

Active Directory User Manager

Synchronisation der Benutzerkonten in heterogenen
Netzwerken

Christoph Thien
Daniel Rohde

Humboldt Universität zu Berlin
Computer- und Medienservice

Inhaltsverzeichnis

1	Einführung	3
2	Einsatzumgebung und Anforderungsprofil	4
3	ADUM Redirector	5
3.1	Das ADUM-Protokoll	5
3.2	Der Datenaustausch aus Sicht des UNIX-Clients	7
3.3	Die Konfiguration	7
3.4	Der Weg der Kontodaten	9
4	ADUM Server	10
4.1	Die Konfiguration	10
4.2	Der Weg der Kontodaten – Teil 2	12
5	Die Kommandozeilenoptionen	13
6	Die Entwicklungsumgebung	14

1 Einführung

Einer der vielen Dienste des Computer- und Medienservices (CMS) umfaßt die Bereitstellung von Konten für Studenten und Angestellte der Humboldt-Universität zu Berlin. Jedes Konto enthält Informationen über den Besitzer und die Zugangsberechtigungen für verschiedene Dienste und Ressourcen innerhalb des Intranets. Es ist das Anliegen des CMS, jedem Benutzer genau ein Konto zuzuweisen. Mit diesem einem Konto soll der Zugriff auf alle Dienste unabhängig von der Plattform geregelt werden. Ausgehend von heterogenen Netzwerken, dem gemischten Betrieb von UNIX- und Windows-basierenden Plattformen, müssen die Konten zwischen diesen Plattformen synchronisiert werden. Dies kann aufgrund der hohen Anzahl der Konten nur automatisiert erfolgen. Dieser Prozeß soll im folgenden beschrieben werden.

Zu Beginn der Entwicklungsarbeiten im Jahre 2001 gab es keine ausgereiften Lösungen für die automatische Synchronisation der Benutzerkonten. Auch heute verzichten viele Administratoren von kleineren gemischten Netzwerken auf eine automatische Synchronisation. Benutzer haben dann meist verschiedene Konten für Windows und UNIX oder müssen sich verschiedene Passwörter merken. Das manuelle Eintragen und Aktualisieren von Konten in der jeweils anderen Umgebung ist zeitaufwändig und fehleranfällig.

Um diese Probleme zu lösen wurde das Programmpaket *Active Directory User Manager (ADUM)* entwickelt. Es enthält zwei Hauptprogramme für die Synchronisation der Benutzerkonten (*Redirector* und *Server*) und mehrere Testprogramme um die Funktionalität zu überprüfen und Programmfehler zu finden. Letztere sind meist nur für die Entwickler interessant. Die hier vorgestellte Lösung ist einfach einzurichten. Gleichzeitig werden sicherheitsrelevante Aspekte nicht vernachlässigt.

2 Einsatzumgebung und Anforderungsprofil

Die Verwaltung der Benutzerkonten für alle Plattformen muß an zentraler Stelle geschehen. Die Abteilung Benutzerverwaltung des CMS benutzt traditionell eine UNIX-basierte Umgebung. Die Windows basierten Domänen wurden später dem Intranet hinzugefügt. Es sollen folgende Aufgaben an zentraler Stelle bearbeitet werden:

- Konto erstellen
- Konto aktivieren und deaktivieren
- Konto löschen
- Passwort ändern
- Passwort zurücksetzen
- Passwort vergleichen

Weiterführende Modifikation der Konten, die Vergabe von Zugriffsrechten, übernehmen die Administratoren, die die Dienste bereitstellen.

Auf der UNIX-Seite sind nur minimale Änderungen notwendig. Die Konten werden weiterhin auf UNIX erstellt und geändert. Die Kontodaten werden an den *ADUM Redirector* gesendet. Dieser wertet die Daten aus und bestimmt anhand seiner Konfiguration die Windows-Domäne, in welcher das Konto erstellt oder bearbeitet wird. Die Kontodaten werden dazu vom *ADUM Redirector* an den entsprechenden *ADUM Server* dieser Domäne weitergeleitet. Der *ADUM Server* führt den Auftrag aus und erstellt oder bearbeitet das Konto in der Domäne.

