schema updates to your forest. See Microsoft's Knowledge Base Article 279978, '[Error message: you do not have sufficient access to the Active Directory](#)', for details.

4. Go to a command line and run the following command:

```
ldifde -i -f fileName.ldif -j c:\log.txt -c "DC=X" "DC=yourBaseDN"
```

where:

- -f filename.ldif is the name of the file with the schema changes in it.
- "DC=yourBaseDN" is the distinguished name of your organisation. In the case of the LSE, it is "DC=LSE,DC=AC,DC=UK". Include the quote marks in the command.
- The options are:
 - **-i** Import mode.
 - **-j c:\log** Write output to the c:\log directory. c:\log\ldif.log is created with the results of the import, c:\log\ldif.err will be created should any errors occur. The directory c:\log needs to exist before you run the command.
 - **-c** During parsing of the input file replace occurrences of "DC=X" with "DC=yourBaseDN"

Note that ldifde can be somewhat fussy. Errors along the lines of "syntax error occurred on line x" in seemingly well formed ldif files are common. Things to pay particular attention to are not having any spaces at the end of lines (i.e. a command followed by a CARRIAGE RETURN). If you do find that on importing that the you get errors part way though and you can identify the cause, once you have corrected the problem you can run the ldifde command again adding the -k 'continue on errors' switch.

5. Using the schema management snap-in you created in (3) check the eduPerson class and eduPerson attributes are in place.

6. Remove yourself from the Schema Admins group.

Further Information

For further information on setting up eduPerson on Active Directory, and how to populate the eduPerson attributes, see our associated [LSE Active Directory Updater Description](#) page.

References

Kouti, S. & Seitsonen, M. *Inside Active Directory: a system administrator's guide*. Addison-Wesley 2002.

Arkills, B. *LDAP directories explained: an introduction and analysis*. Addison-Wesley 2003.

Copyright © SECURe Project Team, 2004

Document last updated: 15/04/04

This document is also available at:
<http://www.angel.ac.uk/SECURe/deliverables/documentation/adconfig.html>.