

SECURE Documentation: LSE Active Directory Updater Description

Most of the LSE's central data held is held in Oracle databases, yet it was recognised that it would also be useful to publish some of this data into Microsoft Active Directory (AD). For example, if staff telephone numbers were published in AD, they would be displayed automatically in the address book of Microsoft Outlook, the school's standard email client. The eduPerson schema presented what is essentially the same requirement.

The main challenge was not just to publish data to AD but to keep it consistent with the database thereafter. The LSE's Business Systems and Services department decided to develop a piece of middleware in-house to solve this problem. The Sun Java 2 platform was chosen to implement this middleware as it included the Java Database Connectivity API (JDBC) and the Java Naming & Directory Interface API (JNDI) besides offering the benefit of platform independence.

At the core of the system is a Java program which processes a queue of required update operations. The queue is implemented as a table in the database and is added to via triggers each time relevant data is modified in the database. Each entry in this queue includes an identifier for the relevant user and the name of the Active Directory attribute to be updated.

The system also includes an abstract UpdaterMapping class which can be extended to provide the code to retrieve from the database (and transform where necessary) the current value(s) a specific attribute of a User in Active Directory should have.

When processing an entry from the queue the program looks for an UpdaterMapping for the specified AD attribute. It then uses that to retrieve the current values(s) and then writes them to Active Directory using the Java JNDI API.

The design is aimed at making it as easy as possible to add and maintain mappings between the database and the directory. For each mapping only two units of code are required - a database trigger and an UpdaterMapping subclass.

One point to note with this design is that the update to Active Directory does not immediately follow the update to the database. The Java program processes the queue on a regular basis, currently configured to every 120 seconds. This was deemed adequate given that Active Directory can take more time itself to replicate a new value to all its servers after an update is made.

The possibility of making the updates to Active Directory directly from the database trigger was investigated however this approach was decided against for the following reasons. First, there was no guarantee over how long such an operation could take, so this could have impacted upon the performance of the database as perceived by the user. Secondly, it was not possible to

include the directory update as part of the transaction thus there was no real advantage.

Copyright © SECURe Project Team, 2004

Document last updated: April 2004

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