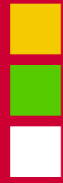


Univention Active Directory Connector in der Praxis

 **univention**
linux for your business

Univention Partner Summit 2010

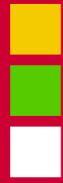
Ingo Steuer, steuer@univention.de



Agenda

- 1) **Einführung, Anwendungsszenarien**
- 2) Technischer Aufbau
- 3) Praxis
 - a) Basiseinrichtung
 - b) Betrieb: Überwachung, Fehleranalyse
 - c) Sonderfälle: Neusynchronisation, erweiterte Konfiguration
- 4) Hinweise und Ausblick
- 5) Fragen/Diskussion





UCS Active Directory Connector

bidirektionale, selektive Synchronisation



der Benutzerkonten

der Gruppendefinitionen

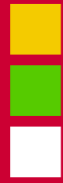
der Passwörter

von beliebigen, weiteren LDAP-Objekten möglich

Migration - einfache Übernahme von Benutzern, Gruppen und anderen Objekten aus Active Directory.

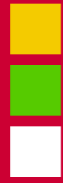
Interoperabilität - paralleler Einsatz beider Verzeichnisdienste; zu jedem beliebigen Zeitpunkt;

Integration - langsame, schrittweise Einführung; einfache und wirtschaftliche Integration Linux-basierte Dienste und Anwendungen;



Häufige Einsatzszenarien

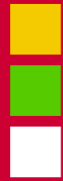
- Integration von UCS in einer AD-Umgebung als Basis für
 - Nutzung von UCS Services/Komponenten in einer AD-Umgebung (Häufigster Fall: Mail/Groupware)
 - Linux/UNIX-Authentifikations-Services (POSIX)
 - Synchronisation AD → UCS
- Integration von AD in einer UCS-Umgebung als Basis für
 - Fachanwendungen mit AD-Anforderung
 - Synchronisation UCS → AD
- Migration
 - Synchronisation UCS ↔ AD



