

**Administrationshandbuch
agorum adminSync
Version 6.5.4**

Copyright 2007-2011, agorum Software GmbH

07.12.2011, Dokumentenversion 1.2.0

Inhaltsverzeichnis

1. Allgemeines	5
2. Der AdminSync-Konfigurator	7
2.1. Starten des Konfigurators	7
2.2. Allgemeine Konfiguration	7
2.2.1. Schlüssel	7
2.2.2. Bei Fehler benachrichtigen	8
2.2.3. Konfigurationen	8
2.3. Login-Einstellungen	8
2.3.1. Konfiguration	8
2.3.2. Verzeichnis-Typ	9
2.3.3. URL	9
2.3.4. Domain	9
2.3.5. Benutzer	9
2.3.6. Passwort	10
2.3.7. Passwort-Server	10
2.3.8. Speichern und Einloggen	10
2.4. Synchronisierungs-Beziehungen	10
2.4.1. Allgemeiner Aufbau	10
2.4.2. Erstellung von Beziehungen	11
2.5. Abschließende Schritte	11
3. agorum core → agorum core	12
3.1. Master	12
3.2. Slave	13
3.3. Anwendungsbeispiel	13
3.3.1. Konfiguration des Masters	14
3.3.2. Konfiguration des Slaves	15
4. agorum core → ADS	16
4.1. Master	16
4.2. Slave	17
4.3. Replikationsserver	18
4.4. Anwendungsbeispiel	18
4.4.1. Konfiguration des Masters	19

4.4.2.	Konfiguration des Slaves	21
4.4.3.	Konfiguration des Replikationsservers	21
5.	ADS → agorum core	22
5.1.	Master	23
5.2.	Slave	23
5.3.	Replikationsserver	25
5.4.	Anwendungsbeispiel	25
5.4.1.	Konfiguration des Masters	27
5.4.2.	Konfiguration des Slaves	27
5.4.3.	Konfiguration der Replikationsserver	30
6.	agorum core → Windows Client	31
6.1.	Master	31
6.2.	Slave	32
6.3.	Anwendungsbeispiel	32
6.3.1.	Konfiguration des Masters	33
6.3.2.	Konfiguration des Slaves	34
7.	agorum core → LDAP	35
7.1.	Default LDAP	35
7.1.1.	Master	35
7.1.2.	Slave	37
7.1.3.	Anwendungsbeispiel	37
7.2.	Posix Ldap	40
7.2.1.	Master	40
7.2.2.	Slave	41
7.2.3.	Anwendungsbeispiel	41
7.3.	Samba Ldap	45
7.3.1.	Master	45
7.3.2.	Slave	46
7.3.3.	Anwendungsbeispiel	46
8.	LDAP → agorum core	51
8.1.	Master	52
8.2.	Slave	53
8.3.	Replikationsserver	54
8.4.	Anwendungsbeispiel	54
8.4.1.	Konfiguration des Masters	56
8.4.2.	Konfiguration des Slaves	56
9.	Installation der Hilfsprogramme	60
9.1.	Der ADS Helper Service	60
9.2.	Der WinClient Helper Service	64

9.3. Der LDAP Helper Service	66
9.3.1. Installation mit agorum core	67
9.3.2. Separate Installation	67
9.3.3. Passwortsynchronisation für Samba	68
10. Beschreibung der MetaDB-Properties	69
10.1. Active	69
10.2. AdditionalGroupSettings	70
10.3. AdditionalUserSettings	70
10.4. AdminDN	71
10.5. AdminPw	72
10.6. AllowInternalPasswordChange	72
10.7. BaseDN	73
10.8. Class	74
10.9. CnGroups	75
10.10 CnUsers	76
10.11 ConnectString	77
10.12 CryptKey	78
10.13 ExcludeUserAttributes	78
10.14 FlatFolderStructure	81
10.15 GroupType	81
10.16 History	82
10.17 LocalServer	83
10.18 LockInsteadOfDelete	84
10.19 NgOsPathOffset	84
10.20 NoGroupInGroup	85
10.21 NotSyncPathControl	86
10.22 ObjectFactory	86
10.23 ParameterNames / ParameterValues	93
10.24 RemotePathOffset	94
10.25 ReplicaServers	95
10.26 Server	95
10.27 SocketTimeout	96
10.28 StateFactory	97
10.29 SyncPathControl	98
10.30 TransactionTimeout	98
10.31 UPNDomainName	99
10.32 UseInternalAuthentication	100
Abbildungsverzeichnis	101
A. Versions-Historie dieses Dokumentes	102
B. Sonstiges	103

Kapitel 1.

Allgemeines

Das AdminSync-Modul bietet die Möglichkeit Benutzer, Gruppen und in Einzelfällen auch ACLs zu anderen Authentifizierungsserver zu synchronisieren, bzw. von diesen zu importieren.

Für den Export können ein anderer **agorum core** Server, ein Active Directory Server von Microsoft, ein Microsoft Windows XP Rechner oder ein (Open)LDAP Server als Slaves verwendet werden. In diesem Fall ist **agorum core** der Master. Für den Import (**agorum core** als Slave) können als Master ein anderer **agorum core** Server oder ein Active Directory Server oder ein (Open)LDAP-Server verwendet werden.

Weitere Features sind unter anderen die Administration von Gruppe in Gruppe auf Systemen, die diese Technik nicht unterstützen, sowie die Rücksynchroisation des Passwortes bei Änderung auf einem Client-Rechner, der an einer ADS- oder Samba-Domäne angemeldet ist ohne eine Installation auf den einzelnen Rechnern.

In der MetaDB werden unter `MAIN_MODULE_MANAGEMENT/ngosadminsync/control/` die Konfigurationen abgelegt.

Unter `MAIN_MODULE_MANAGEMENT/ngosadminsync/samples/` ist für jeden Anwendungsfall eine Beispielkonfiguration zu finden.

In den folgenden Kapiteln wird auf jeden Anwendungsfall u.a. mit einem ausführliches Beispiel eingegangen. Die möglichen MetaDB-Einstellungen werden in dem Kapitel [10](#) eingehend erläutert.

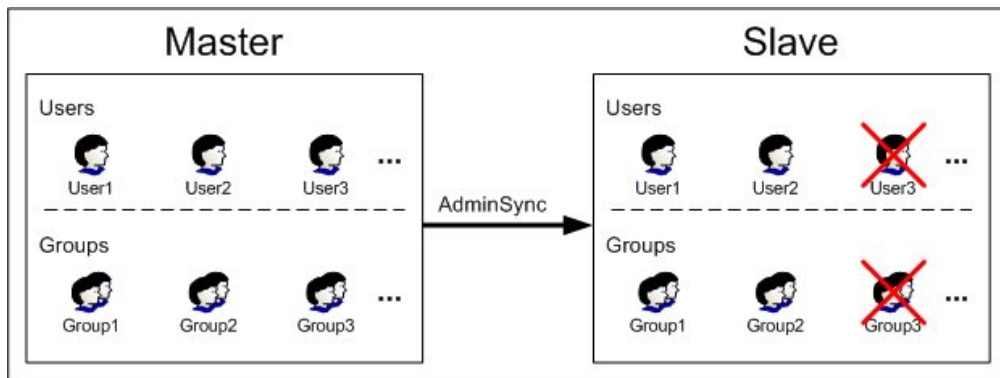


Abbildung 1.1.: Allgemeine Funktionsweise von **agorum** *adminSync*

Kapitel 2.

Der AdminSync-Konfigurator

Zur einfachen Einrichtung einer Synchronisierung mit LDAP-basierten Verzeichnisdiensten – wie zum Beispiel Active Directory – kann der in desk4web integrierte AdminSync-Konfigurator verwendet werden. Der Konfigurator stellt eine grafische Oberfläche zur Erstellung und Bearbeitung von Synchronisierungskonfigurationen bereit. Bitte beachten Sie, dass nur Konfigurationen bearbeitet werden können, die auch durch den Konfigurator erstellt wurden.

2.1. Starten des Konfigurators

Um den AdminSync-Konfigurator zu starten, wechseln Sie in den AdminSync-Bereich von desk4web, der im Portal-Menü unter **Administration** erreichbar ist. Wählen Sie dort im Menü **Vorgang** den Eintrag **Konfigurator**.

2.2. Allgemeine Konfiguration

Nach dem Start des Konfigurators wird die allgemeine Konfigurationsmaske angezeigt.

2.2.1. Schlüssel

Zur Verschlüsselung der Kommunikation mit den Helper-Diensten (siehe Kapitel 9) wird ein Schlüssel benötigt, der auf beiden Seiten übereinstimmen muss. Je länger dieser Schlüssel ist, umso sicherer ist die Übertragung. Um den Schlüssel

zu ändern, geben Sie ihn in das entsprechende Feld ein und klicken Sie auf den zugehörigen **Speichern**-Button.

2.2.2. Bei Fehler benachrichtigen

Wenn bei der Synchronisierung Fehler auftreten, können Benutzer benachrichtigt werden. In dieses Feld können sowohl Benutzernamen als auch E-Mail-Adressen eingetragen werden. Bei mehreren Einträgen verwenden Sie | als Trennzeichen. Lokale Benutzer benötigen Administrationsrechte, um die Meldungen korrekt zu erhalten. Wenn Sie E-Mail-Benachrichtigung verwenden möchten, müssen Sie einen ausgehenden Mailserver konfigurieren. Weitere Informationen hierzu finden Sie im Administrations-Handbuch in Kapitel 3.

2.2.3. Konfigurationen

In dieser Liste werden alle bisher erstellten Konfigurationen angezeigt. Um eine Konfiguration zu bearbeiten, markieren Sie sie und wählen den Button **Konfiguration bearbeiten**. Neue Konfigurationen lassen sich durch den Button **Neue Konfiguration** erstellen.

2.3. Login-Einstellungen

Wird eine vorhandene Konfiguration bearbeitet oder eine neue erstellt, öffnet sich zunächst eine Maske mit Login-Einstellungen.

2.3.1. Konfiguration

Beim Erstellen einer neuen Konfiguration muss hier ein Name gewählt werden. Verwenden Sie bitte ausschließlich alphanumerische Zeichen und _.

2.3.2. Verzeichnis-Typ

Wählen Sie hier den Typ des Benutzerverzeichnisses (LDAP oder Active Directory) aus.

2.3.3. URL

Geben Sie hier die LDAP-URL des Verzeichnis-Servers an. Diese beginnt normalerweise mit `ldap://` oder `ldaps://` (mit SSL-Verschlüsselung) und endet mit der IP-Adresse oder dem Domain-Namen des Servers. Bitte beachten Sie, dass die Synchronisierung von Passwörtern von **agorum core** zu Active Directory eine SSL-Verbindung (`ldaps://`) voraussetzt, für die auf dem Domänencontroller ein Serverzertifikat installiert sein muss¹. Dies betrifft nicht die Synchronisierung von Passwörtern von Active Directory zu **agorum core**, hier sind beide Arten der Verbindung zugelassen.

2.3.4. Domain

Tragen Sie hier den exakten Namen der Domäne Ihres Active Directory ein. Sie können diesen Namen in der Active Directory-Konsole überprüfen.

2.3.5. Benutzer

Geben Sie in dieses Feld den LDAP-Pfad eines Domänen-Administrators ein. Im einfachsten Fall ist das bereits der voreingestellte Wert (Benutzername **Administrator** im Container **Users**). Um diesen Pfad herauszufinden, öffnen Sie auf ihrem Domänencontroller die Active Directory-Konsole und schalten Sie im Menü **Ansicht** die **Erweiterten Features** an. Danach doppelklicken Sie auf den gewünschten Benutzer und öffnen den Reiter **Attribut-Editor**. Dort findet sich der vollständige Pfad des Benutzers unter **distinguishedName**. Das Attribut kann ebenfalls mit einem Doppelklick geöffnet werden, um es in die Zwischenablage kopieren zu können.

Die Angabe der Domänenkomponenten `dc=...` ist an dieser Stelle optional – wenn sie fehlen, wird die unter **Domain** angegebene Domäne verwendet.

¹Ein selbst signiertes Zertifikat genügt für diesen Zweck.

2.3.6. Passwort

Hier wird das Passwort des zuvor festgelegten Benutzers eingetragen.

2.3.7. Passwort-Server

Wenn Sie Passwörter von einem Verzeichnisdienst zu **agorum core** synchronisieren möchten, installieren Sie auf dem Server den zugehörigen Helper-Dienst (siehe Kapitel 9). Im Fall von Active Directory muss dieser Dienst sowohl auf dem Domänencontroller als auch auf allen Replika-Servern (soweit vorhanden) installiert werden. Geben Sie die Liste dieser Server (IP-Adresse oder Domain-Namen) hier durch , getrennt ein. Wenn Sie für den Port eines Helper-Dienstes einen anderen Wert als die Voreinstellung 15016 konfiguriert haben, hängen Sie diese Angabe mit einem : getrennt an die entsprechende Adresse an, also beispielsweise 192.168.0.1:15000 für Port 15000.

2.3.8. Speichern und Einloggen



Klicken Sie auf diesen Button, sobald Sie Ihre Einstellungen eingetragen haben. Sollte ein Fehler auftreten, so wird dieser oben in der Maske angezeigt. Folgen Sie den zugehörigen Anweisungen, um das Problem zu beheben.

2.4. Synchronisierungs-Beziehungen

Nach erfolgreicher Anmeldung am Verzeichnisserver öffnet sich die Synchronisierungsmaske.

2.4.1. Allgemeiner Aufbau

Es werden zwei Verzeichnisbäume angezeigt. Auf der linken Seite sehen Sie eine Übersicht über die Benutzer und Gruppen innerhalb von **agorum core**, auf der rechten Seite ist die gleiche Ansicht für den Verzeichnisdienst zu sehen.

Über beiden Bäumen sind zwei Icons sichtbar: Ein Klick auf  aktualisiert die Ansicht, falls sich inzwischen etwas geändert hat und mit  kann an der aktuell markierten Position ein neuer Ordner angelegt werden.

Unter den Verzeichnisbäumen wird eine Liste der bereits konfigurierten Synchronisierungs-Beziehungen angezeigt (bei einer neu erstellten Konfiguration ist diese noch leer). Wird eine Beziehung ausgewählt, so werden die zugehörigen Ordner in den Verzeichnisbäumen oben hervorgehoben. Die aktuell gewählte Beziehung kann durch einen Klick auf **entfernen** gelöscht werden.

2.4.2. Erstellung von Beziehungen

Um eine neue Synchronisierungs-Beziehung einzurichten, wählen Sie in den Verzeichnisbäumen die beiden Ordner aus, deren Inhalt synchronisiert werden soll. Klicken Sie danach auf einen der beiden Buttons zwischen den Baumansichten. Die Richtung des Pfeils gibt dabei an, in welche Richtung eine Synchronisierung stattfinden soll. Sie können auf diese Weise mehrere Ordner miteinander synchronisieren. Beachten Sie, dass eine Synchronisierung eines Ordners in beide Richtungen gleichzeitig zwar möglich ist, aber nicht empfohlen wird, da somit die Verwaltung der synchronisierten Objekte sehr unübersichtlich wird².

Wenn Sie die Konfiguration abgeschlossen haben, klicken Sie auf **Speichern**, um sie zu sichern und zu aktivieren.

2.5. Abschließende Schritte

Nachdem Sie eine Konfiguration neu erstellt oder geändert haben, sollten sie einmalig alle betroffenen Objekte synchronisieren. Wählen Sie dazu im Menü **Vorgang** im AdminSync-Bereich den Befehl **Sync. aller Objekte** aus. In der folgenden Maske markieren Sie die gewünschte Konfiguration³ und klicken Sie auf **Speichern**.

²Die Verwaltung eines Objekts sollte stets auf der Seite stattfinden, auf der es erzeugt wurde.

³Wenn Sie in einer Konfiguration in beide Richtungen synchronisieren, so werden intern zwei Konfigurationen erstellt und hier angezeigt. Führen Sie in diesem Fall die beschriebenen Schritte für beide Unterkonfigurationen durch.

Kapitel 3.

agorum core → agorum core

Hier wird beschrieben wie Benutzer, Gruppen und ACL mit (fast) allen Eigenschaften zu einem anderen **agorum core** Server synchronisieren werden können.

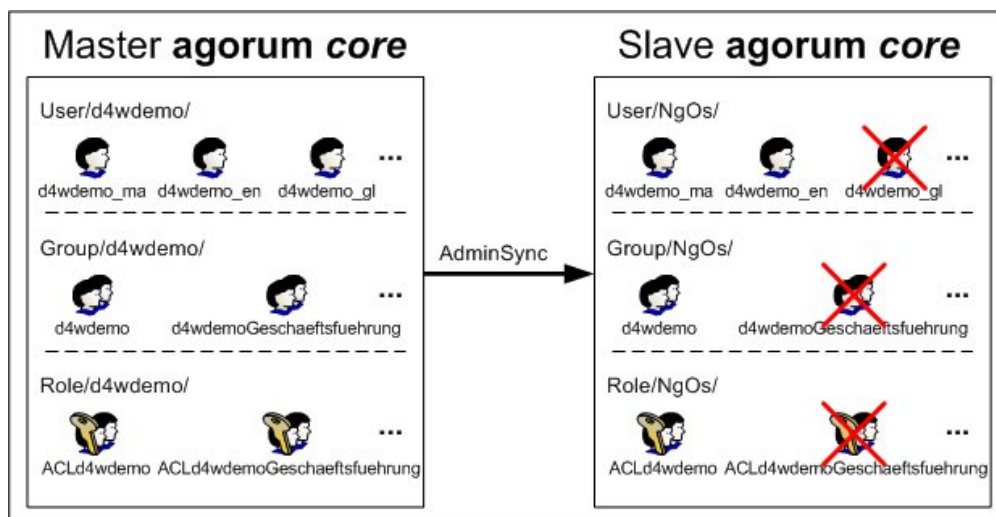


Abbildung 3.1.: Synchronisation von **agorum core** zu einem anderen **agorum core**

3.1. Master

Auf dem Master muss in der MetaDB unter `/MAIN_MODULE_MANAGEMENT/ngosadminsinc/control` eine neue Konfiguration angelegt werden. Unter `/MAIN_MODULE_MANAGEMENT/ngosadminsinc/samples` ist ein Beispiel `NgOsSampleSyncDefinition` für das Synchronisieren zu einem anderen **agorum core** System vorhanden. Sie können dieses Beispiel in den Ordner `control` kopieren und müssen dann nur noch die einzelnen Properties anpassen.

Zwingend benötigte Properties	Optinale Properties
Active SyncPathControl ExcludeUserAttributes Server Class CryptKey	History NotSyncPathControl RemotePathOffset NgOsPathOffset TransactionTimeout

Zu beachten:

- Das Property `Class` muss den Wert `agorum.ngosadminsync.ejb.common.NgOsAdminSyncServiceUtils` haben.
- Das Property `ExcludeUserAttributes` muss auf jeden Fall den Wert `CredentialManager` beinhalten.

3.2. Slave

Auch auf dem Slave muss in der MetaDB unter `/MAIN_MODULE_MANAGEMENT/ngosadminsync/control` eine Konfiguration mit dem gleichen Namen wie auf dem Master angelegt werden. Die Slave-Konfiguration enthält lediglich das `CryptKey`-Property das für die Verschlüsselung notwendig ist.

3.3. Anwendungsbeispiel

In dieser Konfiguration mit dem Namen `TestNgOsSync` sollen Benutzer, Gruppen und ACLs zu einem anderem *agorum core* Server synchronisiert werden.

