XUUDB MANUAL

UNICORE Team

Document Version:	1.0.1
Component Version:	1.3.2-3
Date:	14 12 2011

This work is co-funded by the EC EMI project under the FP7 Collaborative Projects Grant Agreement Nr. INFSO-RI-261611.

Contents

The XUUDB server is Attribute Source implementation which can be used by UNICORE servers. It is used to map user credentials (an X509 certificate or X500 distinguished name) to authorisation and incarnation attributes.

For more information about UNICORE visit http://www.unicore.eu.

1 Overview

The UNICORE XUUDB is is used to map user credentials (such as an X.509 certificate or X.500 distinguished name) to a set of attributes. The attributes are: a list of Unix logins (aka XLogins), a role and a list of projects (UNIX groups).

The XUUDB stores either X.509 certificates (*normal* mode) or distinguished names (*dn* mode), see Section 4.

The XUUDB offers two web services, one for querying, and one for administration of the user database.

Multiple Grid sites can share the XUUDB, even if the attributes are different per Grid site. Grid sites are grouped by the so-called GCID (grid component ID).

Both admin and client access to the XUUDB is protected by client-authenticated SSL.

IMPORTANT NOTE ON PATHS
XUUDB is distributed either as an platform independent and portable bundle (as a part of UNICORE quickstart package) or as an installable, platform dependent package such as RPM.
Depending on the installation package used paths are different. If installing using distribution-specific package the following path prefixes are used:

```
CONF=/etc/unicore/xuudb
BIN=/usr/sbin
ADMIN=/usr/sbin/unicore-xuudb-admin
LOG=/var/log/unicore/xuudb
```

If installing using portable bundle all XUUDB's files are installed under a single directory. Path prefixes used then are as follows, where INST is a directory where XUUDB was installed:

```
CONF=INST/conf
BIN=INST/bin
ADMIN=BIN/admin.sh
LOG=INST/log
```

The above variables (CONF, BIN, ADMIN and LOG) are used throughout the rest of this manual.

2 Installation

UNICORE XUUDB is distributed in the following formats:

1. As a part of platform independent installation bundle called UNICORE Quickstart. UNICORE Quickstart is provided in two forms: one with graphical installer and one with a command line installer.
2. As a RPM available for Scientific Linux 5 platform. This RPM is tested on Scientific Linux, but should work without any problems with recent version of Centos (e.g. Centos 5.5), Fedora (e.g. Fedora 13, 14) and other Red Hat derivatives.

In both cases installation of XUUDB installs both XUUDB Server and XUUDB admin client.

2.1 Installation from Quickstart package

Download the quickstart bundle from the UNICORE project website.

If you use graphical installer follow the on screen instructions and do not forget to check click the XUUDB checkbox when prompted.

If you use text installer then for generic installation instruction review the README file available after extracting the Quickstart bundle. You don't have to change any defaults as XUUDB installation is enabled by default.

In both cases you can preconfigure the XUUDB server during installation (of course this can be done also later) by choosing the XUUDB server host, port and mode.

2.2 Installation from RPM package (RedHat distributions)

Download RPM package from UNICORE download site and install it using rpm command as root user:

```
$> rpm -i unicore-xuudb-VERSION.noarch.rpm
```

You can also (what is suggested) use yum to install (and subsequently update) XUUDB.

The yum installation may be performed as follows (note that first command is needed only if you have not yet installed the EMI yum repository):

```
$> wget --no-check-certificate \
   https://twiki.cern.ch/twiki/pub/EMI/EMI-1/emi_1.repo \
   -O /etc/yum.repos.d/emi-1.repo
$> yum install unicore-xuudb
```

3 The XUUDB server

3.1 Configuration

By default, the configuration is defined in the file `CONF/xuudb_server.conf`. Review the settings. To use a different config file, edit the start script, or use `--start <config_file>` as command line arguments when starting.