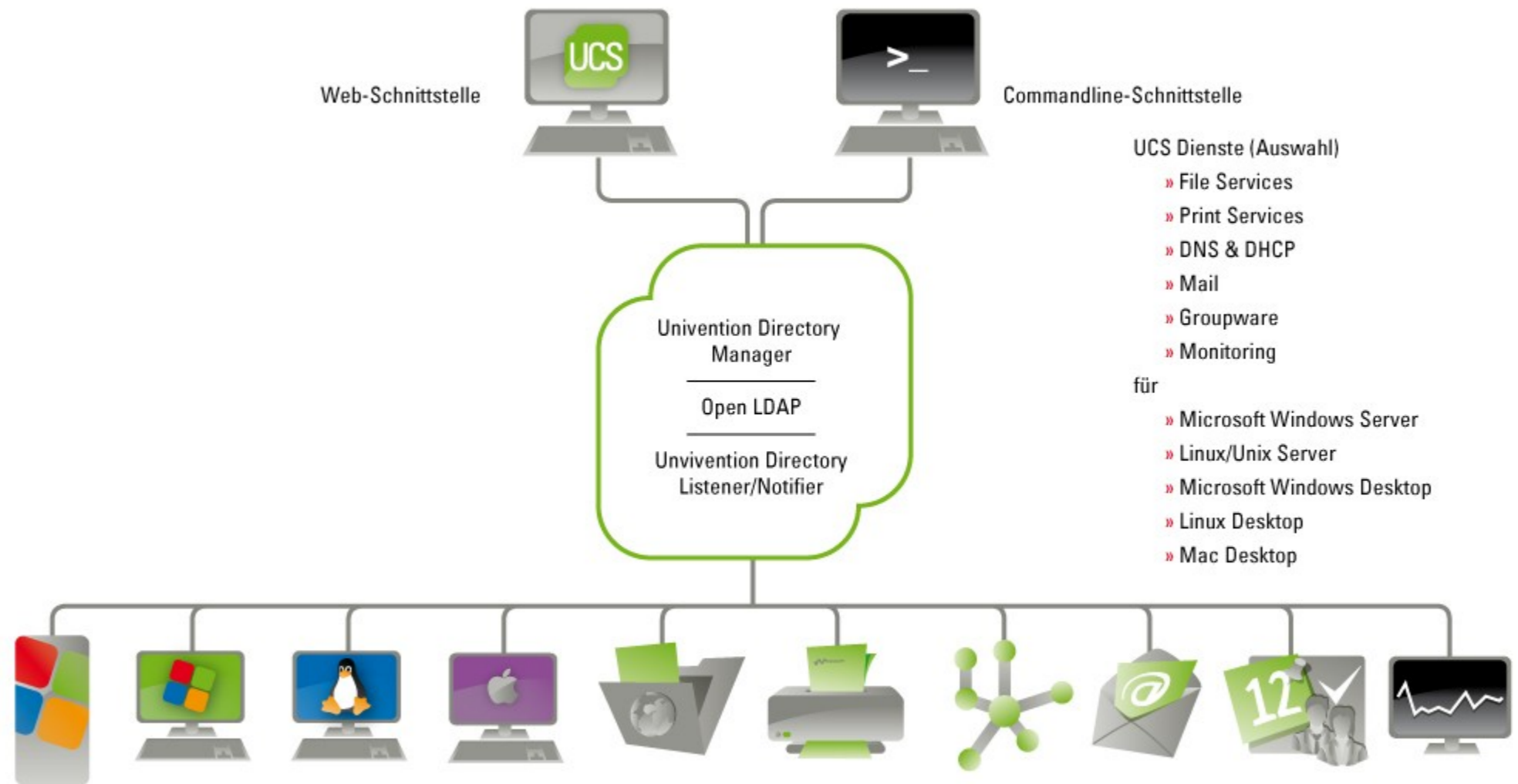
Agenda

- 1) Einführung, Anwendungsszenarien
- 2) Technischer Aufbau**
- 3) Praxis
 - a) Basiseinrichtung
 - b) Betrieb: Überwachung, Fehleranalyse
 - c) Sonderfälle: Neusynchronisation, erweiterte Konfiguration
- 4) Hinweise und Ausblick
- 5) Fragen/Diskussion

