

eduPerson 1.0 Frequently Asked Questions (FAQ)

**DRAFT, Last updated 12-February-2001
Internet2/Educause eduPerson Working Group**

This FAQ is organized into five sections:

- Section 1: Basic Questions
- Section 2: Technical/Implementation Issues
- Section 3: Attribute-specific Issues
- Section 4: Policy Issues
- Section 5: Process Issues

Section 1: Basic Questions

1.1. What is eduPerson?

eduPerson is an auxiliary object class for campus directories designed to facilitate communication among higher education institutions. It consists of a set data elements, or attributes, about individuals within higher education, along with recommendations on the syntax and semantics of the data that may be assigned to those attributes. If widespread agreement and implementation of this object class in campus directories is achieved, a broad and powerful new class of higher education applications can be deployed.

Additional information on eduPerson is available at its home on the web:

<http://www.educause.edu/eduperson>.

1.2. What is a campus directory?

A directory service is based on an integrated institutional data store, and provides authorized users and services on the network with access to information regardless of where or how the original information is stored. This integrated data store is optimized to support lookups of information relating to people, computers, network devices, application preferences and the like. Information contained in the directory is organized in object classes and attributes. Each named attribute holds a specific data element such as phone number, address and so on. An object class is a defined set of attributes relating to a particular type of directory entry. Further details on directory services can be found at <http://middleware.internet2.edu/core/directories.shtml>

1.3. What are the benefits of having a standard object class such as eduPerson? What applications are likely to use eduPerson attributes? What is the value from a business standpoint? What problem does it solve?

Many of the eduPerson attributes are intended to support inter-realm applications such as controlled access to web pages or licensed resources. Many of these inter-realm applications are just beginning to appear; the Directory of Directories for Higher

Education and inter-realm authentication are two examples. Most are related to instructional and research use rather than to internal institutional operations at this point.

If a set of institutions implement the eduPerson object class in their directory schema, a number of inter-institutional applications become possible. For example:

- a. Web pages associated with a course at one campus could be easily and securely opened up to students in another class at another institution. A faculty member teaching Middleware 509 at institution A would be able, through a brief series of pull-down menus, to authorize access to the class web site for students who are enrolled in a linked class, Glueworks 400, at institution B.
- b. An institution (or a group of campuses) could agree to license a database for business students only, and use attributes within eduPerson to implement the access controls.
- c. Scientific researchers could reserve specialized computing resources at distant locations using local services. Approaches such as the Grid (www.gridforum.org) are building advanced scientific computing environments that layer, in part, on top of eduPerson.
- d. A directory of directories within higher education could be created, allowing a user to search effectively and simply through multiple institutional directories in parallel to find public information for a particular person.

The initial applications likely to use eduPerson are regular white pages, Directory of Directories, and Shibboleth/inter-realm authentication and access control. For information about the Directory of Directories and Shibboleth, see:

<http://middleware.internet2.edu/dodhe/>
<http://middleware.internet2.edu/shibboleth/>

1.4. What kinds of attributes are found in eduPerson?

The attributes within this object class are of two types:

- Attributes already given in higher-level or “parent” object classes pre-configured in most commercial directory server products (such as the names, email addresses and security settings found in inetOrgPerson). For these attributes, the work of the eduPerson group has focused on developing higher education appropriate recommendations on syntax, semantics and use to reduce the ambiguity and indeterminacy of the existing documentation and specifications.
- Attributes newly created to facilitate inter-institutional collaborations and applications. This group consists of a few carefully chosen attributes that have clear collaborative benefit within higher education but are not found in available directory schema. For these attributes, eduPerson has defined syntax, semantics and guidance on use.

1.5. Who is involved in this effort? Who is leading it?

Several key universities, such as University of Wisconsin, Georgetown University, University of Washington, and MIT, are lending the technical expertise of their lead IT architects. EDUCAUSE and Internet2 are providing operational support and serving as an interface with other communities of interest within higher education. Keith Hazelton, University of Wisconsin, leads the project team. Other institutions that contributed significantly to the work leading up to version 1.0 include, in alphabetical order: Brown, Carnegie Mellon, Memphis State, Michigan Technological, Penn State, Tufts and the University of California Office of the President.