UNIX —> *ADUM Redirector* —> *ADUM Server* —> Windows Domäne

Jeder Benutzer hat jetzt ein zentrales Konto und Passwort unabhängig von der verwendeten Plattform. Mit gleichem Kontonamen und Passwort kann der Nutzer auf UNIX- und Windows-Dienste zugreifen. Die Kontodaten werden auf beiden Plattformen gespeichert und bei Änderungen synchronisiert. Dieser Prozeß bleibt dem Benutzer verborgen.

3 ADUM Redirector

Diese Komponente hat die Aufgabe, die Kontodaten von der UNIX-Seite entgegenzunehmen und an die entsprechende Windows Domäne weiterzuleiten. *ADUM Redirector* öffnet einen TCP/IP Sockel und arbeitet als TCP/IP Server. Die Kontodaten werden auf der UNIX-Seite gemäß dem einfachen ADUM-Protokoll formatiert und an den *Redirector* gesendet.

3.1 Das ADUM-Protokoll

Das ADUM-Protokoll folgt grundlegend diesem Aufbau:

<Länge>|<Anweisung>|<Parameter>|...

- <Länge> Länge der folgenden Daten in Bytes einschließlich des ersten und letzten „|“
- <Anweisung> Enthält eine Zeichenkette, die die Art der Daten und deren Behandlung spezifiziert, z.B ACCINSERT, PWDCHANGE, ...
- <Parameter> Die formatierten Kontodaten. Die Art und Anzahl der Parameter hängt von <Anweisung> ab.

Die Daten bestehen aus 8-Bit Zeichen. Die Kodierung der Zeichen ist MS Windows Codepage 1252.

die aktuelle ADUM-Version (4.3.1) enthält folgende Anweisungen:

Anweisung	Parameter	Beschreibung
ACCINSERT	%1 %2 %3 %4 %5	erstellt ein neues Konto für einen Benutzer Kontoname, AD-Eigenschaft <code>sAMAccountName</code> , kann höchstens 20 Zeichen lang sein Nachname des Benutzers Vorname des Benutzers Klartextpassword für das Konto OKZ (= Identifikation), erlaubt die weitere Unterscheidung der Benutzer bei der Erstellung des Kontos in unterschiedlichen Containern, Eintrag in unterschiedliche Gruppen usw. – siehe auch Konfiguration des ADUM Servers

Anweisung	Parameter	Beschreibung
	%6	voller Domänenname in DNS-Notation für die Weiterleitung an den zuständigen ADUM Server – siehe auch Konfiguration des ADUM Redirector
PWDCHANGE	%1 %2	das Passwort setzen Kontoname zu setzendes Passwort (Klartext)
PWDVERIFY	%1 %2	das Passwort überprüfen Kontoname zu prüfendes Passwort (Klartext)
ACCACTIVATE	%1	das Konto aktivieren Kontoname
ACCDEACTIVATE	%1	das Konto deaktivieren Kontoname
ACCDELETE	%1	das Konto löschen Kontoname
PWDRESET	%1	setzt für das Konto das Flag: Benutzer muß Passwort bei der nächsten Anmeldung ändern Kontoname

Folgende Beispiele sind gemäß dem ADUM-Protoll formatiert:

```
70|ACCINSERT|julia|Meyer|Julia|B3rserker|5A3x|abctest.xp2k.hu-berlin.de|
28|PWDCHANGE|julia|S4r55skwod|
28|PWDVERIFY|julia|S4r55skwod|
21|ACCDEACTIVATE|julia|
```

Zusätzlich befindet sich in der Quellcodedatei `adum_redirector_main.cpp` die Beschreibung der aktuellen Version des ADUM Protokolls einschließlich aller Anweisungen und ihrer Parameter.