Alle Benutzer unterhalb von `User/d4wdemo` sollen synchronisiert werden, außer dem Benutzer `d4wdemo_g1`, bei Gruppen alle unter `Group/d4wdemo` außer der Gruppe `d4wdemoGeschaeftsfuehrung` und bei den ACLs alle unter `Role/d4wdemo` außer dem ACL `ACLd4wdemoGeschaeftsfuehrung`. Dieses Verhalten wird mit den beiden Properties `SyncPathControl` und `NotSyncPathControl` gesteuert.

Der Ordner `d4wdemo`, den alle drei Bereiche (`User/NgOs`, `Group/NgOs` und `Role/NgOs`) gemeinsam haben, soll durch das Property `NgOsPathOffset` abgeschnitten werden. Auf dem Zielsystem soll an die drei Bereiche durch das Property `RemotePathOffset` der Ordner `NgOs` vorangestellt werden. So wird z.B. aus `User/d4wdemo/d4wdemo_ma` auf dem Zielsystem `User/NgOs/d4wdemo_ma`.

Bei Benutzern soll die E-Mailadressen (`EmailAddresses`) nicht mitsynchronisiert werden. Der `CredentialManager` darf im Normalfall bei **agorum core** → **agorum core** ebenfalls nicht synchronisiert werden! Das wird über das Property `ExcludeUserAttributes` gesteuert.

Weiter soll eine History erzeugt werden (wird unter `/NgOs AdminSync/TestNgOsSync/history` erstellt). Dafür wird das Property `History` auf `true` gesetzt.

Der Schlüssel `CryptKey` besteht aus einer beliebigen Zeichenkette und muss bei Master und Slave identisch sein! Je länger dieser Schlüssel ist, desto sicherer ist die Verschlüsselung.

Der Slave-Server mit der IP 10.1.2.20 wird via HTTPS auf Port 8088 angesprochen und hat einen Timeout von 40 Sekunden. Der Timeout wird über das Property `Timeout` in Millisekunden eingestellt. Über das Property `Server` wird auf einen Server im Server-Bereich der MetaDB (`/MAIN_SERVER_MANAGEMENT`) verwiesen. Hier wird das Protokoll, die IP und der Port eingestellt.



Das Property `Class` darf nicht geändert werden!

3.3.1. Konfiguration des Masters

In der MetaDB wird in dem Ordner `/MAIN_MODULE_MANAGEMENT/ngosadminsync/control` ein neues Property-Bundle mit dem Namen `TestNgOsSync` erstellt. Darunter werden folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Active	true
Class	agorum.ngosadminsync.ejb.common. NgOsAdminSyncServiceUtils
CryptKey	safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru
ExcludeUserAttributes	CredentialManager EmailAddresses
History	true
NgOsPathOffset	d4wdemo
RemotePathOffset	NgOs
Server	NgOsServer
SyncPathControl	User/d4wdemo Role/d4wdemo Group/d4wdemo
TransactionTimeout	40000
NotSyncPathControl	User/d4wdemo/d4wdemo_g1 Group/d4wdemo/ d4wdemoGeschaeftsfuehrung Role/d4wdemo/ ACLD4wdemoGeschaeftsfuehrung

In der MetaDB muss der in dem Property Server definierte Server angelegt werden. Unter /MAIN_SERVER_MANAGEMENT/[AdminSyncServer] wird ein Property-Bundle mit dem gleichen Namen wie im Property Server (also NgOsServer) angelegt und darunter folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Protocol	https
ServerAddress	10.1.2.20
ServerPort	8088

3.3.2. Konfiguration des Slaves

Auf dem Slave-Rechner wird in der MetaDB in dem Ordner /MAIN_MODULE_MANAGEMENT/ngosadminsync/control ein neues Property-Bundle mit dem Namen TestNgOsSync erstellt. Darunter wird nur der zum Master identische CryptKey abgelegt:

Property-Name	Property-Wert
CryptKey	safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru

Kapitel 4.

agorum core → ADS

Hier wird beschrieben wie sie Benutzer und Gruppen zu einem Active Directory Server von Microsoft synchronisieren können. Bei dieser Synchronisation ist der ADS-Server der Slave und **agorum core** der Master.

Neben dem Master und dem Slave kann es noch eine beliebige Anzahl von Replikationsservern (Secondary ActiveDirectory Server) geben, über die geänderte Passwörter synchronisiert werden.

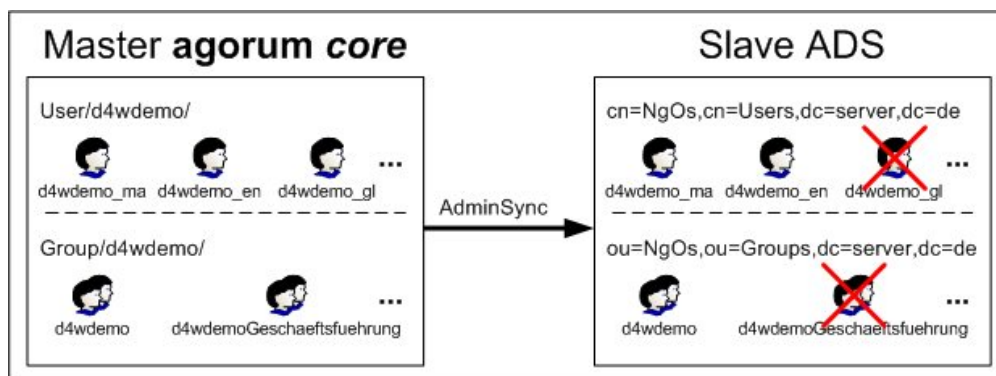


Abbildung 4.1.: Synchronisation von **agorum core** zu einem ADS

4.1. Master

Auf dem Master muss in der MetaDB unter `MAIN_MODULE_MANAGEMENT/ngosadminsinc/control` eine neue Konfiguration angelegt werden. Unter `MAIN_MODULE_MANAGEMENT/ngosadminsinc/samples` gibt es ein Beispiel `ADSSampleSyncDefinition` für das Synchronisieren zu einem ADS-System.

Sie können dieses Beispiel in den Ordner `control` kopieren und müssen dann nur noch die einzelnen Werte anpassen.


Zwingend benötigte Properties	Optinale Properties
Active	History
SyncPathControl	NotSyncPathControl
Server	ExcludeUserAttributes
Class	SocketTimeout
BaseDN	FlatFolderStructure
ConnectString	RemotePathOffset
CnUsers	NgOsPathOffset
CnGroups	UPNDomainName
CryptKey	GroupType
	ReplicaServer

Zu beachten:

- Das Property `Class` muss den Wert `agorum.ngosadminsync.ejb.common.ADSAdminSyncServiceUtils` haben.

CronJobs:

Ist die Synchronisation geänderter Passwörter vom ADS-Server eingestellt, so kann das Intervall über die CronJob Einstellungen unter `MAIN_MODULE_MANAGEMENT/cronjob/control/[AdminSync]/DSPassWordFetchCronJob` geändert werden. In der Standardeinstellung ist das Abholintervall auf eine Minute eingestellt (Cron-Time: `* * * * *`).

	Dieser CronJob gilt für alle agorum core →ADS, ADS→ agorum core , agorum core →LDAP und LDAP→ agorum core Konfigurationen!
---	--

4.2. Slave

Die Installation auf dem Slave erfolgt über das Installationsprogramm des ADS-HelperService (siehe Kapitel 9.1).

Die über das Installationsprogramm gemachten Einstellungen können nachträglich in der Registry unter dem Schlüssel `HKEY_LOCAL_MACHINE\SOFTWARE\agorum\ADSHelper` geändert werden. Dabei dürfen aber nur die Werte für `CryptKey` und `Port` geändert werden und der Dienst muss anschließend neu gestartet werden!

4.3. Replikationsserver

Die Installation auf den Replikationsservern erfolgt über das Installationsprogramm des ADShelperService (siehe Kapitel 9.1).

Die über das Installationsprogramm gemachten Einstellungen können nachträglich in der Registry unter dem Schlüssel `HKEY_LOCAL_MACHINE\SOFTWARE\agorum\ADShelper` geändert werden. Dabei dürfen aber nur die Werte für `CryptKey` und `Port` geändert werden und der Dienst muss anschließend neu gestartet werden!

4.4. Anwendungsbeispiel

In dieser Konfiguration mit dem Namen **TestADSSync** sollen Benutzer und Gruppen zu einem Microsoft Active-Directory-Server synchronisiert werden.

Alle Benutzer unterhalb von `User/d4wdemo` sollen synchronisiert werden, außer dem Benutzer `d4wdemo_g1` und bei Gruppen alle unter `Group/d4wdemo` außer der Gruppe `d4wdemoGeschaefstsfuehrung`. Dieses Verhalten wird mit den beiden Properties `SyncPathControl` und `NotSyncPathControl` gesteuert.

Der Benutzerbereich in *agorum core* (`User/`) soll im ADS nach `cn=Users` gemappt werden. Der Gruppenbereich in *agorum core* (`Group/`) soll im ADS nach `ou=Groups` gemappt werden. Das ergibt z.B. für den ADS-Benutzerbereich zusammen mit einer BaseDN von `dc=server,dc=de` den Distinguished Name `cn=Users,dc=server,dc=de`.

Der Startordner `d4wdemo`, den alle zwei Bereiche (Benutzer und Gruppen) gemeinsam haben, soll abgeschnitten werden (durch `Ng0sPathOffset`). Auf dem Zielsystem soll an die zwei Bereiche über das Property `RemotePathOffset` der Ordner `Ng0s` vorangestellt werden. So wird z.B. aus dem Benutzer `User/d4wdemo/d4wdemo_ma` auf dem Zielsystem `cn=d4wdemo_ma,cn=Ng0s,cn=Users,dc=server,dc=de`.

Bei Benutzern soll der *User Principal Name* nicht die Domain von BaseDN (also `server.de`) als Suffix bekommen, sondern `intern.server.de`. Das wird über das Property `UPNDomainName` gesteuert.

Bei Benutzern soll die E-Mailadressen (`EmailAddresses`) und die Beschreibung (`Description`) nicht mitsynchronisiert werden, was über das Property `ExcludeUserAttributes` gesteuert wird.

Weiter soll eine History erzeugt werden (wird unter `/NgOs AdminSync/TestADSSync/history` erstellt). Dafür wird das Property `History` auf `true` gesetzt.

Der Schlüssel `CryptKey` besteht aus einer beliebigen Zeichenkette und muss bei Master, Slave und den Replikationsservern identisch sein! Je länger dieser Schlüssel ist, desto sicherer ist die Verschlüsselung.

Der ADS-Server mit dem `ADSHelperService` hat die IP `10.1.2.20` und den Port `15016` mit einem Timeout von 30 Sekunden. Der Timeout wird über das Property `Timeout` in Millisekunden eingestellt. Über das Property `Server` wird auf einen Server im Server-Bereich der MetaDB (`/MAIN_SERVER_MANAGEMENT`) verwiesen. Hier wird das Protokoll, die IP und der Port des `ADSHelperService` eingestellt. Da der `ADSHelperService` direkt auf dem ADS Server läuft, verweist der `ConnectionString` auf `localhost`.

Neben dem ADS-Server gibt es noch einen Replikationsserver auf dem ebenfalls der `ADSHelperService` läuft. Dieser Server hat die IP `10.2.2.20`, Port `15016` und einem Timeout von 30 Sekunden. Über das Property `ReplicaServers` wird auf einen Server im Server-Bereich der MetaDB (`/MAIN_SERVER_MANAGEMENT`) verwiesen. Hier wird das Protokoll, die IP und der Port des `ADSHelperService` eingestellt.

Für das holen geänderter Passwörter ist der CronJob `DSPassWordFetchCronJob` zuständig. Er regelt die Periode zum Abholen geänderter Passwörter vom ADS und dem Replikationsserver.



Das Property `Class` darf nicht geändert werden!

4.4.1. Konfiguration des Masters

In der MetaDB wird in dem Ordner `/MAIN_MODULE_MANAGEMENT/ngosadminsinc/control` ein neues Property-Bundle mit dem Namen `TestNgOsSync` erstellt. Darunter werden folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Active	true
History	true
SyncPathControl	User/d4wdemo Group/d4wdemo
NotSyncPathControl	User/d4wdemo/d4wdemo_gl Group/d4wdemo/d4wdemoGeschaeftsfuehrung
ExcludeUserAttributes	EmailAddresses Description
Server	ADSServer
ReplicaServers	ADSReplicaServer
Class	agorum.ngosadminsinc.ejb.common. ADSAdminSyncServiceUtils
BaseDN	dc=server,dc=de
ConnectString	ldap://127.0.0.1
CnUsers	CN=Users
CnGroups	OU=Groups
UPNDomainName	intern.server.de
CryptKey	safg8w3U\sT78jhftdjihZG&u-zvGHfGF%Ru
SocketTimeout	30000
FlatFolderStructure	false
RemotePathOffset	NgOs
NgOsPathOffset	d4wdemo
NoGroupInGroup	false

In der MetaDB müssen die in den Properties `Server` und `ReplicaServers` definierten Server angelegt werden. Dafür werden unter `/MAIN_SERVER_MANAGEMENT/[AdminSyncServer]` Property-Bundles mit dem gleichen Namen `ADSServer` (für das Property `Server`) und `ADSReplicaServer` (für `ReplicaServers`) angelegt und darunter die folgenden Property-Einträge erstellt.

Für `ADSServer`:

Property-Name	Property-Wert
Protocol	ADSService
ServerAddress	10.1.2.20
ServerPort	15016

Für `ADSReplicaServer`:

Property-Name	Property-Wert
Protocol	ADSService
ServerAddress	10.2.2.20
ServerPort	15016

In der MetaDB kann im Order /MAIN_MODULE_MANAGEMENT/cronjob/control/ [AdminSync] der CronJob-Eintrag DSPassWordFetchCronJob angepasst werden.

Standardeinstellungen für DSPassWordFetchCronJob (minütlich):

Property-Name	Property-Wert
Class	agorum.ngosadminsinc.ejb.cron. ADSPassWordFetchCronTaskFactory
CronTime	* * * * *



Dieser CronJob gilt für alle **agorum core**→ADS, ADS→**agorum core**, **agorum core**→LDAP und LDAP→**agorum core** Konfigurationen!

4.4.2. Konfiguration des Slaves

Bei Slave werden die Werte für den Port und für CryptKey bei der Installation des ADShelperService (siehe Kapitel 9.1) gesetzt, bzw. später in der Windows-Registry unter HKEY_LOCAL_MACHINE/SOFTWARE/agorum/ADShelper geändert (Der ADShelperService-Dienst muss danach neu gestartet werden):

Name des Schlüssels	Wert des Schlüssels
Port	15016
CryptKey	safg8w3U\sT78jhftjdjihZG&u-zvgHfGF%Ru

4.4.3. Konfiguration des Replikationservers

Beim Replikationsserver werden die Werte für den Port und für CryptKey bei der Installation des ADShelperService (siehe Kapitel 9.1) gesetzt, bzw. später in der Windows-Registry unter HKEY_LOCAL_MACHINE/SOFTWARE/agorum/ADShelper geändert (Der ADShelperService-Dienst muss danach neu gestartet werden):

Name des Schlüssels	Wert des Schlüssels
Port	15016
CryptKey	safg8w3U\sT78jhftjdjihZG&u-zvgHfGF%Ru

Kapitel 5.

ADS → agorum core

Hier wird beschrieben wie sie Benutzer und Gruppen von einem Active Directory Server synchronisieren, bzw. importieren können. Bei dieser Synchronisation ist der ADS-Server der Master und **agorum core** der Slave, es werden aber trotzdem (fast) alle Einstellungen in **agorum core** vorgenommen.

Neben dem Master und dem Slave kann es noch eine beliebige Anzahl von Replikationsservern (Secondary ActiveDirectory Server) geben, über die geänderte Passwörter synchronisiert werden.

Weiterhin ist zu beachten das sich die synchronisierten Benutzer nicht an **agorum core** authentifizieren, sondern am ADS-Server. Das bedeutet, alle Benutzereinstellungen müssen auf dem ADS-Server vorgenommen werden und nicht in **agorum core**!

Beim initialen Synchronisieren werden die synchronisierten Benutzer im **agorum core** gesperrt. Das liegt daran, dass das Passwort der Benutzer technisch nicht aus dem ADS ausgelesen werden kann und Benutzer ohne Passwort automatisch gesperrt sind. Bei einem ändern bzw. neu setzten des Passwortes über den ADS (hier kann das Passwort technisch abgefangen werden und zu **agorum core** synchronisiert werden) wird der Benutzer freigegeben.

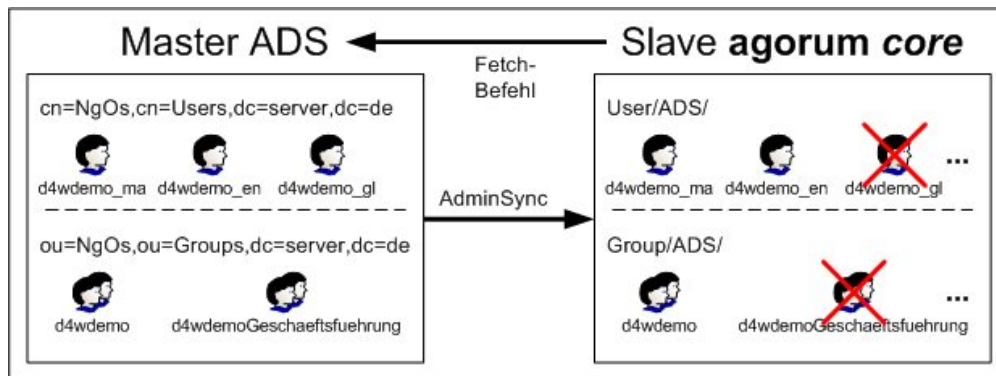


Abbildung 5.1.: Synchronisation von einem ADS zu **agorum core**

5.1. Master

Die Installation für den ADShelperService auf dem Master erfolgt über das Installationsprogramm des ADShelperService (siehe Kapitel 9.1).

Die während der Installation vorgenommenen Einstellungen können nachträglich in der Registry unter dem Schlüssel `HKEY_LOCAL_MACHINE\SOFTWARE\agorum\ADShelper` geändert werden. Dabei sollten aber nur die Werte für `CryptKey` und `Port` geändert werden. Der Dienst muss anschließend neu gestartet werden!

5.2. Slave

Auf dem Slave muss in der MetaDB unter `MAIN_MODULE_MANAGEMENT/ngosadminsinc/control` eine neue Konfiguration angelegt werden. Unter `MAIN_MODULE_MANAGEMENT/ngosadminsinc/samples` ist ein Beispiel `ADSFetchSampleSyncDefinition` für das Synchronisieren von einem ADS-System vorhanden. Sie können dieses Beispiel in den Ordner `control` kopieren und müssen dann nur noch die einzelnen Werte anpassen.