1.6. Are other groups doing this?

Interoperable directories and their associated applications tend to be “community of interest” activities. Other communities of interest have picked up on this need and are launching similar efforts. For example, the National Institute of Health has constructed an object class for their associated parties. (See <http://www.alw.nih.gov/amgtech/docs/schema/current.html>). We know of no other indigenous activities within higher education seeking to create an inter-institutional community object class.

1.7. How is this distinguished from the Instructional Management Systems (IMS) standards?

The IMS Learner Information Packaging (LIP) Specification defines application-independent structured data models for representing a rich panoply of learner information. The eduPerson object class defines how some subset of the same person information might be represented in an enterprise directory. We are in communication with senior IMS staff and will align our ongoing efforts as appropriate. In version 1.0 of IMS LIP, an XML binding for the core data model is provided. That XML binding might well be the most logical point of integration and mapping between the two efforts.

1.8. Who is adopting/planning to adopt eduPerson?

Many institutions have already agreed to implement the eduPerson object class including University of Wisconsin, Georgetown University, Johns Hopkins University, University of Memphis, University of Michigan, Michigan Technological University, etc. Other institutions have said they will do so when they implement an enterprise-wide directory.

1.9. Why does each attribute have an associated “application utility class” with the value of either “core,” “standard” or “extended?”

The application utility class is intended to suggest the class of applications for which this attribute is appropriate. The eduPerson working group defines these classes as follows:

Core: A minimal white pages application will require values for at least the core application class attributes: cn: common name; sn: surname; and eduPersonOrgDN: the distinguished name of another entry in the directory that represents the person’s home

institution. Name and institution are certainly the minimal useful bits of information about a person in the directory.

Standard: An expanded list of attributes that, as a group and complemented with the core attributes, are adequate to support a full-featured white pages and directory of directories. The standard application utility class includes attributes that would be useful for coarse-grained role-based access control decisions.

Extended: The rest of the defined attributes. They support a larger class of potential applications, but we are in the realm of diminishing returns with these items of information.

Note that there are attributes in some of the parent classes, `orgPerson` and `inetOrgPerson` that are not listed in the `eduPerson` documents at all. These are attributes for which we did not find it possible to make meaningful recommendations. Some institutional directories may populate one or more of these, but that would be outside the scope of `eduPerson`.

Section 2: Technical/Implementation Issues

2.1. Is there any place where we can learn more about the associated technical details?

The LDAP Recipe, “A Recipe for Configuring and Operating LDAP Directories,” is an excellent source of straightforward, sensible and sound advice about configuring and operating an enterprise LDAP directory. It includes significant guidance on how to implement and support the `eduPerson` object class. The current version of the LDAP Recipe can be found at: <http://www.georgetown.edu/giia/internet2/ldap-recipe/>. Of course the `eduPerson 1.0` specification available at <http://www.educause.edu/eduperson/> is the authoritative source for information about the `eduPerson` attributes themselves.

2.2. Where should `eduPerson` fit in directory schema in relation to campus-specific object class?

For implementation issues such as this, please see the LDAP Recipe described in #2.1 above.

2.3. How does one correlate multiple affiliations with multiple organizational units? For example, how does one link a work mailing address with a specific organizational unit?

It is difficult to do this correlation in an LDAP directory. This is one of the shortcomings of LDAP. The working group will continue to discuss and investigate possible solutions and work-arounds.

2.4. What is an OID? Why is it needed? How does one get one?

An OID (object identifier) is a dotted numeric string that is used to uniquely identify an object. Institutions first acquire a private enterprise OID that uniquely identifies them and distinguishes them from all other institutions in the world. OIDs for new objects within that institution are then created by appending additional levels of dotted numeric strings to the right end of the enterprise OID. Typical objects that can be identified using OIDs include attributes in X.500/LDAP-based directories, certificate policies and practice statements, MIBS for network management and encryption algorithms. In particular, as a university defines attributes for local use within directories, it will need to assign institutional OIDs to each of these attributes.

OIDs are only used for “equality-matching”. That is, two objects (e.g. directory attributes or certificate policies) are considered to be the same if and only if they have exactly the same OID. There are no implied navigational or hierarchical capabilities with OIDs (unlike IP addresses, for example). OIDs exist simply to provide an easy to generate unique identifier, and are essentially a response to the fact that in a decentralized world, different organizations may happen to pick the same identical name for what are, in fact, different objects. If each has an OID, there can be no ambiguity as to which is being referenced.