3.2 Der Datenaustausch aus Sicht des UNIX-Clients

1. Client-Sockel öffnen und mit dem *Redirector* verbinden
2. Daten auf den Sockel schreiben
3. Antwort des Servers vom Sockel lesen
4. Verbindung schließen

Die Antwort des *Redirectors* ist ebenfalls gemäß des ADUM Protokolls formatiert. Nach der Länge der Antwort in Bytes folgt entweder |SUCCESS| oder eine Fehlermeldung (z. B. |REDIRECTOR:ERROR:0x00005012| – in diesem Fall wurde das angegebene Konto nicht gefunden). Der *Redirector* schreibt Fehlercode und Fehlermeldung in die Log-Datei(en). Dem Client wird jedoch nur der Fehlercode übermittelt.

Da hier sicherheitsrelevante Daten übertragen werden, sollte eine SSL verschlüsselte Verbindung genutzt werden. Im ADUM Paket ist noch keine SSL Schnittstelle implementiert. Um trotzdem SSL benutzen zu können, muß auf dem Rechner, auf dem der *Redirector* läuft, ein SSL Wrapper (z. B. STUNNEL) installiert werden. Der Wrapper agiert als SSL-Server und normaler Client. Die Daten werden an den SSL-Port gesendet, der Wrapper gibt sie unverschlüsselt auf den Port des *Redirectors* aus. Diese Lösung erlaubt die Verwendung von Zertifikaten.

3.3 Die Konfiguration

Das Verhalten des *Redirector* kann vielfältig konfiguriert werden. Alle Anpassungen werden in einer Textdatei spezifiziert. Diese wird während des Starts des *Redirector* gelesen und verarbeitet. Standardmäßig ist der Name der Konfigurationsdatei ADUMRD.INI und befindet sich im gleichen Verzeichnis wie die ausführbare Datei ADUMRD.EXE. Der *Redirector* schreibt keine Daten in die Registrierdatenbank von Windows.

Das Format der Konfigurationsdatei entspricht dem der Windows .INI Dateien. Die mannigfaltigen Konfigurationsdaten werden in Sektionen angeordnet. Es existiert genau eine Sektion mit der Kennung [redirector]. In dieser Sektion wird das globale Verhalten des *Redirector* bestimmt. Folgende Parameter können bestimmt werden:

Parameter	Beschreibung
port=	der Server-Port für eingehende Daten (Standard: port=800)
gc=	Name des Globalen-Katalog-Servers in voller DNS-Notation (z. B. gc=GC://hu-berlin.de)

Parameter	Beschreibung
client=	Zugriff für diesen Client aktivieren, die IP ist in Standardnotation (z. B. client=10.20.30.40) anzugeben. Es können mehrere Clients angegeben werden, alle anderen Clients werden standardmäßig abgewiesen. Existiert kein client= Eintrag, so werden alle Verbindungen akzeptiert.
logfile=	Pfad und Dateiname einer Logdatei (wird erstellt falls nicht vorhanden, sonst werden die Protokolle angefügt). Es können mehrere Logdateien angegeben werden. Mehrere gleiche Einträge führen zu unvorhersagbaren Ereignissen.
maxlogsize=	bestimmt die maximale Größe der Logdatei(en) in Bytes – wird die angegebene Größe überschritten, wird das Log umbenannt in <logname>-xxxx.log, wobei xxxx eine 4-stellige Nummer beginnend bei 0001 ist – der erste freie Dateiname wird verwendet – der voreingestellte Wert beträgt maxlogsize=1048576
maxscreenlog=	bestimmt die Anzahl der Einträge (Zeilen) im Logfenster im GUI-Modus – nach ca. 6000 angenommenen Verbindungen belegt das Logfenster ca. 15 MB Hauptspeicher – der voreingestellte Wert beträgt maxscreenlog=1000 (reicht für ca. 60 Verbindungen)

Der allgemeinen Konfiguration in der Sektion [redirector] folgen beliebig viele Sektionen mit der Bezeichnung [domain]. Hier wird konfiguriert, an welchen *ADUM Server* die Kontodaten zur Bearbeitung weitergeleitet werden:

Parameter	Beschreibung
id=	ein interner Identifikationsstring, nur für die Anzeige der Konfiguration im GUI-Modus relevant
dn=	Domänenname in DNS-Notation (z. B. dn=abc.xyz.huberlin.de)
server=	Die IP-Adresse mit Port eines <i>ADUM Servers</i> für diese Domäne in Standardnotation (z. B. server=10.20.30.40:800). Es können mehrere <i>ADUM Server</i> angegeben werden. Falls ein Server unerreichbar ist, wird ein anderer verwendet.