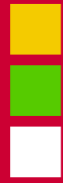




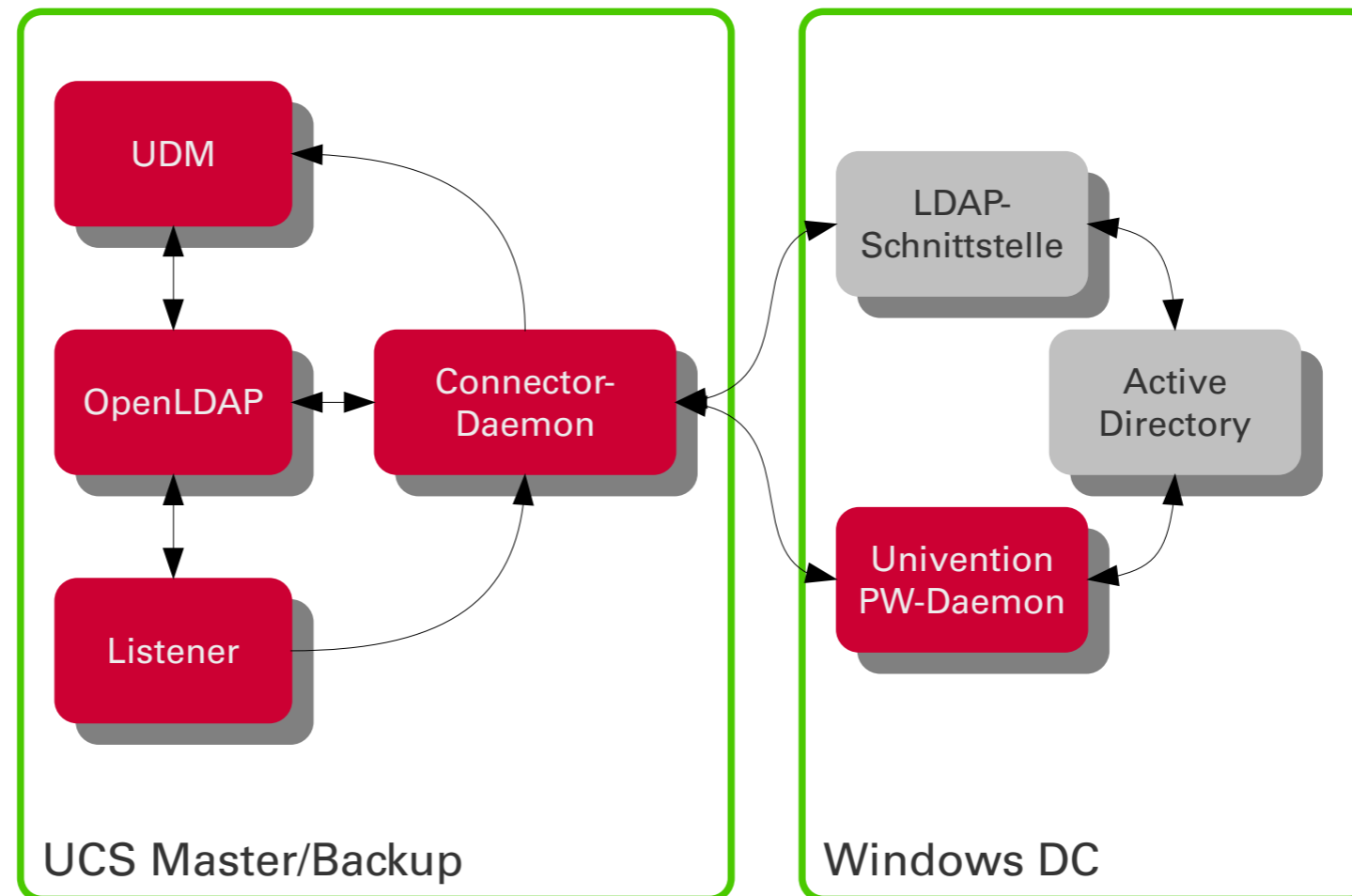
UCS-Managementsystem

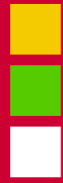


Alle Optionen über Skript- und API-Schnittstellen verfügbar



Schematischer Aufbau

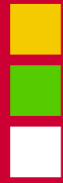




Wichtige Konfigurationsdateien

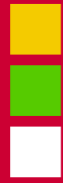
- `/etc/univention/connector/ad/mapping`
 - Definition der zu synchronisierenden Objekte
 - Zuordnung UDM-Attribute ↔ AD-Attribute
 - Erweiterte Einstellungen: Filter, Skripte, ...
 - Basiskonfiguration per UCR (z.B. Windows 2000/2003/2008), manuelle Konfiguration direkt in der Datei
 - Wird beim (Neu-)Start des Connectors in die `mapping.py` ausgewertet, daher können direkt UCR-Template-Ersetzungen genutzt werden

- `/etc/univention/connector/internal.cfg`
 - Interner Status und DN-Mapping-Cache des Connectors
 - Nicht manuell änderbar, muss für Resync AD → UCS gelöscht werden



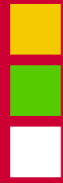
Ablauf der Synchronisation: Allgemein

- Alle Änderungen werden Objekt- und Event-Basiert synchronisiert, d.H.
 - Eine Änderung an einem LDAP-Objekt bewirkt einen Ablauf der Synchronisation
 - Ein UDM-Objekt unter UCS wird einem LDAP-Objekt im AD zugeordnet
- Während der Synchronisation eines Objektes werden
 - UDM-Attribute mit AD-Attributen „gemappt“, dazu kann auch direkt auf OpenLDAP zurückgegriffen werden
 - Attribut-Änderungen an UCS per UDM durchgeführt (auf Basis der Python-Bibliotheken)
 - Attribut-Änderungen an AD über die LDAP-Schnittstelle durchgeführt
 - Passwort-Änderungen mit AD über einen separaten Dienst und unter UCS direkt im LDAP durchgeführt