Institutional OIDs can be obtained from a number of sources. Two formal mechanisms include IANA and ANSI.

- a. To get one from IANA, fill out a form at <http://www.iana.org/cgi-bin/enterprise.pl> . There is no fee and turnaround appears relatively quick.
- b. To get one from ANSI, go to http://web.ansi.org/public/services/reg_org.html. There is a one-time fee and turnaround can take several weeks.

Detailed information can be found in the Guide to OIDs under “Identifiers” at: <http://middleware.internet2.edu/>

2.5. Why is it important for a directory to have the rest of the object class hierarchy in place – Person, orgPerson, inetOrgPerson?

Ultimately, attributes are what are important, but object classes (both eduPerson and its parent classes) give additional confidence that the semantics of the attributes are consistent between sites. It’s the “LDAP way.” Also, other applications will have expectations that object classes exist. Note that even with the full hierarchy in place there is no requirement to populate any of the non-mandatory attributes in the parent object classes.

2.6. Why should a site implement the entire eduPerson object class instead of just the attributes that are considered interesting by the site?

It appears that some LDAP implementations cannot chase referrals at the attribute level, but can chase object class referrals. Furthermore, *none* of the newly defined eduPerson

attributes are mandatory. It is up to the particular implementation to specify which of the attributes, if any, will be populated with actual values. The rest are just place-holders that make life easier for those who support interoperation. See 3.5 for further details.

Section 3: Attribute-Specific Issues

3.1. The controlled vocabulary provided for `eduPersonAffiliation` and `eduPersonPrimaryAffiliation` is far too limiting. Why, to cite just one example, isn't "other" one of the permissible values?

The bywords for work on version 1.0 were radical simplification. The values for affiliation in 1.0 are certainly evidence of that. They capture only the broadest role distinctions of people within higher education: "student," "faculty," "staff," "alum" and "affiliate." There are two additional permissible values: "employee" for institutions (notably some in Great Britain) where no distinction is made between faculty and staff; and "member" as an umbrella term for all individuals to whom the institution may choose to grant a "basket" of privileges typically associated with faculty, staff, student or alum status. For example, at the University of Wisconsin, we often grant such a set of privileges, including library check-out rights, to visiting researchers. In other words, we consider them "members in good standing of our academic community." In our directory, such individuals would carry an `eduPersonAffiliation` value of "member." Note that for consistency, anyone carrying one or more of "student," "faculty," "staff" or "alum" roles is also given the "member" value. "Affiliate," on the other hand, is an appropriate value for someone in the directory, such as a vendor representative, not considered "one of us," and thus ineligible for the standard basket of privileges.

Even categories as broad as these have application utility: Being able to limit a directory search to students, for example, seems a desirable feature in a white pages application. A resource provider may be attracted by the potential market represented by authorizing access to a licensed resource to some such broad categories. If so, they might be induced to sign a license agreement on that basis with a university or consortium of universities implementing the `eduPerson` object class.

"Other" did not seem to add any information about an entry. If, for whatever reason, "none of the above" is the appropriate category for a person, the affiliation attributes should simply be left undefined. The effect is the same, and the approach has the benefit of logical consistency.

The complexities of affiliation that lurk beyond these broad terms deserve significant discussion across the stakeholder communities of higher education. By holding that discussion off until versions beyond 1.0, we guarantee that a better cross section of those communities will be present and able to assist us in making the right choices in these murky waters.

3.2. How will we handle additional values for attributes like affiliation? How do we reconcile campus-specific with inter-institutional attributes?

We recommend each campus begin by defining what is needed locally in the campus-specific object class and then build a set of business rules to populate these at the institution level. These additional values would then need to be mapped using the institution's business rules into the coarser-grained values available for attributes at the eduPerson level. Campuses may need to implement metadirectory-like functions to map campus-specific attributes into inter-institutional ones such as the affiliation attributes in eduPerson.

3.3. How does an institution determine what value to assign to an eduPerson attribute?

Like many other attributes within a directory, it is the prerogative of the institution to decide the value to assign for an individual, particularly in those cases where local factors may affect the decisions. eduPerson documents do make some recommendations, especially in the case where there is already strong consensus on the business rules used to assign values. There is an effort underway to ascertain community consensus on values for those attributes native to eduPerson so there will be greater consistency among the institutions from the outset.