In der Konfigurationsdatei ADUMRD.INI sind alle Konfigurationsmöglichkeiten nochmals beschrieben. Generell wird zwischen Groß- und Kleinschreibung unterschieden. Die ist besonders bei Domännennamen in DNS-Notation zu beachten.

3.4 Der Weg der Kontodaten

Anhand der Konfiguration des *Redirector* und des Inhalts der von der Client-Seite übermittelten Kontodaten kann der *Redirector* einen *ADUM Server* finden, welchem die Kontodaten gesendet werden. Dort werden die Kontodaten entsprechend der Konfiguration des *ADUM Servers* bearbeitet.

Für den Fall, daß ein neues Konto erstellt werden soll, wird die Domäne im ADUM-Protokoll angegeben (Parameter %6 in der Anweisung ACCINSERT). Der Client des *Redirector* muß also bereits für jedes neue Konto die Information über die Domäne mitliefern.

In allen anderen Fällen existiert das Konto bereits. Der *Redirector* erhält den Kontonamen vom Client und sucht im globalen Katalog die Domäne, wo dieses Konto eingetragen ist.

Es ergeben sich folgende Einschränkungen: ein Kontoname darf nur einmal vergeben werden. Auch wenn unterschiedliche Domänen verwendet werden, die keine Verbindung miteinander besitzen. Ansonsten kann der *Redirector* nicht eindeutig entscheiden, welches Konto bearbeitet wird. Allerdings prüft der *Redirector* nicht, ob das Konto einmalig ist. Das erste Konto, welches im globalen Katalog gefunden wird, wird als einzigstes angesehen und die Daten an den betreffenden *ADUM Server* übermittelt.

Die Client-Seite, die die Kontonamen generiert, muß garantieren, daß ein Kontoname nur einmal vergeben wird!

4 ADUM Server

Der *Server* erhält die Daten vom *ADUM Redirector*. Die Kommunikation der beiden Anwendungen wird verschlüsselt. Dafür wird die Windows-CryptoAPI verwendet. Der Schlüsselaustausch verwendet den Diffie-Hellman-Algorithmus.

Der Prozeß des *Servers* muß auf einem Rechner in einer Domäne laufen, deren Konten er verwalten soll. Der *Server* muß nicht, aber kann auf einem Domänen-Controller ausgeführt werden. Wichtig sind Rechte zum Suchen, Anlegen und Bearbeiten von Konten im Active Directory. Der *Server* sollte in einem Kontext mit entsprechenden Rechten gestartet werden.

4.1 Die Konfiguration

Das Verhalten des *Servers* kann vielfältig konfiguriert werden. Wie beim *ADUM Redirector* werden alle Anpassungen in einer Textdatei spezifiziert. Diese wird während des Starts des *Servers* gelesen und verarbeitet. Standardmäßig ist der Name der Konfigurationsdatei ADUMSRV.INI und befindet sich im gleichen Verzeichnis wie die ausführbare Datei ADUMSRV.EXE. Der *Server* schreibt keine Daten in die Registrierdatenbank von Windows.

Das Format der Konfigurationsdatei entspricht dem der Windows .INI Dateien. Die mannigfaltigen Konfigurationsdaten werden in Sektionen angeordnet. Es existiert genau eine Sektion mit der Kennung [server]. In dieser Sektion wird das globale Verhalten des *Servers* bestimmt. Folgende Parameter können bestimmt werden:

Parameter	Beschreibung
port=	der Server-Port für eingehende Daten (Standard: port=800)
client=	Zugriff für diesen Client aktivieren, die IP in Standardnotation (z. B. client=10.20.30.40) angeben. Es können mehrere Clients angegeben werden, alle anderen Clients werden standardmäßig abgewiesen. Existiert kein client= Eintrag, so werden alle Verbindungen akzeptiert.
logfile=	Pfad und Dateiname einer Logdatei (wird erstellt falls nicht vorhanden, sonst werden die Protokolle angefügt). Es können mehrere Logdateien angegeben werden. Mehrere gleiche Einträge führen zu unvorhersagbaren Ereignissen.