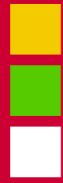
Synchronisation: Intervalle

- Der Connector prüft in Intervallen, ob Änderungen von UCS oder AD vorliegen
 - UCS: Neue Export-Dateien des Listeners?
 - AD: Höhere highestCommittedUSN am LDAP-Basisobjekt?
- Vorliegende Änderungen werden abgearbeitet (erst AD → UCS, dann UCS → AD)
- Wurden Änderungen eingespielt wird erneut auf neue Änderungen geprüft
- Wurden keine Änderungen gefunden, ruht der Connector für die in der UCR-Variable „connector/ad/poll/sleep“ gesetzte Zahl von Sekunden (Default: 5)
- Wurden wiederholt keine Änderungen gefunden, versucht der Connector „Rejects“ (s.u.) einzuspielen, die Anzahl definiert die UCR-Variable „connector/ad/retryrejected“ (Default: 10)



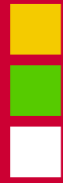
Ablauf der Synchronisation: AD → UCS

- Der Connector-Daemon fragt in Intervallen im AD die „highestCommittedUSN“ ab
- Ist diese größer der intern gespeicherten, werden alle dazwischen liegenden Änderungen vom AD abgefragt
- Eingehende Objekte werden anhand der Filter in der Connector-Konfiguration UDM-Objekten zugeordnet (Container, OU, Benutzer, Gruppe)
 - Das zugehörige UDM-Objekt wird gesucht und geöffnet, ist es nicht vorhanden wird es initialisiert
 - Das Mapping wird gemäß Konfigurationsdatei für alle Attribute durchgeführt
 - Das UDM-Objekt wird gespeichert
 - Die „Post Modify“ und ggf. die „Post Create“ Funktionen werden aufgerufen, eine der Funktionen synchronisiert ggf. Passwort-Hashes
- Besonderheit: Gelöschte Objekte werden von AD in einen „versteckten“ LDAP-Bereich verschoben



Ablauf der Synchronisation: UCS → AD

- Ein Listener-Plugin schreibt Änderungen im UCS Lda („altes“ und „neues“ LDAP-Objekt) in einzelne Dateien
- Der Connector-Daemon liest diese Dateien in regelmässigen Intervallen ein
 - Und mappt die Inhalte anhand der Konfigurations-Datei auf UDM-Module
 - Erkennt Anlegen/Ändern/Löschen anhand der Objekte in der Datei
 - Mappt das UDM-Objekt auf die AD LDAP-Attribute und ändert diese in AD
 - Die „Post Modify“ und ggf. „Post Create“ Funktionen werden aufgerufen, eine der Funktionen synchronisiert ggf. Passwort-Hashes
- War die Änderung erfolgreich wird die Datei gelöscht



Besonderheiten: Verschieben

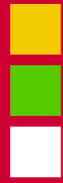
■ UCS → AD

- Das Listener-Plugin vermerkt die geänderte DN im Export-File
- Der Connector verschiebt in einen neuen Container gemäß Mapping-Definition
- ACHTUNG: AD ist restriktiver beim LDAP-Aufbau (z.B. kein Container in Container), daher können Verschiebe-Operationen zu rejects führen

■ AD → UCS

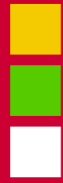
- Der Connector speichert den letzten bekannten DN eines AD-Objekts mit seiner eindeutigen ID (GUID) in der internal.cfg
- Ändert sich die DN weicht sie bei gleicher GUID von der zwischengespeicherten ab
- Sonderfall: gelöschte Objekte sind auch „verschoben“, werden aber am Attribut „isDeleted“ erkannt