3.4. How does this work for multi-campus environments? How are campus-specific attributes vs. system-wide attributes determined?

These are important questions for many institutions and as such, a "Birds of a Feather" session was held during the Internet2 Fall 2000 Member Meeting to identify some of the issues and challenges. A task force is being set up to work through these issues and develop recommendations for addressing them.

3.5. Which attributes must be populated?

The only attributes that must be given values are the two that the X.521(1993) person object class lists as mandatory: cn (common name) and sn (surname). None of the other attributes from person, orgPerson, inetOrgPerson or eduPerson are mandatory. Of course, applications such as white pages will perform more satisfactorily if more of the attributes are populated. The point is that, with the exception of cn and sn, the decision to populate an attribute or not is a local institutional decision.

3.6. What is the intention behind eduPersonPrincipalName (EPPN)?

The EPPN is intended to be an expedient identifier attribute, useful for building some inter-institutional applications. This attribute should prove useful in creating some applications that are based on currently deployed technologies and on code that does not currently use LDAP or require a PKI. This attribute should help to create a framework to create interesting inter-institutional collaborations between sites that use different

technologies. In short, this attribute provides a foundation for yet another abstraction layer.

It is expected that this attribute may become deprecated in some future version of eduPerson. This would occur as LDAP enabled infrastructures and applications become more mature. One metric of this maturity will be the convergence on best practices and their widespread adoption.

3.7. How can eduPersonPrincipalName (EPPN) be useful for authentication and access control in the absence of a mandatory prohibition on reassigning them?

A guiding principle of the eduPerson work has been to leave as much as possible to local institutional discretion. While it makes a lot of sense to be very restrictive in reassigning a given EPPN to different people over time, it is not an issue on which a mandate is practical or enforceable. The problem of reassignment shows up, for example, in the fact that access control lists based on EPPNs will need to be refreshed at some interval. The relying institution or provider needs to take that into account. The person's home institution has a corresponding obligation to make its policies on reassignment clear and public so the relying parties can plan accordingly.

3.8. Why do we need eduPersonNickname? Why is this especially appropriate as a candidate for a self-maintained attribute?

eduPersonNickname is intended to correct a limitation of higher object classes. This attribute can hold a (not necessarily obvious) name by which an individual is accustomed to be hailed. Standard extended matches (Jim, James) will miss many real-world nicknames. Of course nicknames can be stored as one of the values of cn, common name, but this makes it hard to provide a utility to make this attribute self-maintained. Nicknames are best known by the owner (hopefully), so the owner is the ideal "authoritative source" for this attribute. Having a specific attribute for this purpose makes providing a self-maintenance utility a relatively simple matter.

3.9. The uses of eduPersonOrgDN and eduPersonOrgUnitDN are not clear. What attributes are expected to be present in the entry given by that DN? What is the relation to the URL?

EduPersonOrgDN is the distinguished name (DN) of the of the directory entry representing the institution with which the person is associated. With a distinguished name, the client can do an efficient lookup in the institution's directory to find out more about the organization with which the person is associated. The directory entry pointed to by this DN should be represented in the X.521(1993) "organization" object class. The attribute set for organization is defined as follows:

- o (Organization Name, required)

Optional attributes include:

description
localeAttributeSet
postalAttributeSet
telecommunicationsAttributeSet
businessCategory
seeAlso
searchGuide
userPassword

It is recommended the labeledURIObject auxiliary object class be added to the organization object pointed to by this DN, which endows it with a labeledURI attribute. Some directory servers implement this object class by default. For others, the schema may need to be extended using this definition (using the syntax specified by RFC2252):

```
( 1.3.6.1.4.1.250.3.15 NAME 'labeledURIObject' SUP top AUXILIARY
    MAY labeledURI )
```

Similarly, for eduPersonOrgUnitDN.

Each value of eduPersonOrgUnitDN carries the distinguished name (DN) of a directory entry representing one of the person's Organizational Unit(s). The directory entry pointed to by this DN should be represented in the X.521(1993) "organizational unit" object class. Other than having the mandatory attribute of organizationalUnitName, this object class has the same optional attribute set as the organization object class. The same recommendation on including the labeledURIObject auxiliary object class applies to both eduPersonOrgDN and eduPersonOrgUnitDN.