Parameter	Beschreibung
maxlogsize=	bestimmt die maximale Größe der Logdatei(en) in Bytes – wird die angegebene Größe überschritten, wird das Log umbenannt in <logname>-xxxx.log, wobei xxxx eine 4-stellige Nummer beginnend bei 0001 ist – der erste freie Dateiname wird verwendet – der voreingestellte Wert beträgt maxlogsize=1048576
maxscreenlog=	bestimmt die Anzahl der Einträge (Zeilen) im Logfenster im GUI-Modus – nach ca. 6000 angenommenen Verbindungen belegt das Logfenster ca. 15 MB Hauptspeicher – der voreingestellte Wert beträgt maxscreenlog=1000 (reicht für ca. 60 Verbindungen)
deletefile=	gibt eine Datei an, die eine Liste der Konten enthält, die gelöscht werden sollen. Das Löschen muß von Hand erfolgen. Es können mehrere Dateien angegeben werden. Mehrere gleiche Einträge führen zu unvorhersagbaren Ereignissen.

Der allgemeinen Konfiguration in der Sektion [server] folgen beliebig viele Sektionen mit der Bezeichnung [entry]. Hier wird bestimmt, wie ein Konto im Verzeichnis erstellt wird. Es betrifft als nur die Anweisung ACCINSERT des ADUM-Protokolls. Alle anderen Server-Anweisungen benötigen keine spezifische Konfiguration.

Das ADUM-Protokoll liefert für jedes neue Konto auch eine OKZ (historisch: Organisationskennzahl der HU-Berlin). Dadurch können die anzulegenden Konten unterschieden werden. Die OKZ-Information ist nicht auf Zahlen beschränkt – sie kann Buchstaben und Sonderzeichen enthalten (nicht jedoch # – denn damit wird in der Konfigurationsdatei der Beginn eines Kommentars bestimmt). Für jede mögliche OKZ sollte eine [entry]-Sektion definiert werden. Folgende Parameter können bestimmt werden:

Parameter	Beschreibung
id=	ein interner Identifikationsstring, nur für die Anzeige der Konfiguration im GUI-Modus relevant
okz=	die OKZ für diese [entry]-Sektion.
group=	LDAP-Pfad der Gruppe, in die der Account eingetragen wird. Es können beliebig viele Gruppen angegeben werden.
loginscript=	Pfad zum Login-Script

Parameter	Beschreibung
setpwd=	Passwort setzen, mögliche Werte: yes no
changepwd=	Benutzer muß das Passwort beim nächsten Logon wechseln. Achtung: diese Eigenschaft kann nicht gleichzeitig mit UF_DONT_EXPIRE_PASSWORD gesetzt sein.
activate=	Konto aktivieren, mögliche Werte: yes no. Konto, die aktiviert werden sollen, müssen auch ein Passwort setzen. Die folgende Kombination führt daher zum Fehler: setpwd=no und activate=yes
uln=	UserLogonName Suffix – wird kein Suffix angegeben, wird der Suffix des DomainControllers aus dem aktuellen Kontext genommen.
homedrive=	setzt die homeDrive Eigenschaft (2-Byte z.B. E: F: G: H: ...) – wichtig: die Eigenschaft wird nur gesetzt, wenn der Laufwerksbuchstabe noch nicht vergeben ist!
homedir=	setzt die homeDirectory Eigenschaft – an das angegebene Homeverzeichnis wird noch der Kontoname (vom ADUM-Protokoll übermittelt) angehängen. Die Homeverzeichnisse werden nicht angelegt

In der Konfigurationsdatei ADUMSRV.INI sind alle Konfigurationsmöglichkeiten nochmals beschrieben. Generell wird zwischen Groß- und Kleinschreibung unterschieden. Diese ist besonders bei LDAP-Pfaden zu beachten.