Zwingend benötigte Properties	Optinale Properties
Active	History
SyncPathControl	NotSyncPathControl
Server	ExcludeUserAttributes
LocalServer	SocketTimeout
Class	TransactionTimeout
BaseDN	RemotePathOffset
ConnectString	NgOsPathOffset
CnUsers	UseInternalAuthentication
CnGroups	AllowInternalPasswordChange
CryptKey	LockInsteadOfDelete
	ReplicaServers

Zu beachten:

- Das Property `Class` muss den Wert `agorum.ngosadminsynchron.ejb.common.ADSFetchAdminSyncServiceUtils` haben.
- `RemotePathOffset` und `NgOsPathOffset` sind bei dieser Konfiguration vertauscht. (TODO)

CronJobs:

- Das Abholen der geänderten Objekte kann über einen CronJob periodisch gesteuert werden. Die Einstellungen werden in der MetaDB unter `MAIN_MODULE_MANAGEMENT/cronjob/control/[AdminSync]/DSFetchCronJob` gesetzt. In der Standardeinstellung ist das Abholintervall auf eine Stunde eingestellt (CronTime: `0 * * * *`).
- Da bei dieser AdminSync-Variante sich der Benutzer beim anmelden an *agorum core* über den ADS-Server authentifizieren muss, existiert eine XML-Konfigurationsdatei, die bei Änderungen in der MetaDB automatisch aktualisiert wird. Das Intervall für die Überprüfung auf Änderungen wird in der MetaDB unter `MAIN_MODULE_MANAGEMENT/cronjob/control/[AdminSync]/DSConfigurationCronJob` eingestellt. In der Standardeinstellung ist das Intervall auf eine Minute eingestellt (CronTime: `* * * * *`).
- Ist das Abholen der geänderten Passwörter vom ADS-Server eingestellt, so kann das Intervall über die CronJob Einstellungen unter `MAIN_MODULE_MANAGEMENT/cronjob/control/[AdminSync]/DSPassWordFetchCronJob` geändert werden. In der Standardeinstellung ist das Abholintervall auf eine Minute eingestellt (CronTime: `* * * * *`).



Die CronJobs `DSConfigurationCronJob` und `DSFetchCronJob` gelten für alle ADS→*agorum core* und LDAP→*agorum core* Konfigurationen! Der CronJob `DSPassWordFetchCronJob` gilt für alle ADS→*agorum core*, *agorum core*→ADS, LDAP→*agorum core*, *agorum core*→LDAP Konfigurationen!

5.3. Replikationsserver

Die Installation auf den Replikationsservern erfolgt über das Installationsprogramm des `ADSHelperService` (siehe Kapitel 9.1).

Die über das Installationsprogramm gemachten Einstellungen können nachträglich in der Registry unter dem Schlüssel `HKEY_LOCAL_MACHINE\SOFTWARE\agorum\ADSHelper` geändert werden. Dabei dürfen aber nur die Werte für `CryptKey` und `Port` geändert werden und der Dienst muss anschließend neu gestartet werden!

5.4. Anwendungsbeispiel

In dieser Konfiguration mit dem Namen `TestFetchADSSync` sollen Benutzer und Gruppen von einem Microsoft Active Directory Server synchronisiert werden, von 2 Replica Servern sollen geänderte Passwörter synchronisiert werden. Die Besonderheit bei diesem Synchronisationstyp ist, dass die geänderten Benutzer, Gruppen und Passwörter vom Master, bzw. den Replica Servern durch den Slave aktiv geholt werden müssen. Der Slave (*agorum core*) holt die Objekte über einen CronJob, bzw. manuell durch eine Anweisung des Administrators.

Alle Benutzer unterhalb von `cn=Demo, cn=Users` sollen synchronisiert werden, außer dem Benutzer `d4wdemo_g1` und bei Gruppen alle unter `ou=Demo, ou=Groups` außer der Gruppe `d4wdemoGeschaeftsfuehrung`. Dieses Verhalten wird mit den beiden Properties `SyncPathControl` und `NotSyncPathControl` gesteuert.

Der Benutzerbereich im ADS (`cn=Users`) soll in *agorum core* nach `User/` gemappt werden. Der Gruppenbereich im ADS (`ou=Groups`) soll in *agorum core* nach `Group/` gemappt werden. Der ADS-Server hat eine BaseDN von `dc=server, dc=de`.

Der Startordner `ou=Demo`, bzw. `cn=Demo`, den alle zwei Bereiche (Benutzer und Gruppen) gemeinsam haben, soll abgeschnitten werden (durch `RemotePathOffset`). Auf dem Zielsystem soll an die zwei Bereiche über das Property `NgOsPathOffset`

der Ordner ADS vorangestellt werden. So wird z.B. aus `cn=d4wdemo_ma,cn=Demo,cn=Users,dc=server,dc=de` auf dem Zielsystem `User/ADS/d4wdemo_ma`.

Bei Benutzern soll die E-Mailadressen (`EmailAddresses`) und die Beschreibung (`Description`) nicht mitsynchronisiert werden, was über das Property `ExcludeUserAttributes` gesteuert wird.

Wird im ADS ein Benutzer gelöscht, so soll er in **agorum core** nur gesperrt werden. Im ADS gelöschte Gruppen sollen in **agorum core** unverändert bleiben. Das wird mit dem Property `LockInsteadOfDelete` gesteuert.

Die Benutzerauthentifizierung soll über den ADS-Server (`UseInternalAuthentication`) laufen und Benutzer dürfen ihr Passwort nicht in **agorum core** ändern (`AllowInternalPasswordChange`).

Weiter soll eine History erzeugt werden (wird unter `/NgOs AdminSync/TestFetchADSSync/history` erstellt). Dafür wird das Property `History` auf `true` gesetzt.

Der Schlüssel `CryptKey` besteht aus einer beliebigen Zeichenkette und muss bei Master und Slave identisch sein! Je länger dieser Schlüssel ist, desto sicherer ist die Verschlüsselung.

Der Master-ADS-Server mit der IP 10.1.2.40, auf dem der `ADSHelperService` auf Port 15016 mit einem Timeout von 30 Sekunden läuft. Der Timeout wird über das Property `Timeout` in Millisekunden eingestellt. Über das Property `Server` wird auf einen Server im Server-Bereich der MetaDB (`/MAIN_SERVER_MANAGEMENT`) verwiesen. Hier wird das Protokoll, die IP und der Port des `ADSHelperService` eingestellt. Da der `ADSHelperService` auf dem ADS Server läuft, verweist der `ConnectionString` auf `localhost`.

Neben dem ADS-Server gibt es noch zwei Replikationsserver auf denen ebenfalls der `ADSHelperService` läuft. Diese Server haben die IP 10.2.2.10 und 10.3.2.5, beide haben den Port 15016 und einem Timeout von 30 Sekunden. Über das Property `ReplicaServers` wird auf zwei Server im Server-Bereich der MetaDB (`/MAIN_SERVER_MANAGEMENT`) verwiesen. Hier wird das Protokoll, die IP und der Port des `ADSHelperService` eingestellt.

Für diesen Konfigurationstyp sind drei `CronJobs` zuständig:

DSFetchCronJob Regelt die Periode zum Abholen geänderter Benutzer bzw. Gruppen vom ADS (bzw. von LDAP-Servern).

DSConfigurationCronJob Regelt die Periode zum Prüfen geänderter Konfigurationen.

DSPassWordFetchCronJob Regelt die Periode zum Abholen geänderter Passwörter vom ADS (bzw. von LDAP-Servern).



Das Property `Class` darf nicht geändert werden! Ebenso sollte der Wert `DEFAULTSERVER` bei dem Property `LocalServer` nicht geändert werden. Das Property `TransactionTimeout` ist bereits auch sinnvoll vorbelegt.

5.4.1. Konfiguration des Masters

Beim Master werden die Werte für den `Port` und für `CryptKey` bei der Installation des `ADSHelperService` (siehe Kapitel 9.1) gesetzt, bzw. später in der Windows-Registry unter `HKEY_LOCAL_MACHINE/SOFTWARE/agorum/ADSHelper` geändert (Der `ADSHelperService`-Dienst muss danach neu gestartet werden):

Name des Schlüssels	Wert des Schlüssels
Port	15016
CryptKey	safg8w3U\sT78jhftjdjihZG&u-zvgHfGF%Ru

5.4.2. Konfiguration des Slaves

In der MetaDB wird in dem Ordner `/MAIN_MODULE_MANAGEMENT/ngosadminsync/control\path` ein neues Property-Bundle mit dem Namen `TestFetchADSSync` erstellt. Darunter werden folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Active	true
History	true
SyncPathControl	User/d4wdemo Group/d4wdemo
NotSyncPathControl	User/d4wdemo/d4wdemo_g1 Group/d4wdemo/d4wdemoGeschaeftsfuehrung
ExcludeUserAttributes	EmailAddresses Description
ReplicaServers	ADSReplicaServer1 ADSReplicaServer2
Server	ADSServer
Class	agorum.ngosadminsynchron.ejb.common.ADSFetchAdminSyncServiceUtils
BaseDN	dc=server,dc=de
ConnectionString	ldap://127.0.0.1
CnUsers	CN=Users
CnGroups	OU=Groups
UPNDomainName	intern.server.de
CryptKey	safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru
SocketTimeout	30000
RemotePathOffset	Demo
NgOsPathOffset	ADS
LocalServer	DEFAULTSERVER
TransactionTimeout	300000
UseInternalAuthentication	false
AllowInternalPasswordChange	false
LockInsteadOfDelete	true

In der MetaDB müssen die in den Properties `Server` und `ReplicaServers` definierten Server angelegt werden. Unter `/MAIN_SERVER_MANAGEMENT/[AdminSyncServer]` werden Property-Bundles mit dem gleichen Namen wie in den Properties `Server` und `ReplicaServers` (also `ADSServer`, `ADSReplicaServer1` und `ADSReplicaServer2`) angelegt und darunter folgende Property-Einträge erstellt:

Für `ADSServer`:

Property-Name	Property-Wert
Protocol	ADSService
ServerAddress	10.1.2.40
ServerPort	15016

Für `ADSReplicaServer1`:

Property-Name	Property-Wert
Protocol	ADSService
ServerAddress	10.2.2.10
ServerPort	15016

Für ADSReplicaServer2:

Property-Name	Property-Wert
Protocol	ADSService
ServerAddress	10.3.2.5
ServerPort	15016

In der MetaDB können im Order /MAIN_MODULE_MANAGEMENT/cronjob/control/[AdminSync] die drei CronJob-Einträge DSFetchCronJob, DSConfigurationCronJob und DSPassWordFetchCronJob angepasst werden.

Standardeinstellungen für DSFetchCronJob (stündlich):

Property-Name	Property-Wert
Class	agorum.ngosadminsinc.ejb.cron. DSFetchCronTaskFactory
CronTime	0 * * * *

Standardeinstellungen für DSConfigurationCronJob (minütlich):

Property-Name	Property-Wert
Class	agorum.ngosadminsinc.ejb.cron. DSConfigurationCronTaskFactory
CronTime	* * * * *

Standardeinstellungen für DSPassWordFetchCronJob (minütlich):

Property-Name	Property-Wert
Class	agorum.ngosadminsinc.ejb.cron. DSPassWordFetchCronTaskFactory
CronTime	* * * * *



Die CronJobs DSConfigurationCronJob und DSFetchCronJob gelten für alle ADS→*agorum core* und LDAP→*agorum core* Konfigurationen! Der CronJob DSPassWordFetchCronJob gilt für alle ADS→*agorum core*, LDAP→*agorum core*, *agorum core*→ADS und *agorum core*→LDAP Konfigurationen!

5.4.3. Konfiguration der Replikationsserver

Bei den Replikationsservern werden die Werte für den `Port` und für `CryptKey` bei der Installation des `ADSHelperService` (siehe Kapitel 9.1) gesetzt, bzw. später in der Windows-Registry unter `HKEY_LOCAL_MACHINE/SOFTWARE/agorum/ADSHelper` geändert (Der `ADSHelperService`-Dienst muss danach neu gestartet werden):

Name des Schlüssels	Wert des Schlüssels
Port	15016
CryptKey	safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru

Kapitel 6.

agorum core → Windows Client

Hier wird beschrieben wie sie geänderte Benutzerpasswörter zu einem Windows 2000/XP Client synchronisieren können. Benutzer können mit diesem Synchronisationstyp nicht angelegt werden. Es werden nur die Passwörter ausgewertet.

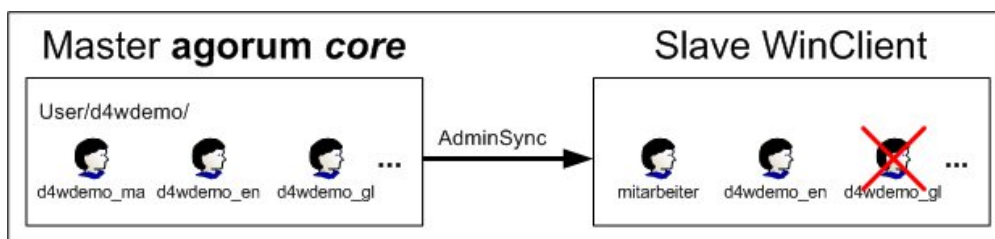


Abbildung 6.1.: Synchronisation von **agorum core** zu einem Windows Client

6.1. Master

Auf dem Master muss in der MetaDB unter `MAIN_MODULE_MANAGEMENT/ngosadminsinc/control` eine neue Konfiguration angelegt werden. Unter `MAIN_MODULE_MANAGEMENT/ngosadminsinc/samples` ist ein Beispiel `WinClientSampleSyncDefinition` für das Synchronisieren zu einem Windows Client vorhanden. Sie können dieses Beispiel in den Ordner `control` kopieren und müssen dann nur noch die einzelnen Werte anpassen.

Zwingend benötigte Properties	Optimale Properties
Active SyncPathControl Server Class CryptKey	History NotSyncPathControl SocketTimeout

Zu beachten:

- Das Property Class muss den Wert `agorum.ngosadminsync.ejb.common.WinClientAdminSyncServiceUtils` haben.

6.2. Slave

Die Installation für den Slave des AdminSyncs erfolgt über das Installationsprogramm des WinClientHelperService (siehe Kapitel 9.2).

Die über das Installationsprogramm gemachten Einstellungen können nachträglich in der Registry unter dem Schlüssel `HKEY_LOCAL_MACHINE\SOFTWARE\agorum\ADSHelper` geändert werden. Dabei sollten aber nur die Werte für `CryptKey` und `Port` geändert werden!

6.3. Anwendungsbeispiel

In dieser Konfiguration mit dem Namen **TestWinClientSync** sollen Benutzer zu einem Microsoft Windows Rechner synchronisiert werden.

Alle Benutzer unterhalb von `User/d4wdemo` sollen synchronisiert werden, außer dem Benutzer `d4wdemo_gl`. Dieses Verhalten wird mit den beiden Properties `SyncPathControl` und `NotSyncPathControl` gesteuert. Der Benutzer `d4wdemo_ma` heist auf dem Zielsystem `mitarbeiter`. Das kann über das Property `AdditionalUserSettings` gemappt werden.

Weiter soll eine History erzeugt werden (wird unter `/NgOs AdminSync/TestWinClientSync/history` erstellt). Dafür wird das Property `History` auf `true` gesetzt.

Der Schlüssel `CryptKey` besteht aus einer beliebigen Zeichenkette und muss bei Master und Slave identisch sein! Je länger dieser Schlüssel ist, desto sicherer ist die Verschlüsselung.

Der Windows Client auf dem der WinClientHelperService auf Port 15016 mit einem Timeout von 30 Sekunden läuft, hat die IP 10.1.2.101. Der Timeout wird über das Property `Timeout` in Millisekunden eingestellt. Über das Property `Server` wird

auf einen Server im Server-Bereich der MetaDB (`/MAIN_SERVER_MANAGEMENT`) verwiesen. Hier wird das Protokoll, die IP und der Port des `WinClientHelperService` eingestellt.



Das Property class darf nicht geändert werden!

6.3.1. Konfiguration des Masters

In der MetaDB wird in dem Ordner `/MAIN_MODULE_MANAGEMENT/ngosadminsync/control` ein neues Property-Bundle mit dem Namen `TestWinClientSync` erstellt. Darunter werden folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Active	true
History	true
SyncPathControl	User/d4wdemo
NotSyncPathControl	User/d4wdemo/d4wdemo_gl
AdditionalUserSettings	d4wdemo_ma:mitarbeiter
Class	agorum.ngosadminsync.ejb.common. ADSFetchAdminSyncServiceUtils
CryptKey	safg8w3U\sT78jhftdjihZG&u-zvGHfGF%Ru
Server	WinClient
SocketTimeout	30000

In der MetaDB muss der in dem Property `Server` definierte Server angelegt werden. Unter `/MAIN_SERVER_MANAGEMENT/[AdminSyncServer]` wird ein Property-Bundle mit dem gleichen Namen wie im Property `Server` (also `WinClient`) angelegt und darunter folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Protocol	WinClient
ServerAddress	10.1.2.101
ServerPort	15016

6.3.2. Konfiguration des Slaves

Bei Slave werden die Werte für den `Port` und für `CryptKey` bei der Installation des `inClientHelperService` (siehe Kapitel 9.2) gesetzt, bzw. später in der Registry unter `HKEY_LOCAL_MACHINE\SOFTWARE\agorum\ADSHelper` geändert (Der `ADSHelperService`-Dienst muss danach neu gestartet werden):

Name des Schlüssels	Wert des Schlüssels
Port	15016
CryptKey	safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru

Kapitel 7.

agorum core → LDAP

Hier wird beschrieben wie sie Benutzer und Gruppen zu einem LDAP Server (z.B. OpenLDAP) synchronisieren können. Zuerst wird auf die Grundkonfiguration einer LDAP-Synchronisierung eingegangen und im Weiteren auf spezielle Implementationen (z.B. Posix oder Samba).

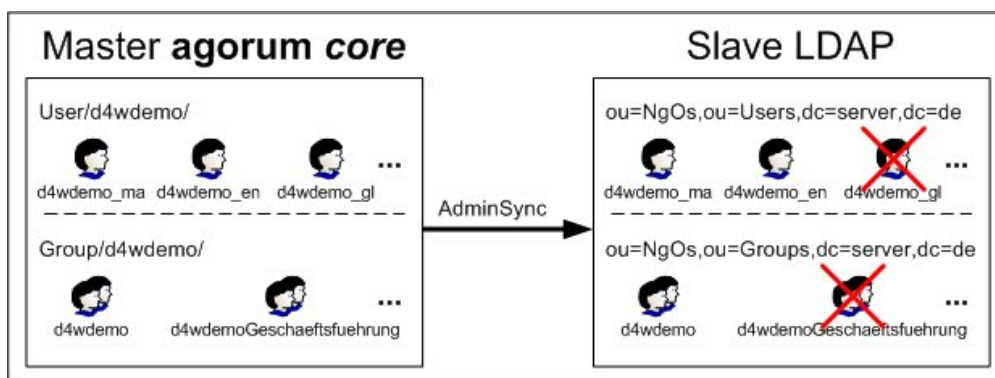


Abbildung 7.1.: Synchronisation von **agorum core** zu LDAP

7.1. Default LDAP

7.1.1. Master

Auf dem Master muss in der MetaDB unter `MAIN_MODULE_MANAGEMENT/ngosadminsinc/control` eine neue Konfiguration angelegt werden. Unter `MAIN_MODULE_MANAGEMENT/ngosadminsinc/samples` ist ein Beispiel `LDAPSampleSyncDefinition` für das Synchronisieren zu einem LDAP-System vorhanden. Sie können dieses Beispiel in den Ordner `control` kopieren und müssen dann nur noch die einzelnen Werte anpassen.