3.10. Where does one store field of study/major? Class level?

These and several other attributes have been suggested for inclusion in the eduPerson object class and will be incorporated in some fashion in future versions. Among other attributes that have been suggested are: school/college name, primary school/college name, FERPA flag, athlete, job classification, and research interests.

3.11 There seem to be a lot of attributes and some apparent overlap between attributes relating to postal addresses. What's the story?

According to X.500(1993), the attribute postalAddress defined in the orgPerson object class "specifies the address information required for the physical postal delivery to an object." It refers to a business address and typically contains multiple lines separated by "\$" that taken together make up a complete delivery address. InetOrgPerson contains a complementary homePostalAddress with similar format. The most important issue, however, is that several other attributes in orgPerson contain elements of a complete business address, and the standard makes clear that these should be matched by a corresponding fragment of the postalAddress attribute. In other words, these attributes must be edited as a set. The affected attributes are: postalCode, postOfficeBox and street.

Section 4: Policy Issues

4.1. There are lots of attributes that my campus wants to keep track of but which do not appear in eduPerson. Why?

The goals of implementing eduPerson are to promote directories in general and to create mechanisms to support inter-institutional collaborations in particular. There are many intra-institutional attributes (such as parking permit number, or food service billing code) that might warrant inclusion in a local schema. It is anticipated that, for example, Georgetown University will develop a georgetownEduPerson object class to capture these attributes.

4.2. There is a lot of common institutional data within the object class. Where does it come from? How is it populated from existing multiple sources?

Very few data items have the enterprise directory as their ultimate authoritative source; most information is fed to the directory from legacy systems across campus. The directory in turn makes that information available to other authorized enterprise systems, LAN administrative systems, and desktop clients. The primary role of a directory is to organize and manage the sharing of data, not to create it. Whether the directory becomes the system of record for some attributes is a matter of institutional policy. Every campus has a different set of sources for the data and different policies/practices for data administration that dictate how the directory is populated.

4.3. Does eduPerson provide a unique identifier for persons in the directory? What are the issues around persistent, globally unique identifiers.

eduPerson does not mandate a unique permanent identifier. This would be difficult for regulatory and marketplace reasons and it is not required for the inter-institutional authentication we are pursuing. eduPersonPrincipalName is a unique identifier, but decisions on populating it or not and on its persistence are local institutional policy matters. There are certainly privacy concerns with anything that might serve as a persistent, globally unique identifier. That is why some institutions will opt not to populate this attribute, or may populate it with relatively short-lived identifiers.

4.4. What is private/public? Who should have access to read and/or write the various attributes in eduPerson?

Many of these attributes raise profound questions about privacy. However, this is not first and foremost an eduPerson issue, but rather a local policy issue. With the inter-institutional focus of eduPerson 1.0, the working assumption is that FERPA-protected attributes for a given person will simply not be accessible via eduPerson-defined attributes. Each institution faces this issue when it plans campus directory deployment. FERPA guidelines provide considerable guidance, where students are concerned, on some of these matters, but FERPA is generally interpreted on a per-campus basis. It may

be useful for a university to consider three rough categories of access – personal access (readable only by the individual), campus community access, and external world access. Discussions of privacy issues are sure to be a prominent feature of ongoing work on eduPerson beyond 1.0.

4.5. Is there an audit trail for the use of data in the directory?

Auditing processes are a local policy and directory implementation issue. eduPerson per se does not address these issues.

Section 5: Process Issues

5.1. How can I stay apprised of the further development of eduPerson? How can I participate in shaping future versions?

Consult the web site at <http://www.educause.edu/eduperson> and the announcements posted to Internet2 and EDUCAUSE mailing lists. If you have any comments on the existing version or suggestions for the future, by all means please send them to eduperson-comments@internet2.edu.

5.2. What does it mean that it is version 1.0?

It is anticipated that the eduPerson object class will evolve rather rapidly over the next few years as we gain more experience in directory operations, inter-institutional electronic sharing, and middleware in general. Updates to eduPerson will be approved and promulgated as needed with all due respect for backward compatibility. That is to say, changes will come in the form of additions and expansions to the object class and the allowed values for certain attributes. The firm commitment is to avoid altering or tinkering with existing features and definitions to the maximum possible extent.