4.2 Der Weg der Kontodaten – Teil 2

Für den Fall das ein neues Konto erstellt werden soll, kann der Administrator diesen Prozeß anpassen. So kann der Ort des Kontos im Verzeichnis bestimmt und initiale Eigenschaften gesetzt werden. Die entscheidende Information für den *Server* ist der Identifikationsstring, OKZ genannt. Dieser wird mit den Kontodaten übermittelt und von der UNIX-Seite bereitgestellt. Der *Server* sucht anhand der OKZ das entsprechende Profil und erstellt das Konto mit den konfigurierten Eigenschaften. Damit ist auch im Active Directory eine vielschichtige Organisation der Konten möglich.

Für alle neuen Kontos entspricht die Eigenschaft `commonName` (CN=...) immer dem Kontonamen (`sAMAccountName`). Weiterhin werden die Eigenschaften `displayName` und `description` immer auf »Vorname Nachname« gesetzt. Die Eigenschaften `given-`

Name und sn werden auf »Vorname« und »Nachname« gesetzt. Dieses Verhalten kann z. Z. nur im Quellcode geändert werden.

In allen anderen Fällen der Bearbeitung von Konten im Active Directory durch den *Server* ist keine Konfiguration nötig. Soll der *Server* das Passwort eines Kontos überprüfen, so verbindet er sich mit dem Konto unter Verwendung des angegebenen Kontonamens und des mitgelieferten Passworts (anstelle der des Kontextes in dem der Prozeß ausgeführt wird). War die Verbindung erfolgreich, so stimmt das Passwort des Kontos mit dem übermittelten Passwort überein. Dann ist der Rückgabewert auch [SUCCESS]. Andernfalls wird dem *ADUM Redirector* ein Fehler übermittelt.

5 Die Kommandozeilenoptionen

ADUM Redirector und *ADUM Server* besitzen mehrere Möglichkeiten der Ausgabe von Meldungen zur Laufzeit. Eine Schnittstelle kann mittels eines Kommandozeilenparameters ausgewählt und konfiguriert werden.

Option	Beschreibung
-h	zeigt alle Kommandozeilenoptionen an
<i>keine</i>	graphische Schnittstelle (Fenstermodus)
-con	Ausgabe auf eine Konsole
-quiet	keine Ausgabe (außer konfigurierte Logdateien)
-svc[=name]	installiert die Anwendung als Windows-Dienst, der Name des Dienstes kann angegeben werden (falls mehrere ADUM als Dienst ausgeführt werden sollen).
-acc=...	installiert den Dienst unter diesem Konto
-pwd=...	installiert den Dienst mit diesem Passwort, benötigt -acc
-rmsvc[=name]	entfernt den Dienst, evtl. den Namen des Dienstes angeben
-cfg=...	gibt eine abweichende Konfigurationsdatei an (gilt für alle Schnittstellen) – erlaubt die Verwendung verschiedener Profile im gleichen Verzeichnis.

6 Die Entwicklungsumgebung

Die ADUM-Programme wurden mit MS Visual Studio 6 entwickelt. Es sollte mindestens Service Pack 5 für Visual Studio 6, besser Service Pack 6 für Visual C++ 6 installiert sein. Erfolgreich getestet wurde die Quellcodekompatibilität mit folgenden Compilern:

Compiler	System	Hardware
Visual C++ 6 und SP 5	Windows 2000/XP	x86-32bit
Visual C++ 6 und SP 6	Windows XP	x86-32bit
Visual C++ 2003	Windows XP	x86-32bit
Visual C++ 2005	Windows XP	x86-32bit
Visual C++ 2005 und SP1	Windows XP	x86-32bit
Intel C++ Compiler 8.0	Windows XP	x86-32bit
Intel C++ Compiler 9.0	Windows XP	x86-32bit

Die Quellcodekompatibilität mit 64-Bit Compilern auf der Windows64-Plattform ist z. Z. nicht nachgewiesen. Ebenso ist die Ausführung der ADUM-Programme als native 64-Bit Anwendung noch nicht getestet.

Es existieren z. Z. keine Informationen über die Erzeugung der ausführbaren Dateien mittels Cross-Compiler auf UNIX-Systemen.