3.2 Starting the XUUDB server

Start the server with

```
BIN/start.sh
```

In case if XUUDB was installed with binary package use:

```
/etc/init.d/unicore-xuudb start
```

3.3 Stopping the server

Stop the server with

```
BIN/stop.sh
```

This sends a TERM signal to the XUUDB process. Please do not use `kill -9` to stop XUUDB, to avoid corrupting the database.

In case if XUUDB was installed with binary package use:

```
/etc/init.d/unicore-xuudb stop
```

3.4 Logging

The logging settings are controlled in `CONF/logging.properties`, and changes to this file take effect at runtime. By default log files are found in the LOG directory.

4 Normal mode vs. DN mode

The XUUDB Database supports two mode, *normal* and *dn*, controlled by a setting in the server configuration file `conf/xuudb_server.conf`. Running in normal mode uses the whole X.509 PEM encoded certificate of the users to perform a match. This particularly means, if a user certificate is not valid any more the user has to be readded with a new certificate. When running in dn mode, only the DN of the x509 certificate is stored in the database, so a user can access UNICORE with a new certificate, if the DN is equal to the old one.

The admin tool has a command `adddn` which will add an XUUDB entry using just the DN.

Note
When extracting the DN from a certificate file using OpenSSL, make sure to use the `RFC2253` option, for example:

```
openssl x509 -in demouser.pem -noout -subject -nameopt RFC2253
```

5 Clients for the XUUDB

5.1 Admin client (see also Section 6.2)

The admin client is used to add, remove, list and update certificates and user information. It is configured in the file `CONF/xuudb_client.conf`. To use the client, do

```
ADMIN <command> <options>
```

You can get detailed usage info by calling the admin script without any options. As it was noted above the actual utility path is dependent on how XUUDB was installed: it is either `/usr/sbin/unicore-xuudb-admin` or `INST/bin/admin.sh`.

Note
to switch on the confirmation message asked by the `add` command, edit the admin.sh script, so that the `xuudb.batch` property is set to `false`.

5.2 UNICORE 6

UNICORE 6 includes the XUUDB as default authorisation component.

5.3 UNICORE Rich client plugin

There is a plugin for the UNICORE Rich client that allows for editing the XUUDB remotely.

6 Security

6.1 Basic SSL

Client-authenticated SSL is used to protect the XUUDB. Therefore you will need certificates for the XUUDB and all Grid components that want to talk to the XUUDB. In general the XNJS and the XUUDB-admin need to connect to the XUUDB-server. To grant them access, you have to put the following certificates as trusted certs into the XUUDB's server truststore:

- CA certificate(s) of the UNICORE/X server(s) that query the XUUDB
- CA certificate(s) of the XUUDB-admin user certificate(s)

6.2 Administrative access

The XUUDB provides two web service interfaces, one for querying the XUUDB (i.e. mapping certificates or DNs to user information), and a second one for administration of the XUUDB (i.e. adding and editing entries). All access to the XUUDB (including the administration utility!) is through these web services. To prevent arbitrary Grid users from modifying the XUUDB, the administrative interface has to be protected.

Starting with UNICORE 6.3, the access control mechanism of the administrative interface has been simplified. An ACL file is used, which is a text file containing the distinguished names of the administrators. At least it has to contain the DN of the certificate used by the administration utility.

An example ACL file (`CONF/xuudb.acl`) is part of the distribution, which contains the DN of the default XUUDB server certificate.

The ACL file can be changed at runtime to easily add or remove administrators.

To change the location of the ACL file, edit the server configuration and set a configuration parameter, e.g.:

```
xuudb_acl_file=CONF/xuudb.acl
```

The ACL entries are expected in RFC2253 format. To get the name from a certificate in the correct format using openssl, you can use the following OpenSSL command:

```
$> openssl x509 -in demouser.pem -noout -subject -nameopt RFC2253
```

7 The admin client

The admin client is used to edit the XUUDB, using a web service interface.