Zwingend benötigte Properties	Optinale Properties
Active SyncPathControl Server Class BaseDN ConnectString AdminDN AdminPw CnUsers CnGroups CryptKey ObjectFactory StateFactory	History NotSyncPathControl ExcludeUserAttributes SocketTimeout RemotePathOffset NgOsPathOffset FlatFolderStructure NoGroupInGroup

Zu beachten:

- Das Property `Class` muss den Wert `agorum.ngosadminsynchron.ejb.common.LDAPAdminSyncServiceUtils` haben.
- Das Property `ObjectFactory` muss den Wert `agorum.ldapadminsynchronservice.factories.DefaultLdapObjectFactory` haben.
- Das Property `StateFactory` muss den Wert `agorum.ldapadminsynchronservice.factories.DefaultLdapStateFactory` haben.

Alternativen

Mit folgenden Factories wird anstelle des Gruppentyps `GroupOfNames` der Typ `GroupOfUniqueNames` verwendet.

Property-Name	Property-Wert
ObjectFactory	<code>agorum.ldapadminsynchronservice.factories.GroupOfUniqueNamesLdapObjectFactory</code>
StateFactory	<code>agorum.ldapadminsynchronservice.factories.GroupOfUniqueNamesLdapStateFactory</code>

Mit folgenden Factories wird anstelle des Attributtyps `cn` für den Benutzernamen der Typ `uid` verwendet.

Property-Name	Property-Wert
ObjectFactory	<code>agorum.ldapadminsynchronservice.factories.UidLdapObjectFactory</code>
StateFactory	<code>agorum.ldapadminsynchronservice.factories.UidLdapStateFactory</code>

7.1.2. Slave

Der Slave (LDAPHelperService) ist standardmäßig bei einer **agorum core** Installation dabei und läuft auf dem **agorum core** Rechner (siehe DefaultLDAP Beispiel in Kapitel 7.1), kann aber auch separat auf einem anderen Rechner installiert werden (siehe PosixLDAP Beispiel in Kapitel 7.2). Die Konfiguration wird in der XML-Datei `/Pfad/zu/agorum core/jboss/server/default/deploy/roi.ear/config/ldapadminsynchronservice/LDAPAdminSyncService.xml` abgelegt, bzw. beim Start des LDAPHelperService mitgegeben.

Inhalt der Konfigurationsdatei:

```
1 <LDAPAdminSyncService>
2   <Port>15016</Port>
3   <CryptKey>Put in any random key for encrypting the
      communication. Has to be the same as on the remote
      system.</CryptKey>
4   <QueuePath>/opt/agorum/ldapadminsynchronservice/queues</
      QueuePath>
5 </LDAPAdminSyncService>
```

Wird der LDAPHelperService auf einem anderen Rechner installiert, dann muss beim Start des Service der volle Pfad zur Konfigurationsdatei übergeben werden. Beispiel:

```
1 java agorum.ldapadminsynchronservice.common.LDAPHelperService /
      opt/agorum/ldapadminsynchronservice/config/
      LDAPAdminSyncService.xml
```

7.1.3. Anwendungsbeispiel

In dieser Konfiguration mit dem Namen **TestLDAPSync** sollen Benutzer und Gruppen zu einem (Open-)LDAP Server synchronisiert werden.

Alle Benutzer unterhalb von `User/d4wdemo` sollen synchronisiert werden, außer dem Benutzer `d4wdemo_g1` und bei Gruppen alle unter `Group/d4wdemo` außer der Gruppe `d4wdemoGeschaeftsfuehrung`. Dieses Verhalten wird mit den beiden Properties `SyncPathControl` und `NotSyncPathControl` gesteuert.

Der Benutzerbereich in **agorum core** (`User/`) soll im LDAP nach `ou=Users` gemappt werden. Der Gruppenbereich in **agorum core** (`Group/`) soll im LD-

AP nach `ou=Groups` gemappt werden. Das ergibt beispielsweise für den LDAP-Benutzerbereich zusammen mit einer BaseDN von `dc=server,dc=de` den Distinguished Name `ou=Users,dc=domain,dc=de`.

Der Startordner `d4wdemo`, den alle zwei Bereiche (Benutzer und Gruppen) gemeinsam haben, soll abgeschnitten werden (durch `NgOsPathOffset`). Auf dem Zielsystem soll an die zwei Bereiche über das Property `RemotePathOffset` der Ordner `NgOs` vorangestellt werden. So wird z.B. aus `User/d4wdemo/d4wdemo_ma` auf dem Zielsystem `cn=d4wdemo_ma,ou=NgOs,ou=Users,dc=server,dc=de`.

Bei Benutzern soll die E-Mailadressen (`EmailAddresses`) und die Beschreibung (`Description`) nicht mit synchronisiert werden, was über das Property `ExcludeUserAttributes` gesteuert wird.

Weiter soll eine History erzeugt werden (wird unter `/NgOs AdminSync/TestLDAPSync/history` erstellt). Dafür wird das Property `History` auf `true` gesetzt.

Der Schlüssel `CryptKey` besteht aus einer beliebigen Zeichenkette und muss bei Master und Slave identisch sein! Je länger dieser Schlüssel ist, desto sicherer ist die Verschlüsselung.

Der Slave-Server hat die IP 10.1.2.30 auf dem der LDAP-Server auf Port 389 läuft (`ConnectString`). Der `LDAPHelperService` läuft auf dem *agorum core* Rechner (lokal) auf Port 15016 mit einem Timeout von 30 Sekunden. Der Timeout wird über das Property `Timeout` in Millisekunden eingestellt. Über das Property `Server` wird auf einen Server im Server-Bereich der MetaDB (`/MAIN_SERVER_MANAGEMENT`) verwiesen. Hier wird das Protokoll, die IP und der Port des `LDAPHelperService` eingestellt. Da der `LDAPHelperService` lokal läuft, verweisen die Einträge bei den Servereinstellungen auf den lokalen Rechner, die Property für den `ConnectString` auf den LDAP-Server.



Die Properties `Class`, `ObjectFactory` und `StateFactory` dürfen nicht geändert werden!

Konfiguration des Masters

In der MetaDB wird in dem Ordner `/MAIN_MODULE_MANAGEMENT/ngosadminsinc/control` ein neues Property-Bundle mit dem Namen `TestLDAPSync` erstellt. Darunter werden folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Active	true
Class	agorum.ngosadminsinc.ejb.common.LDAPAdminSyncServiceUtils
ObjectFactory	agorum.ldapadminsincservice.factories.DefaultLdapObjectFactor
StateFactory	agorum.ldapadminsincservice.factories.DefaultLdapStateFactor
CryptKey	safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru
ExcludeUserAttributes	EmailAddresses Description
History	true
NgOsPathOffset	d4wdemo
RemotePathOffset	NgOs
Server	ADSServer
BaseDN	dc=server,dc=de
ConnectString	ldap://10.1.2.30
AdminDN	cn=Manager,dc=server,dc=de
AdminPw	secret
CnUsers	ou=Users
CnGroups	ou=Groups
SyncPathControl	User/d4wdemo Group/d4wdemo
TransactionTimeout	40000
SocketTimeout	30000
NotSyncPathControl	User/d4wdemo/d4wdemo_g1 Group/d4wdemo/d4wdemoGeschaefthsfuehrung
FlatFolderStructure	false
NoGroupInGroup	false

In der MetaDB muss der in dem Property `Server` definierte Server angelegt werden. Unter `/MAIN_SERVER_MANAGEMENT/[AdminSyncServer]` wird ein Property-Bundle mit dem gleichen Namen wie im Property `Server` (also `ADSServer`) angelegt und darunter folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Protocol	ADSService
ServerAddress	127.0.0.1
ServerPort	15016

Konfiguration des Slaves

Für den Slave werden die Werte für den Port und für CryptKey in der XML-Datei LDAPAdminSyncService.xml gespeichert, die unter /opt/agorum/desk4web/jboss/server/default/deploy/roi.ear/config/ldapadminsynchronservice liegt:

```

1 <LDAPAdminSyncService>
2   <Port>15016</Port>
3   <CryptKey>safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru</CryptKey>
4   <QueuePath>/opt/agorum/ldapadminsynchronservice/queues</
      QueuePath>
5 </LDAPAdminSyncService>

```

7.2. Posix Ldap

7.2.1. Master

Eine der Spezialkonfigurationen für den LDAP sind Posix Accounts. Posix Accounts werden verwendet, wenn sich ein Unix/Linux-System gegen einen LDAP-Server authentifizieren soll. Alle hier beschriebenen Einstellungen bauen auf dem Default-LDAP-Einstellungen auf.

Zwingend benötigte Properties	Optinale Properties
Active	History
SyncPathControl	NotSyncPathControl
Server	ExcludeUserAttributes
Class	SocketTimeout
BaseDN	RemotePathOffset
ConnectString	NgOsPathOffset
AdminDN	FlatFolderStructure
AdminPw	NoGroupInGroup
CnUsers	AdditionalUserSettings
CnGroups	AdditionalGroupSettings
CryptKey	ParameterNames
ObjectFactory	ParameterValues
StateFactory	

Zu beachten:

- Das Property Class muss den Wert `agorum.ngosadminsynchron.ejb.common.LDAPAdminSyncServiceUtils` haben.
- Das Property ObjectFactory muss den Wert `agorum.ldapadminsynchronservice.factories.PosixLdapObjectFactory` haben.
- Das Property StateFactory muss den Wert `agorum.ldapadminsynchronservice.factories.PosixLdapStateFactory` haben.
- Keine Umlaute im Benutzernamen und im Gruppennamen, da sonst der Homefolder nicht angelegt werden kann!
- SuSE YAST trägt am Ende der Datei `/etc/passwd` (vllt. auch `/etc/groups?`) folgende Zeile ein: `+:::..`. Das weist auf weitere, externe Einträge (LDAP) hin.

7.2.2. Slave

Siehe Kapitel *Slave* bei der Default LDAP Konfiguration (Kapitel 7.1.2).

7.2.3. Anwendungsbeispiel

In dieser Konfiguration mit dem Namen **TestPosixLDAPSync** sollen Benutzer und Gruppen zu einem (Open-)LDAP Server für Posix-Authentifikation synchronisiert werden.

Alle Benutzer unterhalb von `User/d4wdemo` sollen synchronisiert werden, außer dem Benutzer `d4wdemo_g1` und bei Gruppen alle unter `Group/d4wdemo` außer der Gruppe `d4wdemoGeschaeftsfuehrung`. Dieses Verhalten wird mit den beiden Properties `SyncPathControl` und `NotSyncPathControl` gesteuert.

Der Benutzerbereich in **agorum core** (`User/`) soll im LDAP nach `ou=Users` gemappt werden. Der Gruppenbereich in **agorum core** (`Group/`) soll im LDAP nach `ou=Groups` gemappt werden. Das ergibt beispielsweise für den LDAP-Benutzerbereich zusammen mit einer BaseDN von `dc=server,dc=de` den Distinguished Name `ou=Users,dc=domain,dc=de`.

Der Startordner `d4wdemo`, den alle zwei Bereiche (Benutzer und Gruppen) gemeinsam haben, soll abgeschnitten werden (durch `Ng0sPathOffset`). Auf dem Zielsystem soll an die zwei Bereiche über das Property `RemotePathOffset` der Ordner `Ng0s` vorangestellt werden. So wird z.B. aus `User/d4wdemo/d4wdemo_ma` auf dem Zielsystem `cn=d4wdemo_ma,ou=Ng0s,ou=Users,dc=server,dc=de`.

Bei Benutzern soll die E-Mailadressen (`EmailAddresses`) und die Beschreibung (`Description`) nicht mit synchronisiert werden, was über das Property `ExcludeUserAttributes` gesteuert wird.

Weiter soll eine History erzeugt werden (wird unter `/NgOs AdminSync/TestPosixLDAPSync/history` erstellt). Dafür wird das Property `History` auf `true` gesetzt.

Der Schlüssel `CryptKey` besteht aus einer beliebigen Zeichenkette und muss bei Master und Slave identisch sein! Je länger dieser Schlüssel ist, desto sicherer ist die Verschlüsselung.

Spezielle Einstellungen für die Posix-Authentifizierung:

- Da ein Posix-System nicht mit Gruppen In Gruppen umgehen kann, ist es notwendig das Property `NoGroupInGroup` auf `true` zu setzen. Das bewirkt, das Untergruppen in die Benutzer aufgelöst werden.
- User-Ids sollen im LDAP bei 1500 beginnen (`StartUidNumber` bei `ParameterNames-/Values`)
- Gruppen-Ids sollen bei 2000 beginnen (`StartGidNumber` bei `ParameterNames-/Values`)
- Benutzer sollen in der Standardgruppe mit der GID 100 sein (`UserGidNumber` bei `ParameterNames-/Values`)
- Die Benutzerverzeichnisse liegen unter `/home` (`HomeDirectoryPrefix` bei `ParameterNames-/Values`)
- Die Loginshell soll `/bin/bash` sein (`LoginShell` bei `ParameterNames-/Values`)
- Der **agorum core** Benutzer `root` soll abweichende Einstellungen bekommen (gesteuert über `AdditionalUserSettings`):
 - User-Id: 0
 - Standardgruppen-Id: 0
 - Homeverzeichnis: `/root`
 - Loginshell: `/bin/zsh`
- Die **agorum core** Gruppe `root` soll abweichende Einstellungen bekommen (gesteuert über `AdditionalGroupSettings`):
 - Gruppen-Id: 0

Der Slave-Server hat die IP 10.1.2.30 auf dem der LDAP-Server auf Port 389 läuft. Der `LDAPHelperService` läuft ebenfalls auf dem Slave-Server auf auf Port 15016 mit einem Timeout von 30 Sekunden. Der Timeout wird über das Property `Timeout` in Millisekunden eingestellt. Über das Property `Server` wird auf einen Server im Server-Bereich der MetaDB (`/MAIN_SERVER_MANAGEMENT`) verwiesen. Hier wird das Protokoll, die IP und der Port des `LDAPHelperService` eingestellt. Da der `LDA-HelperService` auf dem Slave-Server läuft, verweisen die Einträge bei den Server-

einstellungen auf die IP des Slave-Servers, die Property für den `ConnectionString` auf den lokalen Rechner (LDAP-Server).



Die Properties `Class`, `ObjectFactory` und `StateFactory` dürfen nicht geändert werden!

Konfiguration des Masters

In der MetaDB wird in dem Ordner `/MAIN_MODULE_MANAGEMENT/ngosadminsync/control` ein neues Property-Bundle mit dem Namen `TestPosixLDAPSync` erstellt. Darunter werden folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Active	true
History	true
SyncPathControl	User/d4wdemo Group/d4wdemo
NotSyncPathControl	User/d4wdemo/d4wdemo_gl Group/d4wdemo/d4wdemoGeschaeftsfuehrung
ExcludeUserAttributes	EmailAddresses Description
Server	LDAPServer
Class	agorum.ngosadminsinc.ejb.common.LDAPAdminSyncServiceUtils
ObjectFactory	agorum.ldapadminsincservice.factories.PosixLdapObjectFactory
StateFactory	agorum.ldapadminsincservice.factories.PosixLdapStateFactory
BaseDN	dc=server,dc=de
ConnectString	ldap://localhost
AdminDN	cn=Manager,dc=server,dc=de
AdminPw	secret
CnUsers	ou=Users
CnGroups	ou=Groups
CryptKey	safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru
SocketTimeout	30000
FlatFolderStructure	false
RemotePathOffset	NgOs
NgOsPathOffset	d4wdemo
NoGroupInGroup	true
NgOsPathOffset	d4wdemo
ParameterNames	StartUidNumber StartGidNumber UserGidNumber HomeDirectoryPrefix LoginShell
ParameterValues	1500 1500 100 /home /bin/bash
AdditionalUserSettings	root:0:0:/root:/bin/bash
AdditionalGroupSettings	root:0

In der MetaDB muss der in dem Property `Server` definierte Server angelegt werden. Unter `/MAIN_SERVER_MANAGEMENT/[AdminSyncServer]` wird ein Property-Bundle mit dem gleichen Namen wie im Property `Server` (also `LDAPServer`) angelegt und darunter folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Protocol	LDAPService
ServerAddress	10.1.2.30
ServerPort	15016

Konfiguration des Slaves

Für den Slave werden die Werte für den Port und für CryptKey in der XML-Datei `LDAPAdminSyncService.xml` gespeichert, die unter `/opt/agorum/ldapadminsyncservice/config` liegt (LDAPHelperService ist lokal auf dem Slave-Server installiert):

```
1 <LDAPAdminSyncService>
2   <Port>15016</Port>
3   <CryptKey>safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru</CryptKey>
4   <QueuePath>/opt/agorum/ldapadminsyncservice/queues</
   QueuePath>
5 </LDAPAdminSyncService>
```

Gestartet wird der LDAPHelperService mit dem Befehl

```
1 java agorum.ldapadminsyncservice.common.LDAPHelperService /
   opt/agorum/ldapadminsyncservice/config/
   LDAPAdminSyncService.xml
```

7.3. Samba Ldap

7.3.1. Master

Eine der Spezialkonfigurationen für den LDAP sind Samba Accounts. Samba Accounts werden verwendet, um eine Windows NT Domain zu erstellen. Ähnlich zu einem Active Directory Server, können sich so Windows Clients authentifizieren. Alle hier beschriebenen Einstellungen bauen auf den Posix-LDAP-Einstellungen auf.

Zwingend benötigte Properties	Optinale Properties
Active	History
SyncPathControl	NotSyncPathControl
Server	ExcludeUserAttributes
Class	SocketTimeout
BaseDN	RemotePathOffset
ConnectString	NgOsPathOffset
AdminDN	FlatFolderStructure
AdminPw	NoGroupInGroup
CnUsers	AdditionalUserSettings
CnGroups	AdditionalGroupSettings
CryptKey	ParameterNames
ObjectFactory	ParameterValues
StateFactory	ReplicaServes

Zu beachten:

- Das Property `Class` muss den Wert `agorum.ngosadminsync.ejb.common.LDAPAdminSyncServiceUtils` haben.
- Das Property `ObjectFactory` muss den Wert `agorum.ldapadminsyncservice.factories.SambaLdapObjectFactory` oder `agorum.ldapadminsyncservice.factories.UidSambaLdapObjectFactory` haben.
- Das Property `StateFactory` muss den Wert `agorum.ldapadminsyncservice.factories.PosixLdapStateFactory` oder `agorum.ldapadminsyncservice.factories.UidPosixLdapStateFactory` haben.
- Keine Umlaute im Benutzernamen und im Gruppennamen, da sonst der Homefolder nicht angelegt werden kann!

7.3.2. Slave

Siehe Kapitel *Slave* bei der Default LDAP Konfiguration (Kapitel 7.1.2).

7.3.3. Anwendungsbeispiel

In dieser Konfiguration mit dem Namen **TestSambaLDAPSync** sollen Benutzer und Gruppen zu einem (Open-)LDAP Server für die Samba-Authentifikation synchronisiert werden.

Alle Benutzer unterhalb von `User/d4wdemo` sollen synchronisiert werden, außer dem

Benutzer `d4wdemo_g1` und bei Gruppen alle unter `Group/d4wdemo` außer der Gruppe `d4wdemoGeschaefstsfuehrung`. Dieses Verhalten wird mit den beiden Properties `SyncPathControl` und `NotSyncPathControl` gesteuert.

Der Benutzerbereich in *agorum core* (`User/`) soll im LDAP nach `ou=Users` gemappt werden. Der Gruppenbereich in *agorum core* (`Group/`) soll im LDAP nach `ou=Groups` gemappt werden. Das ergibt beispielsweise für den LDAP-Benutzerbereich zusammen mit einer BaseDN von `dc=server,dc=de` den Distinguished Name `ou=Users,dc=domain,dc=de`.