7.1 Commands

```
add         <gcID>  <pemfile>  <xlogin> <role>
            [project1[,project2[,...]]]
adddn       <gcID>  <DN>  <xlogin> <role> [project1[,project2[,...]]]
remove      ALL|gcID=x|pemfile=file|dn="DN"|role=x|xlogin=x|project=x
list        gcID=x|pemfile=file|dn="DN"|role=x|xlogin=x|project=x
update      <gcID> <pemfile or DN> gcID=x|pemfile=file|dn="dn"|role=x|
            xlogin=x|project=x
export      <csv-file> [overwrite]
import      <csv-file> [clearDB]
check-cert  <gcid> <pemfile>
check-dn    <gcid> <dn>
--init      creates a new admin client configuration in ./conf
            if not existing
```

Note
when the server runs in dn mode you can use *dn=* parameter for remove, list and update

Common options:

gcID
The so-called "grid component ID" is used to group entries, and must match the setting in the UNICORE/X configuration file `uas.config`. For example if you have two systems with different user name mappings, you can handle both with a single XUUDB, since you can store two user name mappings for each certificate, by choosing a different gcID for both systems. When updating xuudb entries, the special gcid * can be used as wildcard for updating user entries on all systems.

pemfile
A file containing the public key in PEM format

DN
The distinguished name of a user

xlogin
xlogins (from UNIX login) are used for incarnation. Grid user's request which results in invocation of operations on a target system (usually through BSS) must be mapped to a local UNIX user. This attribute specifies the XLogins which are valid for the user. The first one is also used as a default one, if user does not request a particular one. Multiple logins can be specified using a `:`

project
: Defines a primary group UNIX group for a user. If it is undefined then a default group for the XLogin is used.

role
: The usual roles in UNICORE are `user` for a normal user, and `admin` for an administrator. Custom roles can be added, and can be assigned permissions in the UNICORE/X security policy file.

7.2 Adding entries using `add` or (in DN mode) `adddn`

Example using a pem file:

```
$> ADMIN add DEMO-SITE /path/to/usercert.pem userlogin user
```

Example using the DN (works only if server runs in DN mode):

```
$> ADMIN adddn DEMO-SITE "CN=John Doe, O=Test Inc" userlogin user
```

7.3 Checking the content

Apart from `list`, you can use the `check-cert` and `check-dn` commands to see what the XUUDB contains for a certain certificate or DN.

7.4 Removing entries

HINT: before removing you can check with the list command which takes the same parameters, that your are removing the correct entries.

To remove all entries from xuudb (you will have to confirm this)

```
$> ADMIN remove ALL
```

To remove some entries, you have to specify attributes.

To remove a user with cert cert.pem at gcid MYSITE:

```
$> ADMIN remove gcid=id001 pemfile=/path/cert.pem
```

To remove all users from gcid OLDMACHINE:

```
$> ADMIN remove gcid=OLDMACHINE
```

To remove a user with xlogin jdoe from all gcids:

```
$> ADMIN remove xlogin=jdoe
```

etc...

7.5 Exporting/importing

The export command creates a csv file, which will contain the complete XUUDB database:

```
$> ADMIN export uudb.csv
```

If the file already exists, the export tool will complain. To override this, please specify the `overwrite` option, e.g.

```
$> ADMIN export uudb.csv overwrite
```

The import command takes the a csv file (as generated by `export`) and imports all entries. Already existing entries will not be changed. To do updates, execute `admin.sh remove ALL` before, or specify `clearDB` as a second argument

```
$> ADMIN import uudb.csv
```

7.6 Updating entries

The `update` command can be used to modify existing entries, for example to replace the certificate or the login. For example,

```
$> ADMIN update DEMO-SITE certs/demouser.pem xlogin=jb007
```

would update the entry identified by the gcID *DEMO-SITE* and the given pem file, and assign a new xlogin. If you want to update a user's entry on all the sites, you would use

```
$> ADMIN update \* certs/demouser.pem xlogin=jb007
```

Note that the wildcard * is a special character for the shell and needs to be escaped with a backslash.