Der Startordner `d4wdemo`, den alle zwei Bereiche (Benutzer und Gruppen) gemeinsam haben, soll abgeschnitten werden (durch `Ng0sPathOffset`). Auf dem Zielsystem soll an die zwei Bereiche über das Property `RemotePathOffset` der Ordner `Ng0s` vorangestellt werden. So wird z.B. aus `User/d4wdemo/d4wdemo_ma` auf dem Zielsystem `cn=d4wdemo_ma,ou=Ng0s,ou=Users,dc=server,dc=de`.

Bei Benutzern soll die Beschreibung (`Description`) nicht mit synchronisiert werden, was über das Property `ExcludeUserAttributes` gesteuert wird.

Weiter soll eine History erzeugt werden (wird unter `/Ng0s AdminSync/TestSambaLDAPSync/history` erstellt). Dafür wird das Property `History` auf `true` gesetzt.

Der Schlüssel `CryptKey` besteht aus einer beliebigen Zeichenkette und muss bei Master und Slave identisch sein! Je länger dieser Schlüssel ist, desto sicherer ist die Verschlüsselung.

Spezielle Einstellungen für die Samba-Authentifizierung:

- Da ein Samba-System nicht mit Gruppen In Gruppen umgehen kann, ist es notwendig das Property `NoGroupInGroup` auf `true` zu setzen. Das bewirkt, das Untergruppen in die Benutzer aufgelöst werden.
- Benutzer sollen in der Standardgruppe mit der GID 513 sein (`UserGidNumber` bei `ParameterNames-/Values`)
- Die Benutzerverzeichnisse liegen unter `/home` (`HomeDirectoryPrefix` bei `ParameterNames-/Values`)
- Die Loginshell soll `/bin/bash` sein (`LoginShell` bei `ParameterNames-/Values`)
- Die SID der Domain ist `S-1-5-21-3259675530-1368597822-2486849306` (`SambaSIDPrefix` bei `ParameterNames-/Values`)
- Die aktuelle GID und UID wird im LDAP unter `sambaDomainName=TestDomain,ou=samba,dc=server,dc=de` abgelegt (`SambaUnixIdPool` bei `ParameterNames-/Values`)
 - Standardgruppen-Id: 513

- Homeverzeichnis: /root
- Loginshell: /bin/zsh
- SID: S-1-5-21-3259675530-1368597822-2486849306
- SambaUnixIdPool: sambaDomainName=TestDomain,ou=samba,dc=server,dc=de

Der Slave-Server hat die IP 10.1.2.30 auf dem der LDAP-Server auf Port 389 läuft. Der LDAPHelperService läuft ebenfalls auf dem Slave-Server auf auf Port 15016 mit einem Timeout von 30 Sekunden. Der Timeout wird über das Property `Timeout` in Millisekunden eingestellt. Über das Property `Server` wird auf einen Server im Server-Bereich der MetaDB (/MAIN_SERVER_MANAGEMENT) verwiesen. Hier wird das Protokoll, die IP und der Port des LDAPHelperService eingestellt. Da der LDAPHelperService auf dem Slave-Server läuft, verweisen die Einträge bei den Server-einstellungen auf die IP des Slave-Servers, die Property für den `ConnectionString` auf den lokalen Rechner (LDAP-Server).



Die Properties `Class`, `ObjectFactory` und `StateFactory` dürfen nicht geändert werden!

Konfiguration des Masters

In der MetaDB wird in dem Ordner /MAIN_MODULE_MANAGEMENT/ngosadminsync/control ein neues Property-Bundle mit dem Namen `TestSambaLDAPSync` erstellt. Darunter werden folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Active	true
History	true
SyncPathControl	User/d4wdemo Group/d4wdemo
NotSyncPathControl	User/d4wdemo/d4wdemo_gl Group/d4wdemo/d4wdemoGeschaeftsfuehrung
ExcludeUserAttributes	Description
Server	LDAPServer
Class	agorum.ngosadminsinc.ejb.common.LDAPAdminSyncServiceUtils
ObjectFactory	agorum.ldapadminsincservice.factories.SambaLdapObjectFactory
StateFactory	agorum.ldapadminsincservice.factories.SambaLdapStateFactory
BaseDN	dc=server,dc=de
ConnectString	ldap://localhost
AdminDN	cn=Manager,dc=server,dc=de
AdminPw	secret
CnUsers	ou=Users
CnGroups	ou=Groups
CryptKey	safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru
SocketTimeout	30000
FlatFolderStructure	false
RemotePathOffset	NgOs
NgOsPathOffset	d4wdemo
NoGroupInGroup	true
NgOsPathOffset	d4wdemo
ParameterNames	UserGidNumber HomeDirectoryPrefix LoginShell SambaSIDPrefix SambaUnixIdPool
ParameterValues	513 /home /bin/bash S-1-5-21-3259675530-1368597822-2486849306 sambaDomainName=TestDomain,ou=samba,dc=server,dc=de

In der MetaDB muss der in dem Property `Server` definierte Server angelegt werden. Unter `/MAIN_SERVER_MANAGEMENT/[AdminSyncServer]` wird ein Property-Bundle mit dem gleichen Namen wie im Property `Server` (also `LDAPServer`) angelegt und darunter folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Protocol	LDAPService
ServerAddress	10.1.2.30
ServerPort	15016

Konfiguration des Slaves

Für den Slave werden die Werte für den Port und für CryptKey in der XML-Datei `LDAPAdminSyncService.xml` gespeichert, die unter `/opt/agorum/ldapadminsyncservice/config` liegt (LDAPHelperService ist lokal auf dem Slave-Server installiert):

```
1 <LDAPAdminSyncService>
2   <Port>15016</Port>
3   <CryptKey>safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru</CryptKey>
4   <QueuePath>/opt/agorum/ldapadminsyncservice/queues</
      QueuePath>
5 </LDAPAdminSyncService>
```

Gestartet wird der LDAPHelperService mit dem Befehl

```
1 java agorum.ldapadminsyncservice.common.LDAPHelperService /
   opt/agorum/ldapadminsyncservice/config/
   LDAPAdminSyncService.xml
```

Kapitel 8.

LDAP → agorum core

Hier wird beschrieben wie sie Benutzer und Gruppen von einem LDAP-Server synchronisieren, bzw. importieren können. Bei dieser Synchronisation ist der LDAP-Server der Master und **agorum core** der Slave, es werden aber trotzdem (fast) alle Einstellungen in **agorum core** vorgenommen.

Neben dem Master und dem Slave kann es noch eine beliebige Anzahl von Replikationsservern (Backup Domain Controller) geben, über die geänderte Passwörter synchronisiert werden.

Weiterhin ist zu beachten das sich die synchronisierten Benutzer nicht an **agorum core** authentifizieren, sondern am LDAP-Server. Das bedeutet, alle Benutzereinstellungen müssen auf dem ADS-Server vorgenommen werden und nicht in **agorum core**!

Beim initialen Synchronisieren werden die synchronisierten Benutzer im **agorum core** gesperrt. Das liegt daran, dass das Passwort der Benutzer technisch nicht aus dem LDAP ausgelesen werden kann und Benutzer ohne Passwort automatisch gesperrt sind. Bei einem ändern bzw. neu setzen des Passwortes über den Samba-LDAP (hier kann das Passwort technisch abgefangen werden und zu **agorum core** synchronisiert werden) wird der Benutzer freigegeben.

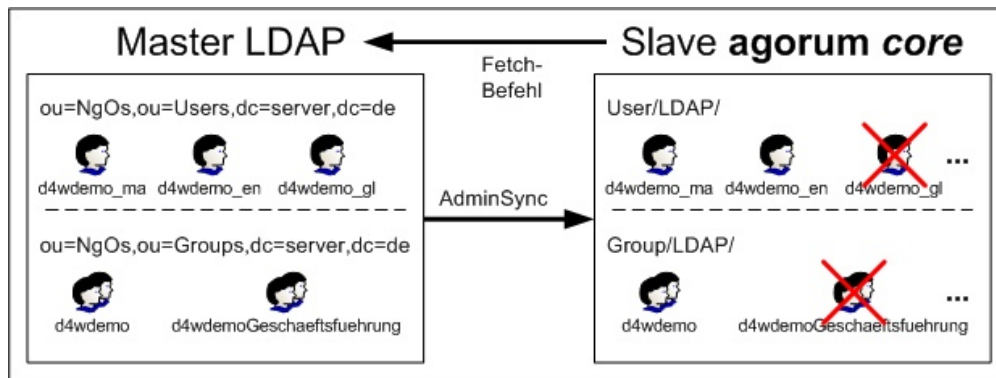


Abbildung 8.1.: Synchronisation von LDAP zu *agorum core*

8.1. Master

Der LDAPHelperService ist standardmäßig bei einer *agorum core* Installation dabei und läuft auf dem *agorum core* Rechner (siehe DefaultLDAP Beispiel in Kapitel 7.1), kann aber auch separat auf einem anderen Rechner installiert werden (siehe PosixLDAP Beispiel in Kapitel 7.2). Die Konfiguration wird in der XML-Datei `/Pfad/zu/agorum core/jboss/server/default/deploy/roi.ear/config/ldapadminsynchronservice/LDAPAdminSyncService.xml` abgelegt, bzw. beim Start des LDAPHelperService mitgegeben.

Inhalt der Konfigurationsdatei:

```

1 <LDAPAdminSyncService>
2   <Port>15016</Port>
3   <CryptKey>Put in any random key for encrypting the
      communication. Has to be the same as on the remote
      system.</CryptKey>
4   <QueuePath>/opt/agorum/ldapadminsynchronservice/queues</
      QueuePath>
5 </LDAPAdminSyncService>

```

Wird der LDAPHelperService auf einem anderen Rechner installiert, dann muss beim Start des Service der volle Pfad zur Konfigurationsdatei übergeben werden. Beispiel:

```

1 java agorum.ldapadminsynchronservice.common.LDAPHelperService /
      opt/agorum/ldapadminsynchronservice/config/
      LDAPAdminSyncService.xml

```

8.2. Slave

Auf dem Slave muss in der MetaDB unter `MAIN_MODULE_MANAGEMENT/ngosadminsync/control` eine neue Konfiguration angelegt werden. Unter `MAIN_MODULE_MANAGEMENT/ngosadminsync/samples` sind die Beispiele `LDAPFetchSampleSyncDefinition` und `LDAPSambaFetchSampleSyncDefinition` für das Synchronisieren von einem LDAP-System vorhanden. Sie können diese Beispiele in den Ordner `control` kopieren und müssen dann nur noch die einzelnen Werte anpassen.

Zwingend benötigte Properties	Optinale Properties
Active	History
SyncPathControl	NotSyncPathControl
Server	ExcludeUserAttributes
LocalServer	SocketTimeout
Class	TransactionTimeout
BaseDN	RemotePathOffset
ConnectString	NgOsPathOffset
CnUsers	AllowInternalPasswordChange
CnGroups	LockInsteadOfDelete
CryptKey	ReplicaServers
AdminDN	
AdminPw	
ObjectFactory	
StateFactory	

Zu beachten:

- Das Property `Class` muss den Wert `agorum.ngosadminsync.ejb.common.LDAPFetchAdminSyncServiceUtils` haben.
- `RemotePathOffset` und `NgOsPathOffset` sind bei dieser Konfiguration vertauscht.

CronJobs:

- Das Abholen der geänderten Objekte kann über einen CronJob periodisch gesteuert werden. Die Einstellungen werden in der MetaDB unter `MAIN_MODULE_MANAGEMENT/cronjob/control/[AdminSync]/DSFetchCronJob` gesetzt. In der Standardeinstellung ist das Abholintervall auf eine Stunde eingestellt (`CronTime: 0 * * * *`).
- Da bei dieser AdminSync-Variante sich der Benutzer beim anmelden an *agorum core* über den LDAP-Server authentifizieren muss, existiert ei-

ne XML-Konfigurationsdatei, die bei Änderungen in der MetaDB automatisch aktualisiert wird. Das Intervall für die Überprüfung auf Änderungen wird in der MetaDB unter `MAIN_MODULE_MANAGEMENT/cronjob/control/[AdminSync]/DSConfigurationCronJob` eingestellt. In der Standardeinstellung ist das Intervall auf eine Minute eingestellt (CronTime: * * * * *).

- Ist das Abholen der geänderten Passwörter vom LDAP-Server (nur Samba!) eingestellt, so kann das Intervall über die CronJob Einstellungen unter `MAIN_MODULE_MANAGEMENT/cronjob/control/[AdminSync]/DSPassWordFetchCronJob` geändert werden. In der Standardeinstellung ist das Abholintervall auf eine Minute eingestellt (CronTime: * * * * *).



Die CronJobs `DSConfigurationCronJob` und `DSFetchCronJob` gelten für alle `ADS→agorum core` und `LDAP→agorum core` Konfigurationen! Der CronJob `DSPassWordFetchCronJob` gilt für alle `ADS→agorum core`, `agorum core→ADS`, `LDAP→agorum core`, `agorum core→LDAP` Konfigurationen!

8.3. Replikationsserver

Die Installation auf den Replikationsservern erfolgt über das Installationsprogramm des LDAPHelperService (siehe Kapitel ??).

8.4. Anwendungsbeispiel

In dieser Konfiguration mit dem Namen **TestFetchLDAPSync** sollen Benutzer und Gruppen von einem LDAP-Server mit Samba-Kopplung synchronisiert werden, von 2 Replica Servern (Backup Domain Controller) sollen geänderte Passwörter synchronisiert werden. Die Besonderheit bei diesem Synchronistai-onstyp ist, dass die geänderten Benutzer, Gruppen und Passwörter vom Master, bzw. den Replica Servern durch den Slave aktiv geholt werden müssen. Der Slave (**agorum core**) holt die Objekte über einen CronJob, bzw. manuell durch eine Anweisung des Administrators.

Alle Benutzer unterhalb von `cn=Demo,ou=Users` sollen synchronisiert werden, außer dem Benutzer `d4wdemo_g1` und bei Gruppen alle unter `ou=Demo,ou=Groups` außer der Gruppe `d4wdemoGeschaeftsfuehrung`. Dieses Verhalten wird mit den beiden Properties `SyncPathControl` und `NotSyncPathControl` gesteuert.

Der Benutzerbereich im LDAP (`ou=Users`) soll in **agorum core** nach `User/` gemappt werden. Der Gruppenbereich im ADS (`ou=Groups`) soll in **agorum core** nach `Group/` gemappt werden. Der ADS-Server hat eine BaseDN von `dc=server, dc=de`.

Der Startordner `ou=Demo`, bzw. `ou=Demo`, den alle zwei Bereiche (Benutzer und Gruppen) gemeinsam haben, soll abgeschnitten werden (durch `RemotePathOffset`). Auf dem Zielsystem soll an die zwei Bereiche über das Property `NgOsPathOffset` der Ordner LDAP vorangestellt werden. So wird z.B. aus `cn=d4wdemo_ma,ou=Demo,ou=Users,dc=server,dc=de` auf dem Zielsystem `User/LDAP/d4wdemo_ma`.

Bei Benutzern soll die E-Mailadressen (`EmailAddresses`) und die Beschreibung (`Description`) nicht mitsynchronisiert werden, was über das Property `ExcludeUserAttributes` gesteuert wird.

Wird im LDAP ein Benutzer gelöscht, so soll er in **agorum core** nur gesperrt werden. Im LDAP gelöschte Gruppen sollen in **agorum core** unverändert bleiben. Das wird mit dem Property `LockInsteadOfDelete` gesteuert.

Weiter soll eine History erzeugt werden (wird unter `/NgOs AdminSync/TestFetchLDAPSync/history` erstellt). Dafür wird das Property `History` auf `true` gesetzt.

Der Schlüssel `CryptKey` besteht aus einer beliebigen Zeichenkette und muss bei Master, Slave und den Replica-Servern identisch sein! Je länger dieser Schlüssel ist, desto sicherer ist die Verschlüsselung.

Der LDAP-Server (PDC) mit der IP 10.1.2.40, auf dem der `LDAPHelperService` auf Port 15016 mit einem Timeout von 30 Sekunden läuft. Der Timeout wird über das Property `Timeout` in Millisekunden eingestellt. Über das Property `Server` wird auf einen Server im Server-Bereich der MetaDB (`/MAIN_SERVER_MANAGEMENT`) verwiesen. Hier wird das Protokoll, die IP und der Port des `LDAPHelperService` eingestellt. Da der `LDAPHelperService` auf dem LDAP-Server läuft, verweist der `ConnectionString` auf `localhost`.

Neben dem LDAP-Server gibt es noch zwei Replikationsserver (BDC) auf denen ebenfalls der `LDAPHelperService` läuft. Diese Server haben die IP 10.2.2.10 und 10.3.2.5, beide haben den Port 15016 und einem Timeout von 30 Sekunden. Über das Property `ReplicaServers` wird auf zwei Server im Server-Bereich der MetaDB (`/MAIN_SERVER_MANAGEMENT`) verwiesen. Hier wird das Protokoll, die IP und der Port des `LDAPHelperService` eingestellt.

Für diesen Konfigurationstyp sind drei CronJobs zuständig:

DSFetchCronJob Regelt die Periode zum Abholen geänderter Benutzer bzw. Gruppen vom LDAP (bzw. von ADS-Servern).

DSConfigurationCronJob Regelt die Periode zum Prüfen geänderter Konfigurationen.

DSPassWordFetchCronJob Regelt die Periode zum Abholen geänderter Passwörter vom LDAP (bzw. von ADS-Servern).



Das Property `Class` darf nicht geändert werden! Ebenso sollte der Wert `DEFAULTSERVER` bei dem Property `LocalServer` nicht geändert werden. Das Property `TransactionTimeout` ist bereits auch sinnvoll vorbelegt.

8.4.1. Konfiguration des Masters

Für den Master werden die Werte für den `Port` und für `CryptKey` in der XML-Datei `LDAPAdminSyncService.xml` gespeichert, die unter `/opt/agorum/ldapadminsyncservice/config` liegt (LDAPHelperService ist lokal auf dem Slave-Server installiert):

```
1 <LDAPAdminSyncService>
2   <Port>15016</Port>
3   <CryptKey>safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru</CryptKey>
4   <QueuePath>/opt/agorum/ldapadminsyncservice/queues</
      QueuePath>
5 </LDAPAdminSyncService>
```

Gestartet wird der LDAPHelperService mit dem Befehl

```
1 java agorum.ldapadminsyncservice.common.LDAPHelperService /
   opt/agorum/ldapadminsyncservice/config/
   LDAPAdminSyncService.xml
```

8.4.2. Konfiguration des Slaves

In der MetaDB wird in dem Ordner `/MAIN_MODULE_MANAGEMENT/ngosadminsync/control\path` ein neues Property-Bundle mit dem Namen `TestFetchLDAPSync` erstellt. Darunter werden folgende Property-Einträge erstellt:

Property-Name	Property-Wert
Active	true
AdminDN	cn=Manager,dc=server,dc=de
AdminPw	secret
History	true
SyncPathControl	User/d4wdemo Group/d4wdemo
NotSyncPathControl	User/d4wdemo/d4wdemo_g1 Group/d4wdemo/ d4wdemoGeschaeftsfuehrung
ExcludeUserAttributes	EmailAddresses Description
ReplicaServers	LDAPReplicaServer1 LDAPReplicaServer2
Server	LDAPServer
Class	agorum.ngosadminsinc.ejb.common. LDAPFetchAdminSyncServiceUtils
ObjectFactory	agorum.ldapadminsincservice.factories. SambaLdapObjectFactory
StateFactory	agorum.ldapadminsincservice.factories. SambaLdapStateFactory
BaseDN	dc=server,dc=de
ConnectString	ldap://127.0.0.1
CnUsers	OU=Users
CnGroups	OU=Groups
CryptKey	safg8w3U\sT78jhftdjihZG&u-zvgHfGF%Ru
SocketTimeout	30000
RemotePathOffset	Demo
NgOsPathOffset	LDAP
LocalServer	DEFAULTSERVER
LockInsteadOfDelete	true
TransactionTimeout	300000
AllowInternalPasswordChange	false
LockInsteadOfDelete	true

In der MetaDB müssen die in den Properties `Server` und `ReplicaServers` definierten Server angelegt werden. Unter `/MAIN_SERVER_MANAGEMENT/[AdminSyncServer]` werden Property-Bundles mit dem gleichen Namen wie in den Properties `Server` und `ReplicaServers` (also `LDAPServer`, `LDAPReplicaServer1` und `LDAPReplicaServer2`) angelegt und darunter folgende Property-Einträge erstellt:

Für `LDAPServer`:

Property-Name	Property-Wert
Protocol	LDAPService
ServerAddress	10.1.2.40
ServerPort	15016

Für LDAPReplicaServer1:

Property-Name	Property-Wert
Protocol	LDAPService
ServerAddress	10.2.2.10
ServerPort	15016

Für LDAPReplicaServer2:

Property-Name	Property-Wert
Protocol	LDAPService
ServerAddress	10.3.2.5
ServerPort	15016

In der MetaDB können im Order `/MAIN_MODULE_MANAGEMENT/cronjob/control/[AdminSync]` die drei CronJob-Einträge `DSFetchCronJob`, `DSConfigurationCronJob` und `DSPassWordFetchCronJob` angepasst werden.

Standardeinstellungen für `DSFetchCronJob` (stündlich):

Property-Name	Property-Wert
Class	<code>agorum.ngosadminsync.ejb.cron.DSFetchCronTaskFactory</code>
CronTime	<code>0 * * * *</code>

Standardeinstellungen für `DSConfigurationCronJob` (minütlich):

Property-Name	Property-Wert
Class	<code>agorum.ngosadminsync.ejb.cron.DSConfigurationCronTaskFactory</code>
CronTime	<code>* * * * *</code>

Standardeinstellungen für `DSPassWordFetchCronJob` (minütlich):

Property-Name	Property-Wert
Class	agorum.ngosadminsync.ejb.cron. DSPassWordFetchCronTaskFactory
CronTime	* * * * *



Die CronJobs `DSConfigurationCronJob` und `DSFetchCronJob` gelten für alle `ADS→agorum core` und `LDAP→agorum core` Konfigurationen! Der CronJob `DSPassWordFetchCronJob` gilt für alle `ADS→agorum core`, `LDAP→agorum core`, `agorum core→ADS` und `agorum core→LDAP` Konfigurationen!

Kapitel 9.

Installation der Hilfprogramme

9.1. Der ADS Helper Service

Der ADS Helper Services ist für die Synchronisation mit einem Active Directory Server notwendig. Im Normalfall wird der Helper Service auf dem ADS-Server installiert.

Um den ADS Helper Service zu installieren sind folgende Schritte durchzuführen:

1. Start des Installationsprogrammes `setup-agorum-adshelperservice-6_X_X.exe`.
2. Auswählen der Installationssprache (siehe Abbildung 9.1).
3. Installer starten (siehe Abbildung 9.2).
4. Lizenz bestätigen (siehe Abbildung 9.3).
5. Eingeben des Schlüssels und des Ports (siehe Abbildung 9.4).
6. Auswählen des Installationspfades (siehe Abbildung 9.5).
7. Auswählen des Startmenü-Ordners (siehe Abbildung 9.6).
8. Rechte für die Verzeichnisse `c:\Programme\agorum\ADSHelperService\queue`, `c:\Programme\agorum\ADSHelperService\queue2` und die Datei `c:\WINDOWS\system32\ADSPasswordFilterHelper.exe` auf ein Minimum setzten (siehe Abbildung 9.7). Dieser Schritt wird **nicht** durch das Installationsprogramm durchgeführt und muss von Hand gemacht werden!

9. Den Server neu booten (siehe Abbildung 9.8).



Abbildung 9.1.: Auswahl der Sprache

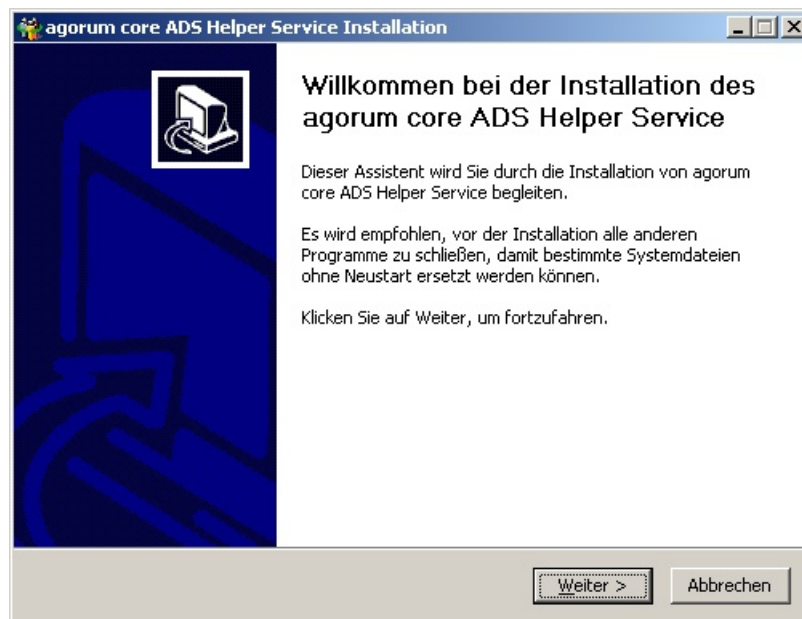


Abbildung 9.2.: Installer starten

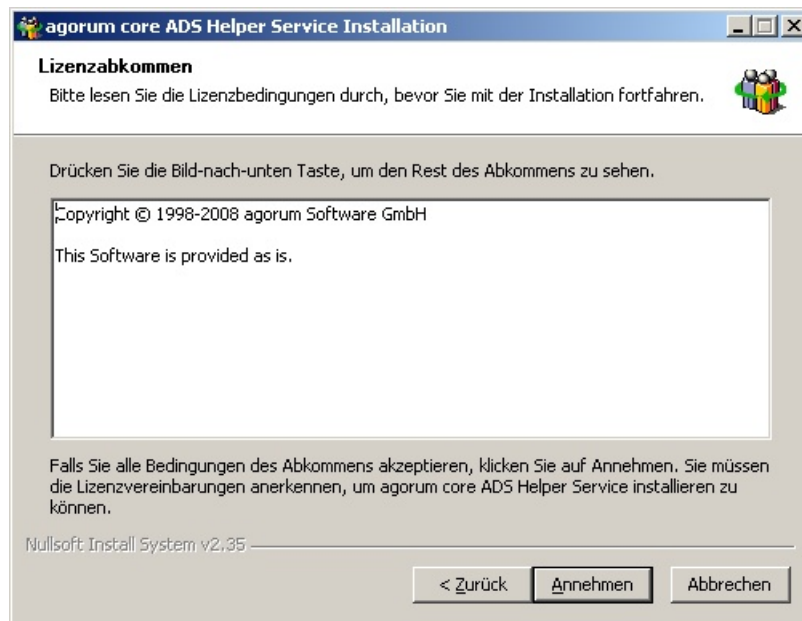


Abbildung 9.3.: Lizenz bestätigen

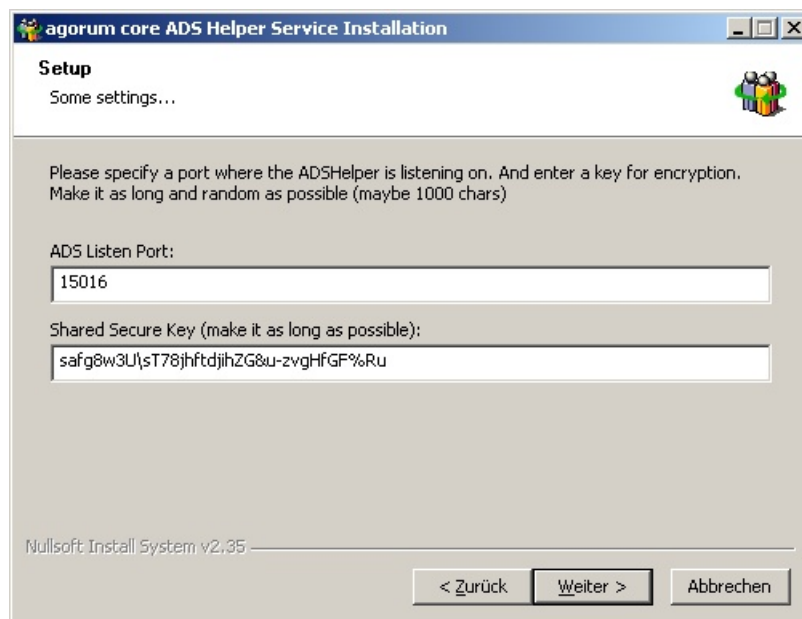


Abbildung 9.4.: Service konfigurieren

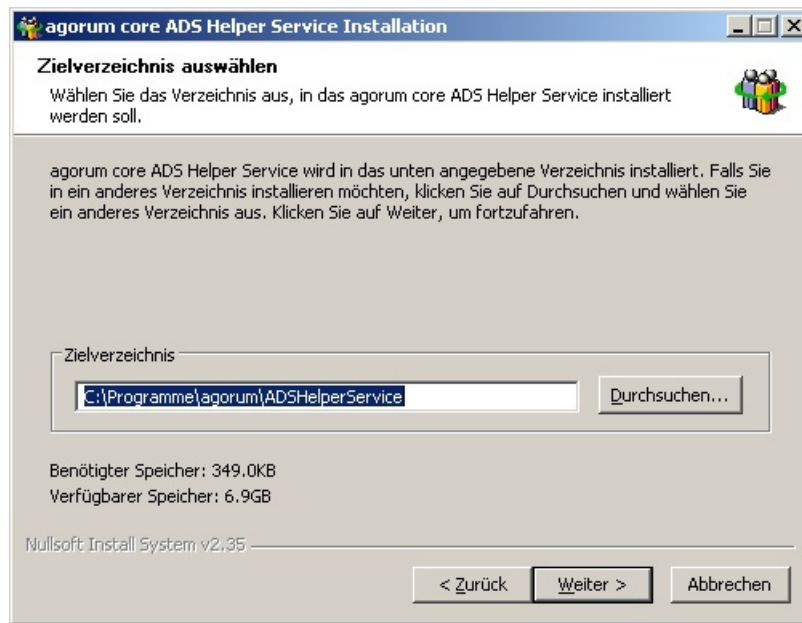


Abbildung 9.5.: Pfad auswählen

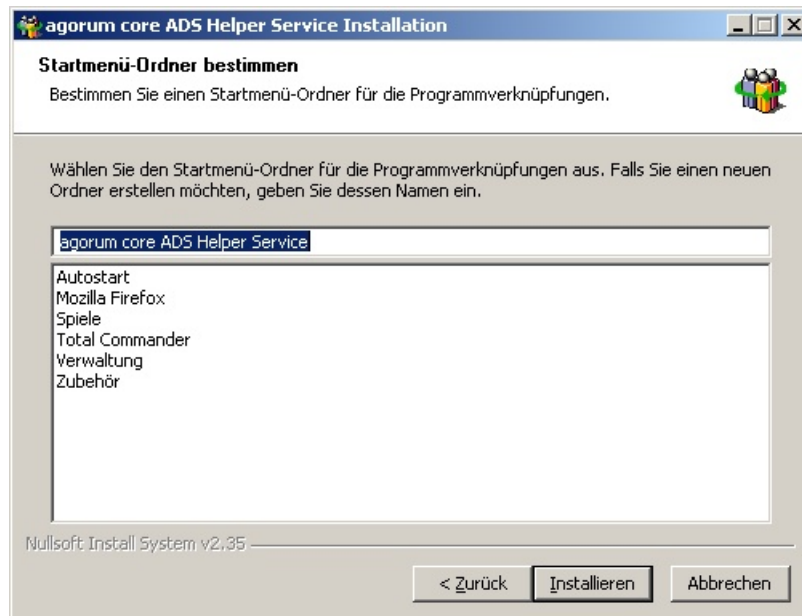


Abbildung 9.6.: Startmenü-Ordner auswählen

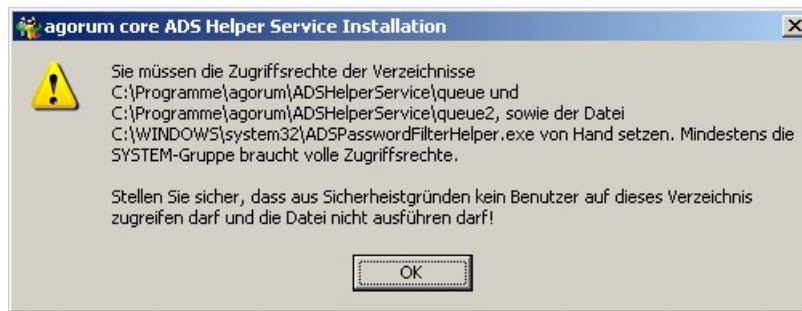


Abbildung 9.7.: Warnhinweis wegen Zugriffsrechten



Abbildung 9.8.: Server neu starten

9.2. Der WinClient Helper Service

Der WinClient Helper Services ist für die Synchronisation zu einem Microsoft Windows XP Rechner notwendig. Der Helper Service wird auf dem Windows Client installiert.

Um den WinClient Helper Service zu installieren sind folgende Schritte durchzuführen:

1. Doppelklick auf das Installationsprogramm (siehe Abbildung 9.9).
2. Auswählen des Installationspfades (siehe Abbildung 9.10).
3. Eingeben des Schlüssels und des Ports (siehe Abbildung 9.11).
4. Beenden der Installation (siehe Abbildung 9.12).

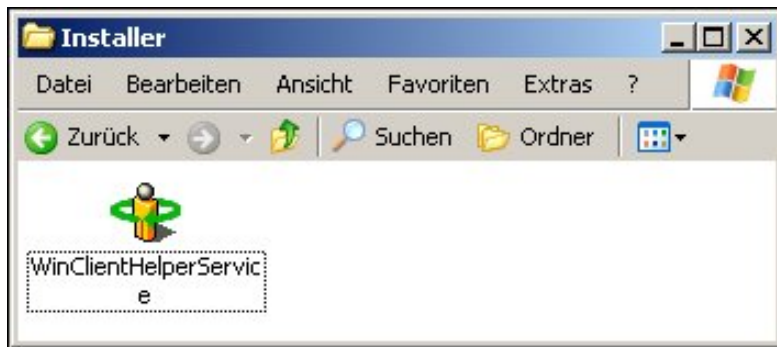


Abbildung 9.9.: Installer starten

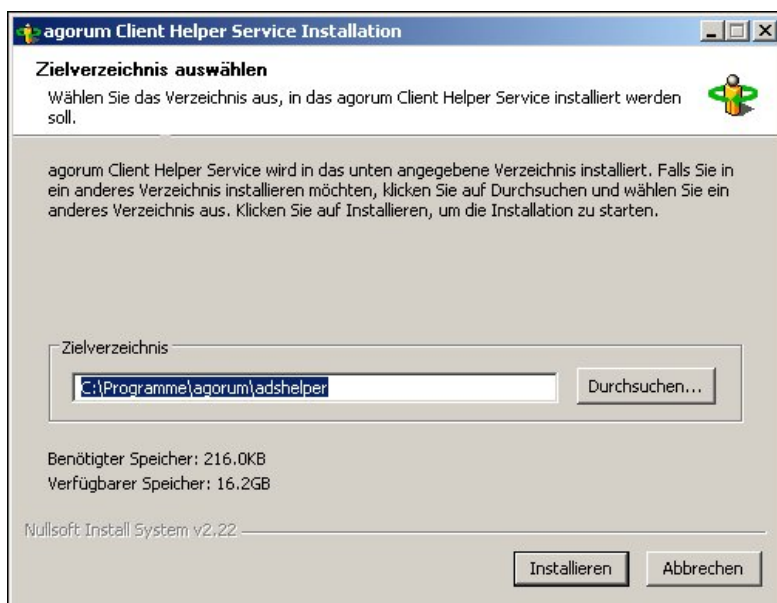


Abbildung 9.10.: Pfad auswählen

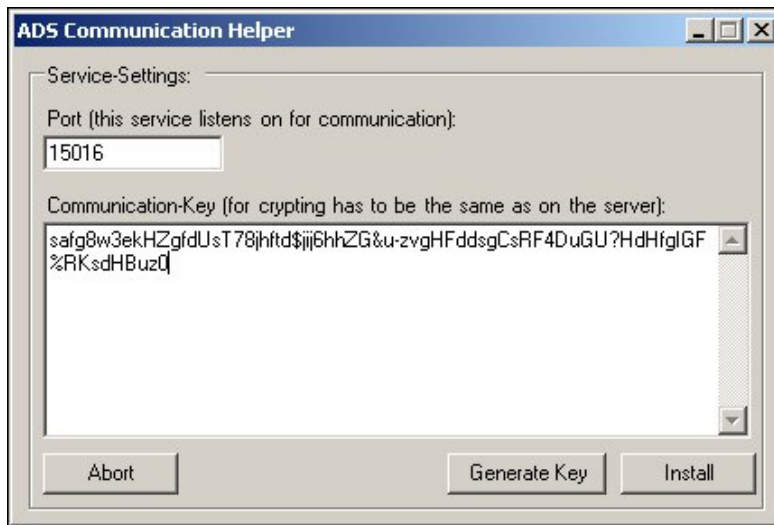


Abbildung 9.11.: Service konfigurieren

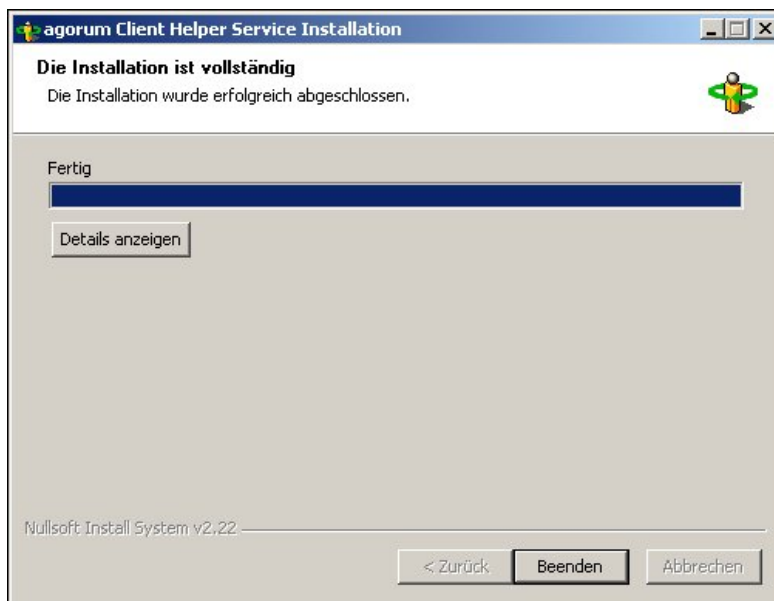


Abbildung 9.12.: Installation abschliessen

9.3. Der LDAP Helper Service

Der LDAPHelperService ist standardmäßig bei der **agorum core** Installation dabei und läuft auf dem **agorum core** Rechner, kann aber auch separat auf einem anderen Rechner installiert werden.

9.3.1. Installation mit agorum core

Die Konfiguration wird beim vorinstallierten LDAPHelperService in der XML-Datei `<InstallDir>/jboss/server/default/deploy/roi.ear/config/ldapadminsynchronservice/LDAPAdminSyncService.xml` abgelegt, bzw. bei einer separaten Installation beim Start des Services mitgegeben.

Inhalt der Konfigurationsdatei für die separate Installation:

```
1 <LDAPAdminSyncService>
2   <Port>15016</Port>
3   <CryptKey>Put in any random key for encrypting the
      communication. Has to be the same as on the remote
      system.</CryptKey>
4   <QueuePath>/opt/agorum/ldaphelperservice/queues</QueuePath
      >
5 </LDAPAdminSyncService>
```

9.3.2. Separate Installation

Der LDAPHelperService kann optional auf einem anderen Rechner installiert werden. Meist ist das nur sinnvoll, wenn Passwörter von einer Samba-Installation ins **agorum core** synchronisiert werden sollen. Auf dem Rechner muss eine Java-Runtime (min. Version 1.5) installiert sein.

Entpacken Sie dafür die Datei `install-agorum-ldaphelperservice-X_X_X.zip` in das Verzeichnis `/opt/agorum/ldaphelperservice`. Wird das Paket in einen anderen Pfad entpackt, müssen Sie die Installations-Pfade in den Dateien `scripts/setenv.sh`, `scripts/ldapadminsynchronservice.sh`, `scripts/ngospasswd.sh` und `config/LDAPAdminSyncService.xml` anpassen!

Folgende Scripte stehen zur Verfügung:

setenv.sh Datei für die Umgebungsvariablen des LDAPHelperService. In dieser Datei müssen eventuell das Installationsverzeichnis (`INSTALLPATH`), der Pfad zur Configurationsdatei (`CONFIGFILE`) und der Pfad zum Java Runtime Environment (`JAVAEXC`) angepasst werden.

ldapadminsynchronservice.sh Start-/Stop-Script des LDAPHelperService. Es kann `start` oder `stop` als Parameter übergeben werden. Bei einer Installation in

ein anderes Verzeichnis wie `/opt/agorum/ldaphelperservice` muss auch der Pfad zu `setenv.sh` angepasst werden!

ngospasswd.sh Script zum Abfangen einer Passwortänderung bei einer Samba-Konfiguration (Kapitel 9.3.3). Bei einer Installation in ein anderes Verzeichnis wie `/opt/agorum/ldaphelperservice` muss auch der Pfad zu `setenv.sh` angepasst werden!

9.3.3. Passwortsynchronisation für Samba

Mit dem Script `ngospasswd.sh` können bei einer Samba-Konfiguration (Master oder Salve) Passwortänderungen zum **agorum core** synchronisiert werden. Dafür muss in der Samba-Konfigurationsdatei `smb.conf` folgendes angepasst werden:

```
1 unix password sync = yes
2 passwd program = /Pfad/zum/Script/ngospasswd.sh %u
3 passwd chat = *Enter*NEW*password:* %n\n*Password*changed*
4 passwd chat debug = false
```

Kapitel 10.

Beschreibung der MetaDB-Properties

Übersicht aller erlaubten Properties für AdminSync in der MetaDB.

10.1. Active

(De-)Aktiviert diese Konfiguration. Wenn diese Property auf `false` steht, dann sind alle Checks ausgeschaltet und es werden keine Änderungen synchronisiert.

Verwendung

Typ	Pflicht	Optional	Ignoriert
<code>agorum core→agorum core</code>	•		
<code>agorum core→ADS</code>	•		
<code>ADS→agorum core</code>	•		
<code>agorum core→WinClient</code>	•		
<code>agorum core→LDAP</code>	•		
<code>LDAP→agorum core</code>	•		

Mögliche Werte

Wert	Beschreibung
true	Alle Checks sind eingeschaltet und Änderungen werden synchronisiert.
false	Alle Checks sind ausgeschaltet und es werden keine Änderungen synchronisiert.

10.2. AdditionalGroupSettings

Dieses Attribut findet nur beim PosixLDAP-Typ Anwendung. Es besteht so die Möglichkeit für eine Gruppe (erster Wert) die GruppenID (zweiter Wert) fest einzustellen.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>			•
agorum <i>core</i> →ADS			•
ADS→agorum <i>core</i>			•
agorum <i>core</i> →WinClient			•
agorum <i>core</i> →LDAP		•	
LDAP→agorum <i>core</i>			•

Beispiel

```
1 root:0
```

10.3. AdditionalUserSettings

Dieses Attribut findet beim PosixLDAP-Typ und beim WinClient-Typ Anwendung. Beim PosixLDAP-Typ besteht so die Möglichkeit für einen Benutzer (erster Wert) die UserID (zweiter Wert), die GruppenID (dritter Wert), den Homefolder (vierter Wert) sowie die Login-Shell (fünfter Wert) fest einzustellen. Beim

WinClient-Typ besteht die Möglichkeit den **agorum core** Benutzernamen auf einen anderen Benutzernamen auf dem Client-Rechner zu mappen.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum core→agorum core			•
agorum core→ADS			•
ADS→agorum core			•
agorum core→WinClient		•	
agorum core→LDAP		•	
LDAP→agorum core			•

Beispiel

```
1 root:0:0:/root:/bin/bash
```

oder

```
1 d4wdemo_ma:mitarbeiter
```

10.4. AdminDN

DN-Name des Admin-Benutzers auf dem LDAP-Server.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum core→agorum core			•
agorum core→ADS			•
ADS→agorum core			•
agorum core→WinClient			•
agorum core→LDAP	•		
LDAP→agorum core	•		

Beispiel

```
1 cn=Manager ,DC=agorum ,DC=com
```

10.5. AdminPw

TODO: Metadb encrypted!

Das Passwort des Admin-Benutzer auf dem LDAP-Server.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>			•
agorum <i>core</i> →ADS			•
ADS→agorum <i>core</i>			•
agorum <i>core</i> →WinClient			•
agorum <i>core</i> →LDAP	•		
LDAP→agorum <i>core</i>	•		

Beispiel

```
1 secret
```

10.6. AllowInternalPasswordChange

Dieses Property wird bei ADS→agorum *core* und beim LDAP→agorum *core* ausgewertet. Ist es auf `true` gesetzt, so können Benutzer ihr Passwort in agorum *core* ändern. Das geänderte Passwort wird aber nicht zum ADS bzw. zum LDAP zurücksynchronisiert!



Wenn `AllowInternalPasswordChange` auf `true` steht, dann sollte die Authentifizierung **nicht** am ActiveDirectory erfolgen (also `UseInternalAuthentication` **nicht** auf `false` gesetzt sein), da bei einer Passwortänderung das Passwort nicht zum ADS synchronisiert wird und die Authentifizierung dann fehlschlägt!

Verwendung

Typ	Pflicht	Optional	Ignoriert
<code>agorum core</code> → <code>agorum core</code>			•
<code>agorum core</code> →ADS			•
ADS→ <code>agorum core</code>		•	
<code>agorum core</code> →WinClient			•
<code>agorum core</code> →LDAP			•
LDAP→ <code>agorum core</code>		•	

Mögliche Werte

Wert	Beschreibung
<code>true</code>	Password kann über agorum core (z.B. über das Webportal <code>agorum desk4web</code>) geändert werden.
<code>false</code>	Password kann nicht über agorum core geändert werden.

10.7. BaseDN

Dieses Property gibt die BaseDN bei ADS- und LDAP-Servern an.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>			•
agorum <i>core</i> →ADS	•		
ADS→agorum <i>core</i>	•		
agorum <i>core</i> →WinClient			•
agorum <i>core</i> →LDAP	•		
LDAP→agorum <i>core</i>	•		

Beispiel

```
1 dc=domain ,dc=server ,dc=com
```

10.8. Class

Klasse für den entsprechenden Synchronisationstyp. Darf nicht geändert werden!

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>	•		
agorum <i>core</i> →ADS	•		
ADS→agorum <i>core</i>	•		
agorum <i>core</i> →WinClient	•		
agorum <i>core</i> →LDAP	•		
LDAP→agorum <i>core</i>	•		

Mögliche Werte

Wert	Beschreibung
agorum.ngosadminsynchron.ejb.common.NgOsAdminSyncServiceUtils	agorum <i>core</i> → agorum <i>core</i>
agorum.ngosadminsynchron.ejb.common.ADSAdminSyncServiceUtils	agorum <i>core</i> → ADS
agorum.ngosadminsynchron.ejb.common.ADSFetchAdminSyncServiceUtils	ADS → agorum <i>core</i>
agorum.ngosadminsynchron.ejb.common.WinClientAdminSyncServiceUtils	agorum <i>core</i> → Windows Client
agorum.ngosadminsynchron.ejb.common.LDAPAdminSyncServiceUtils	agorum <i>core</i> → LDAP
agorum.ngosadminsynchron.ejb.common.LDAPFetchAdminSyncServiceUtils	LDAP → agorum <i>core</i>

10.9. CnGroups

Bereich in dem die Gruppen abgelegt sind. Dieses Attribut kann bei ADS → agorum *core*-Konfigurationen leer gelassen werden. Der Gruppenbereich kann identisch zum Bereich für die Benutzer (CnUsers) sein.

Bei den Konfigurationen agorum *core* → ADS und agorum *core* → LDAP bestimmt CnGroups auch den Objekttyp des Ordners (CN für Container und OU für OrganisationalUnits) unterhalb des Gruppenbereiches, wenn Strukturen von agorum *core* synchronisiert werden sollen.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> → agorum <i>core</i>			•
agorum <i>core</i> → ADS		•	
ADS → agorum <i>core</i>	•		
agorum <i>core</i> → WinClient			•
agorum <i>core</i> → LDAP	•		
LDAP → agorum <i>core</i>	•		

Beispiel

Für ADS und LDAP:

```
1 ou=groups
```

Mit `dc=domain,dc=server,dc=com` für BaseDN und `ou=groups` für CnGroups ergibt sich für den Gruppenbereich `ou=groups,dc=domain,dc=server,dc=com`

10.10. CnUsers

Bereich in dem die Benutzer abgelegt sind. Dieses Attribut kann bei ADS→**agorum core**-Konfigurationen leer gelassen werden. Der Benutzerbereich kann identisch zum Bereich für die Gruppen (CnGroups) sein.

Bei den Konfigurationen **agorum core**→ADS und **agorum core**→LDAP bestimmt CnUsers auch den Objekttyp der Ordner (CN für Container und OU für OrganisationalUnits) unterhalb des Benutzerbereiches, wenn Strukturen von **agorum core** synchronisiert werden sollen.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum core → agorum core			•
agorum core →ADS		•	
ADS→ agorum core	•		
agorum core →WinClient			•
agorum core →LDAP	•		
LDAP→ agorum core	•		

Beispiel

Für ADS:

```
1 cn=users
```

Für LDAP:

```
1 ou=users
```

Mit `dc=domain,dc=server,dc=com` für BaseDN und `ou=users` für CnUsers ergibt sich so für den Benutzerbereich `ou=users,dc=domain,dc=server,dc=com`

10.11. ConnectString

ADS- bzw. LDAP-Server mit dem sich der HelperService verbinden soll. Kommt der ADShelperService zum Einsatz, so wird hier meist auf localhost verwiesen, da der HelperService auf dem ADS-Server läuft. Beim LDAPHelperService wird meist die IP des LDAP-Servers angegeben, da der LDAPHelperService standardmäßig auf dem **agorum core** Server läuft. Der LDAPHelperService kann aber auch auf dem LDAP-Server installiert werden (dann ist `ldap://localhost` die richtig Einstellung).

Verwendung

Typ	Pflicht	Optional	Ignoriert
<code>agorum core</code> → <code>agorum core</code>			•
<code>agorum core</code> →ADS	•		
ADS→ <code>agorum core</code>	•		
<code>agorum core</code> →WinClient			•
<code>agorum core</code> →LDAP	•		
LDAP→ <code>agorum core</code>	•		

Beispiel

```
1 ldap://10.1.2.20:389
```

oder

```
1 ldap://localhost
```

10.12. CryptKey

Zeichenkette für die verschlüsselte Datenübertragung. Je länger diese Zeichenkette ist, desto sicherer ist die Verschlüsselung.



Diese Zeichenkette muss mit der Zeichenkette auf dem Remotesystem identisch sein!

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>	•		
agorum <i>core</i> →ADS	•		
ADS→agorum <i>core</i>	•		
agorum <i>core</i> →WinClient	•		
agorum <i>core</i> →LDAP	•		
LDAP→agorum <i>core</i>	•		

Beispiel

```
1 safg8w3ekHZgfdUsT78jhftd\ $ji j6hhZG\&u-zvgHFddsgCsRF4DuGU?  
HdHfgIGF\%RKsdHBuz0
```

10.13. ExcludeUserAttributes

Attribute von Benutzern, die nicht synchronisiert werden sollen. Wenn mehrere Attribute gefiltert werden sollen, werden diese mit zwei Pipes (||) getrennt.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>	•		
agorum <i>core</i> →ADS		•	
ADS→agorum <i>core</i>		•	
agorum <i>core</i> →WinClient		•	
agorum <i>core</i> →LDAP		•	
LDAP→agorum <i>core</i>		•	

Mögliche Werte

Wert	Unterstützte Konfigurationen	Beschreibung
CredentialManager	agorum <i>core</i> → agorum <i>core</i>	Muss bei agorum <i>core</i> → agorum <i>core</i> gesetzt sein!
AdminEnabled	agorum <i>core</i> → agorum <i>core</i>	
GivenName	Alle	
FamilyName	Alle	
EmailAddresses	Alle	
Language	agorum <i>core</i> → agorum <i>core</i> , agorum <i>core</i> → ADS, ADS → agorum <i>core</i>	
MandatorIdentifier	agorum <i>core</i> → agorum <i>core</i>	
Description	Alle	
Lockstate	agorum <i>core</i> → agorum <i>core</i> , agorum <i>core</i> → ADS, ADS → agorum <i>core</i>	
IsRole	agorum <i>core</i> → agorum <i>core</i>	
DefaultRoleName		
IsAssociatedRolesNameSet	agorum <i>core</i> → agorum <i>core</i>	
AssociatedRolesNameName	agorum <i>core</i> → agorum <i>core</i>	
Aliasesset	agorum <i>core</i> → agorum <i>core</i>	
Aliases	agorum <i>core</i> → agorum <i>core</i>	

Beispiel

```
1 CredentialManager || Description
```

10.14. FlatFolderStructure

Benutzer und Gruppen werden flach, ohne Unterordner abgelegt.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>			•
agorum <i>core</i> →ADS		•	
ADS→agorum <i>core</i>			•
agorum <i>core</i> →WinClient			•
agorum <i>core</i> →LDAP		•	
LDAP→agorum <i>core</i>			•

Mögliche Werte

Wert	Beschreibung
true	User und Gruppen werden flach, ohne Unterordner abgelegt.
false	User und Gruppen werden in evtl vorhandenen Unterordnern abgelegt.

10.15. GroupType

Dieses Attribute wird nur bei der ADS Synchronisation ausgewertet. Mit diesem Property kann der Gruppentyp bestimmt werden. Dabei darf nur eins der vier Werte LOCAL_GROUP, DOMAIN_LOCAL_GROUP, GLOBAL_GROUP und UNIVERSAL_GROUP verwendet werden. Der Wert SECURITY_ENABLED kann zusätzlich übergeben werden, um eine Security-Gruppe zu erzeugen. Ist dieser Wert nicht gesetzt, wird eine Verteilergruppe erzeugt.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>			•
agorum <i>core</i> →ADS		•	
ADS→agorum <i>core</i>			•
agorum <i>core</i> →WinClient			•
agorum <i>core</i> →LDAP			•
LDAP→agorum <i>core</i>			•

Mögliche Werte

Wert	Beschreibung
LOCAL_GROUP	
DOMAIN_LOCAL_GROUP	
GLOBAL_GROUP	
UNIVERSAL_GROUP	
SECURITY_ENABLED	

Standard Wert

1 GLOBAL_GROUP || SECURITY_ENABLED

Beispiel

1 UNIVERSAL_GROUP || SECURITY_ENABLED

10.16. History

Erzeugt eine History für diese Konfiguration. Die History wird im AdminSync-Bereich (*Administration* → *AdminSync*) in dem Ordner `history`, unterhalb der jeweiligen Konfiguration gespeichert.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>		•	
agorum <i>core</i> →ADS		•	
ADS→agorum <i>core</i>		•	
agorum <i>core</i> →WinClient		•	
agorum <i>core</i> →LDAP		•	
LDAP→agorum <i>core</i>		•	

Mögliche Werte

Wert	Beschreibung
true	Alle übertragenen XML-Objekte werden historisiert.
false	Die übertragenen XML-Objekte werden nicht historisiert

10.17. LocalServer

Lokaler Server für eine ADS→agorum *core* oder LDAP→agorum *core* Konfiguration. Der Server wird unter *MAIN_SERVER_MANAGEMENT* konfiguriert und muss auf den lokalen agorum *core* Server verweisen.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>			•
agorum <i>core</i> →ADS			•
ADS→agorum <i>core</i>	•		
agorum <i>core</i> →WinClient			•
agorum <i>core</i> →LDAP			•
LDAP→agorum <i>core</i>	•		

Standard Wert

10.18. LockInsteadOfDelete

Dieses Property wird nur bei ADS→**agorum core** und bei LDAP→**agorum core** ausgewertet. Ist es auf `true` gesetzt, so werden im ADS bzw. im LDAP gelöschte Benutzer in **agorum core** nur gesperrt. Im ADS/LDAP gelöschte Gruppen bleiben in **agorum core** unverändert.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum core → agorum core			•
agorum core →ADS			•
ADS→ agorum core		•	
agorum core →WinClient			•
agorum core →LDAP			•
LDAP→ agorum core		•	

Mögliche Werte

Wert	Beschreibung
<code>true</code>	Benutzer werden nicht gelöscht, sondern gesperrt.
<code>false</code>	Benutzer und Gruppen werden gelöscht

10.19. NgOsPathOffset

Pfadstück das im Mastersystem entfernt wird.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>		•	
agorum <i>core</i> →ADS		•	
ADS→agorum <i>core</i>		•	
agorum <i>core</i> →WinClient		•	
agorum <i>core</i> →LDAP		•	
LDAP→agorum <i>core</i>		•	

Beispiel

1 d4wdemo

So wird aus User/d4wdemo/d4wdemo_ma auf dem Zielsystem User/d4wdemo_ma.

10.20. NoGroupInGroup

Sind auf der **agorum core** Seite Gruppen in Gruppen vorhanden, so werden die Gruppen in Benutzer aufgelöst. Damit besteht die Möglichkeit Gruppen in Gruppen in **agorum core** zu administrieren, auch wenn das sich authentifizierende System nur mit Usern in Gruppen umgehen kann (z.B. Samba).

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>		•	
agorum <i>core</i> →ADS		•	
ADS→agorum <i>core</i>			•
agorum <i>core</i> →WinClient			•
agorum <i>core</i> →LDAP		•	
LDAP→agorum <i>core</i>			•

Mögliche Werte

Wert	Beschreibung
true	Nur User werden in Gruppen abgelegt.
false	User und Gruppen werden in Gruppen abgelegt.

10.21. NotSyncPathControl

Pfade/Objekte die nicht überwacht werden sollen. Beginnt immer mit **User/** für den Benutzerbereich, mit **Group/** für den Gruppenbereich und mit **Role/** für den ACL-Bereich. Es können nicht nur Pfade angegeben werden, sondern auch einzelne User, Gruppen oder ACLs.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>		•	
agorum <i>core</i> →ADS		•	
ADS→agorum <i>core</i>		•	
agorum <i>core</i> →WinClient		•	
agorum <i>core</i> →LDAP		•	
LDAP→agorum <i>core</i>		•	

Beispiel

```
1 User/d4wdemo/d4wdemo_gl||Group/d4wdemo/
   d4wdemoGeschaeftsfuehrung
```

10.22. ObjectFactory

Dieses Attribute wird nur bei der LDAP Synchronisation ausgewertet. In dieser Klasse sind die Lesefunktionen der LDAP-Attribute implementiert.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>			•
agorum <i>core</i> →ADS			•
ADS→agorum <i>core</i>			•
agorum <i>core</i> →WinClient			•
agorum <i>core</i> →LDAP	•		
LDAP→agorum <i>core</i>	•		

Mögliche Werte

- agorum.ldapadminsynchronservice.factories.DefaultLdapObjectFactory

Diese Factory wird zum Synchronisieren von einfachen LDAP-Objekten benutzt. Es gelten folgende Eigenschaften:

Eigenschaft	Objektyp / Attribut
Benutzer ObjectClass	inetOrgPerson
Gruppe ObjectClass	groupOfNames
Benutzername Attribut	cn
Gruppenname Attribut	cn
Gruppenmitglied Attribut	member

Folgende Attribute für Benutzer werden synchronisiert:

Attribut		vom	ins
LDAP	agorum <i>core</i>	LDAP	LDAP
cn	Name	•	•
sn	Familiennamen	•	•
givenname	Vorname	•	•
description	Beschreibung	•	•
userPassword	Passwort		•
preferredLanguage	Sprache	•	•
mail	E-Mail-Adresse	•	•

Folgende Attribute für Gruppen werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum core		
cn	Name	•	•
description	Beschreibung	•	•
member	Gruppenmitglieder	•	•

- `agorum.ldapadminsynchronservice.factories.GroupOfUniqueNamesLdapObjectFactory`

Diese Factory wird zum Synchronisieren von einfachen LDAP-Objekten benutzt. Es gelten folgende Eigenschaften:

Eigenschaft	Objekttyp / Attribut
Benutzer ObjectClass	<code>inetOrgPerson</code>
Gruppe ObjectClass	<code>groupOfUniqueNames</code>
Benutzername Attribut	<code>cn</code>
Gruppenname Attribut	<code>cn</code>
Gruppenmitglied Attribut	<code>uniqueMember</code>

Folgende Attribute für Benutzer werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum core		
cn	Name	•	•
sn	Familiennamenname	•	•
givenname	Vorname	•	•
description	Beschreibung	•	•
userPassword	Passwort		•
preferredLanguage	Sprache	•	•
mail	E-Mail-Adresse	•	•

Folgende Attribute für Gruppen werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum core		
cn	Name	•	•
description	Beschreibung	•	•
uniqueMember	Gruppenmitglieder	•	•

- `agorum.ldapadminsynchronservice.factories.UidLdapObjectFactory`

Diese Factory wird zum Synchronisieren von einfachen LDAP-Objekten benutzt. Es gelten folgende Eigenschaften:

Eigenschaft	Objekttyp / Attribut
Benutzer ObjectClass	inetOrgPerson
Gruppe ObjectClass	groupOfNames
Benutzername Attribut	uid
Gruppenname Attribut	cn
Gruppenmitglied Attribut	member

Folgende Attribute für Benutzer werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum core		
uid	Name	•	•
sn	Familiennamenname	•	•
givenname	Vorname	•	•
description	Beschreibung	•	•
userPassword	Passwort		•
preferredLanguage	Sprache	•	•
mail	E-Mail-Adresse	•	•

Folgende Attribute für Gruppen werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum core		
cn	Name	•	•
description	Beschreibung	•	•
member	Gruppenmitglieder	•	•

- `agorum.ldapadminsynchronservice.factories.PosixLdapObjectFactory`

Diese Factory wird zum Synchronisieren von POSIX-LDAP-Objekten benutzt. Es gelten folgende Eigenschaften:

Eigenschaft	Objekttyp / Attribut
Benutzer ObjectClass	posixAccount
Gruppe ObjectClass	posixGroup
Benutzername Attribut	uid
Gruppenname Attribut	cn
Gruppenmitglied Attribut	memberUid

Folgende Attribute für Benutzer werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum core		
uid	Name	•	•
sn	Familiennamen	•	•
givenname	Vorname	•	•
description	Beschreibung	•	•
userPassword	Passwort		•
preferredLanguage	Sprache	•	•
mail	E-Mail-Adresse	•	•
uidNumber	–		•
gidNumber	–		•
homeDirectory	–		•
loginShell	–		•

Folgende Attribute für Gruppen werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum core		
cn	Name	•	•
description	Beschreibung	•	•
memberUid	Gruppenmitglieder	•	•
gidNumber	–		•

- `agorum.ldapadminsynchronservice.factories.SambaLdapObjectFactory`

Diese Factory wird zum Synchronisieren von Samba-LDAP-Objekten benutzt. Es gelten folgende Eigenschaften:

Eigenschaft	Objekttyp / Attribut
Benutzer ObjectClass	<code>sambaSamAccount</code>
Gruppe ObjectClass	<code>sambaGroupMapping</code>
Benutzername Attribut	<code>cn</code>
Gruppenname Attribut	<code>cn</code>
Gruppenmitglied Attribut	<code>memberUid</code>

Folgende Attribute für Benutzer werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum core		
cn	Name	•	•
sn	Familiennamen	•	•
givenname	Vorname	•	•
description	Beschreibung	•	•
userPassword	Passwort		•
preferredLanguage	Sprache	•	•
mail	E-Mail-Adresse	•	•
uidNumber	–		•
gidNumber	–		•
homeDirectory	–		•
loginShell	–		•
sambaAcctFlags	Gesperrt	•	•
sambaLMPasswort	Passwort		•
sambaNTPasswort	Passwort		•
sambaSID	–		•

Folgende Attribute für Gruppen werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum core		
cn	Name	•	•
description	Beschreibung	•	•
memberUid	Gruppenmitglieder	•	•
gidNumber	–		•
sambaSID	–		•

- `agorum.ldapadminsynchronservice.factories.Samba2LdapObjectFactory`

Diese Factory wird zum Synchronisieren von Samba-LDAP-Objekten benutzt. Es gelten folgende Eigenschaften:

Eigenschaft	Objekttyp / Attribut
Benutzer ObjectClass	<code>sambaSamAccount</code>
Gruppe ObjectClass	<code>posixGroup</code>
Benutzername Attribut	<code>uid</code>
Gruppenname Attribut	<code>cn</code>
Gruppenmitglied Attribut	<code>memberUid</code>

Folgende Attribute für Benutzer werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum core		
uid	Name	•	•
sn	Familiennamen	•	•
givenname	Vorname	•	•
description	Beschreibung	•	•
userPassword	Passwort		•
preferredLanguage	Sprache	•	•
mail	E-Mail-Adresse	•	•
uidNumber	–		•
gidNumber	–		•
homeDirectory	–		•
loginShell	–		•
sambaAcctFlags	Gesperrt	•	•
sambaLMPasswort	Passwort		•
sambaNTPasswort	Passwort		•
sambaSID	–		•

Folgende Attribute für Gruppen werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum core		
cn	Name	•	•
description	Beschreibung	•	•
memberUid	Gruppenmitglieder	•	•
gidNumber	–		•

- `agorum.ldapadminsynchronservice.factories.UidSambaLdapObjectFactory`

Diese Factory wird zum Synchronisieren von Samba-LDAP-Objekten benutzt. Es gelten folgende Eigenschaften:

Eigenschaft	Objekttyp / Attribut
Benutzer ObjectClass	<code>sambaSamAccount</code>
Gruppe ObjectClass	<code>sambaGroupMapping</code>
Benutzername Attribut	<code>uid</code>
Gruppenname Attribut	<code>cn</code>
Gruppenmitglied Attribut	<code>memberUid</code>

Folgende Attribute für Benutzer werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum <i>core</i>		
uid	Name	•	•
sn	Familiennamen	•	•
givenname	Vorname	•	•
description	Beschreibung	•	•
userPassword	Passwort		•
preferredLanguage	Sprache	•	•
mail	E-Mail-Adresse	•	•
uidNumber	–		•
gidNumber	–		•
homeDirectory	–		•
loginShell	–		•
sambaAcctFlags	Gesperrt	•	•
sambaLMPasswort	Passwort		•
sambaNTPasswort	Passwort		•
sambaSID	–		•

Folgende Attribute für Gruppen werden synchronisiert:

Attribut		vom LDAP	ins LDAP
LDAP	agorum <i>core</i>		
cn	Name	•	•
description	Beschreibung	•	•
memberUId	Gruppenmitglieder	•	•
gidNumber	–		•
sambaSID	–		•

10.23. ParameterNames / ParameterValues

Diese beiden Attribute gehören zusammen und findet nur beim PosixLDAP-Typ anwendung.

StartUIdNumber bestimmt den Startwert für Posix-User-Ids.

StartGidNumber bestimmt den Startwert für Posix-Gruppen-Ids.

UserGidNumber bestimmt die Posix-Gruppen-Id eines Benutzers.

HomeDirectoryPrefix ist der Ordner, in dem die einzelnen privaten Ordner liegen.

LoginShell bestimmt die Login-Shell für Posix-User.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>			•
agorum <i>core</i> →ADS			•
ADS→agorum <i>core</i>			•
agorum <i>core</i> →WinClient			•
agorum <i>core</i> →LDAP		•	
LDAP→agorum <i>core</i>			•

Mögliche Werte

ParameterNames	ParameterValues
StartUidNumber	1500
StartGidNumber	1500
UserGidNumber	100
HomeDirectoryPrefix	/home
LoginShell	/bin/bash

10.24. RemotePathOffset

Pfadstück das im Slavesystem vor den Pfad angehängt wird.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>		•	
agorum <i>core</i> →ADS		•	
ADS→agorum <i>core</i>		•	
agorum <i>core</i> →WinClient		•	
agorum <i>core</i> →LDAP		•	
LDAP→agorum <i>core</i>		•	

Beispiel

1 Ng0s

So wird aus User/d4wdemo/d4wdemo_ma auf dem Zielsystem User/Ng0s/d4wdemo/d4wdemo_ma.

10.25. ReplicaServers

Alle ADS- bzw. LDAP/Samba-Replikationsserver für diese Konfiguration. Die Server werden unter MAIN_SERVER_MANAGEMENT konfiguriert.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>			•
agorum <i>core</i> →ADS		•	
ADS→agorum <i>core</i>		•	
agorum <i>core</i> →WinClient			•
agorum <i>core</i> →LDAP		•	
LDAP→agorum <i>core</i>		•	

Beispiel

1 ADSReplicaServer1 || ADSReplicaServer2

10.26. Server

Remote-Server für diese Konfiguration. Der Server wird unter MAIN_SERVER_MANAGEMENT konfiguriert.

Verwendung

Typ	Pflicht	Optional	Ignoriert
<i>agorum core</i> → <i>agorum core</i>	•		
<i>agorum core</i> →ADS	•		
ADS→ <i>agorum core</i>	•		
<i>agorum core</i> →WinClient	•		
<i>agorum core</i> →LDAP	•		
LDAP→ <i>agorum core</i>	•		

Beispiel

1 Ng0sServer

10.27. SocketTimeout

Der Standardwert ist 30000 und wird in Millisekunden angegeben. Er definiert den Timeout für die Verbindung zum ADS-/LDAP-HelperService. Dieser Wert muss im Normalfall nicht geändert werden.

Verwendung

Typ	Pflicht	Optional	Ignoriert
<i>agorum core</i> → <i>agorum core</i>			•
<i>agorum core</i> →ADS		•	
ADS→ <i>agorum core</i>		•	
<i>agorum core</i> →WinClient		•	
<i>agorum core</i> →LDAP		•	
LDAP→ <i>agorum core</i>		•	

Standard Wert

1 30000

10.28. StateFactory

Dieses Attribute wird nur bei der LDAP Synchronisation ausgewertet. In dieser Klasse sind die Schreibfunktionen der LDAP-Attribute implementiert.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> → agorum <i>core</i>			•
agorum <i>core</i> → ADS			•
ADS → agorum <i>core</i>			•
agorum <i>core</i> → WinClient			•
agorum <i>core</i> → LDAP	•		
LDAP → agorum <i>core</i>	•		

Mögliche Werte

Hier die Möglichen Klasse für die *StateFactory*. Eine genaue Beschreibung finden Sie bei den *ObjectFactories*.

- `agorum.ldapadminsyncservice.factories.DefaultLdapStateFactory`
- `agorum.ldapadminsyncservice.factories.GroupOfUniqueNamesLdapStateFactory`
- `agorum.ldapadminsyncservice.factories.UidLdapStateFactory`
- `agorum.ldapadminsyncservice.factories.PosixLdapStateFactory`
- `agorum.ldapadminsyncservice.factories.SambaLdapStateFactory`
- `agorum.ldapadminsyncservice.factories.Samba2LdapStateFactory`
- `agorum.ldapadminsyncservice.factories.UidSambaLdapStateFactory`

10.29. SyncPathControl

Pfade/Objekte die überwacht werden sollen. Beginnt immer mit `User/` für den Benutzerbereich, mit `Group/` für den Gruppenbereich und mit `Role/` für den ACL-Bereich. Es können nicht nur Pfade angegeben werden, sondern auch einzelne User, Gruppen oder ACLs.

Verwendung

Typ	Pflicht	Optional	Ignoriert
<code>agorum core→agorum core</code>	•		
<code>agorum core→ADS</code>	•		
<code>ADS→agorum core</code>	•		
<code>agorum core→WinClient</code>	•		
<code>agorum core→LDAP</code>	•		
<code>LDAP→agorum core</code>	•		

Beispiel

```
1 User/d4wdemo || Group/d4wdemo || Role/d4wdemo
```

10.30. TransactionTimeout

Der Standardwert ist 300000 und wird in Millisekunden angegeben. Er definiert den Timeout für die Webservice-Verbindung. Dieser Wert muss im Normalfall nicht geändert werden.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>		•	
agorum <i>core</i> →ADS			•
ADS→agorum <i>core</i>		•	
agorum <i>core</i> →WinClient			•
agorum <i>core</i> →LDAP			•
LDAP→agorum <i>core</i>		•	

Standard Wert

1 300000

10.31. UPNDomainName

Dieses Attribute wird nur bei der ADS Synchronisation ausgewertet. Mit diesem Property kann der Suffix des User Principal Name angegeben werden. Ist dieses Property nicht angegeben wird der Suffix aus dem Property BaseDN zusammengesetzt.

Verwendung

Typ	Pflicht	Optional	Ignoriert
agorum <i>core</i> →agorum <i>core</i>			•
agorum <i>core</i> →ADS		•	
ADS→agorum <i>core</i>			•
agorum <i>core</i> →WinClient			•
agorum <i>core</i> →LDAP			•
LDAP→agorum <i>core</i>			•

Beispiel

1 server.com

Mit `server.com` für `UPNDomainName` und dem Benutzernamen `max.mustermann` ergibt sich für den User Principal Name `max.mustermann@server.com`

Das Property `UPNDomainName` ist nicht angegeben, `dc=domain,dc=server,dc=com` für `BaseDN`, und dem Benutzernamen `max.mustermann` ergibt sich für den User Principal Name `max.mustermann@domain.server.com`

10.32. UseInternalAuthentication

Dieses Property wird nur beim ADS-Fetch ausgewertet. Ist es auf `true` gesetzt, so werden Benutzer nicht am ADS authentifiziert, sondern intern. Dazu muss man aber beachten, dass das Passwort intern gesetzt ist (entweder über das Property `AllowInternalPasswordChange` oder mit dem Passwort-Fetch CronJob).



Wenn `UseInternalAuthentication` auf `false` steht (Authentifizierung am ActiveDirectory), dann sollte das in **agorum core** geändert werden können (`AllowInternalPasswordChange` muss auf `true` stehen)! Wenn `AllowInternalPasswordChange` auf `false` gesetzt ist, dann wird bei einer Passwortänderung das Passwort nicht zum ADS synchronisiert und die Authentifizierung schlägt fehl!

Verwendung

Typ	Pflicht	Optional	Ignoriert
<code>agorum core</code> → <code>agorum core</code>			•
<code>agorum core</code> →ADS			•
ADS→ <code>agorum core</code>		•	
<code>agorum core</code> →WinClient			•
<code>agorum core</code> →LDAP			•
LDAP→ <code>agorum core</code>			•

Mögliche Werte

Wert	Beschreibung
<code>true</code>	Es wird im agorm core authentifiziert.
<code>false</code>	Es wird im ADS authenifiziert.

Abbildungsverzeichnis

1.1. Allgemeine Funktionsweise von agorum <i>adminSync</i>	6
3.1. Synchronisation von agorum <i>core</i> zu einem anderen agorum <i>core</i>	12
4.1. Synchronisation von agorum <i>core</i> zu einem ADS	16
5.1. Synchronisation von einem ADS zu agorum <i>core</i>	23
6.1. Synchronisation von agorum <i>core</i> zu einem Windows Client	31
7.1. Synchronisation von agorum <i>core</i> zu LDAP	35
8.1. Synchronisation von LDAP zu agorum <i>core</i>	52
9.1. Auswahl der Sprache	61
9.2. Installer starten	61
9.3. Lizenz bestätigen	62
9.4. Service konfigurieren	62
9.5. Pfad auswählen	63
9.6. Startmenü-Ordner auswählen	63
9.7. Warnhinweis wegen Zugriffsrechten	64
9.8. Server neu starten	64
9.9. Installer starten	65
9.10. Pfad auswählen	65
9.11. Service konfigurieren	66
9.12. Installation abschliessen	66

Anhang A.

Versions-Historie dieses Dokumentes

Version 1.0.0: Erstellung der ersten Dokumenten-Version.

Version 1.1.0: Beschreibung von ReplicaServern.

Version 1.1.1: Beschreibung der Initialsynchronisation (Passwörter) und Passwortwechsel bei Samba, sowie Scripte des LDAPHelperServices.

Version 1.1.2: Fehler in der Property-Beschreibung von `CnGroups` und `CnUsers` behoben. Property-Beschreibung von `AllowInternalPasswordChange` und `UseInternalAuthentication` erweitert.

Version 1.1.3: Fehlende LDAP-Factories ergänzt und alle Factories beschrieben.

Version 1.2.0: Beschreibung des Konfigurators.

Anhang B.

Sonstiges

Soweit in dieser Dokumentation Marken- oder Produktbezeichnungen verwendet werden, unterliegen diese ausschließlich den Schutzrechten des Inhabers, auch wenn dies nicht explizit kenntlich gemacht wird.