- Container und Organisationseinheiten werden immer rekursiv verschoben



Besonderheiten: „Rejects“

- „Rejects“ sind fehlgeschlagene Änderungen
 - UCS → AD: eine Export-Datei wird nicht gelöscht und als Reject in der internal.cfg registriert
 - AD → UCS: die Create/Change-USN eines nicht synchronisierten Objektes wird in der „internal.cfg“ gespeichert
- Auflisten von Rejects per „univention-connector-list-rejected“
- Rejects werden im Rahmen von Synchronisations-Intervallen ohne sonstige Änderungen und beim (Neu-)Start des Connectors eingespielt
 - Ist die Listener-Exportdatei nicht mehr vorhanden wird der Reject UCS → AD verworfen (passiert nur durch manuelles Löschen)
 - Ist die ChangeID nicht mehr in AD vorhanden wird der Reject AD → UCS verworfen (das Objekt wurde erneut geändert und entweder synchronisiert oder ein zweites mal „rejected“)



Praxis: „Typische“ Rejects

❑ Allgemein

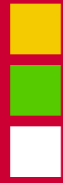
- ❑ Abweichende Attribut-Syntax (z.B. Telefonnummern), Lösung: Mapping-Funktionen in der Konfiguration
- ❑ Abweichende Single/Multivalue-Attribute, z.B. bei Mailadressen

❑ AD → UCS

- ❑ Systemaccounts in AD mit für UCS fehlenden Eigenschaften
- ❑ Primäre Gruppe eines AD-Benutzers ist in UCS nicht vorhanden (Meist: AD-Benutzer ist nicht in „Domain Users“)

❑ UCS → AD

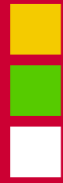
- ❑ Nicht unterstützte Container-Strukturen (CN unter CN)
- ❑ Gruppen in Gruppen (wird in AD erst mit Windows 2008 vergleichbar unterstützt)



Agenda

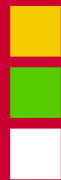
- 1) Einführung, Anwendungsszenarien
- 2) Technischer Aufbau
- 3) Praxis
 - a) **Basiseinrichtung**
 - b) Betrieb: Überwachung, Fehleranalyse
 - c) Sonderfälle: Neusynchronisation, erweiterte Konfiguration
- 4) Hinweise und Ausblick
- 5) Fragen/Diskussion





Praxis: Ersteinrichtung

- ACHTUNG: Teil der Aufgabe des AD-Connectors ist auch das Löschen von Objekten. Dies kann bei Fehlkonfigurationen zu Datenverlusten führen. Daher:
 - Vor der Produktivnahme immer ein Backup erstellen und den Disaster-Recovery-Fall prüfen
 - Das Setup in einer Testumgebung vorbereiten
- Ablauf der Ersteinrichtung:
 - Connector-Paket installieren (univention-ad-connector)
 - Passwort-Dienst und Zertifikate auf AD einrichten/exportieren
 - Basiseinrichtung per UMC durchführen
 - Start
 - Beobachten/Auswerten der Logdateien



Active Directory Connector-Konfiguration

Rechnername des Active Directory Servers

[BasisDN ermitteln](#)

BasisDN des Active Directorys

DN des Replikationsbenutzers

Verwendete Sprache für das Gruppen-Mapping

▼

Polling-Intervall (in Sekunden)

Debug-Level des Active Directory Connectors

▼

Version des Windows-Servers

▼

Passwort des Replikationsbenutzers

Synchronisationsmodus des Active Directory Connectors

▼

Wiederholungsintervall für abgelehnte Objekte

Debug-Ausgaben für Funktionen ausgeben

▼

Hinweis: alle Einstellungen werden erst nach dem (Neu)Start des Active Directory Connectors wirksam.


[Änderungen speichern](#)

[Schließen](#)

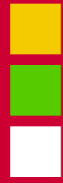
Active Directory Connector-Konfiguration

 [Active Directory-Zertifikat hochladen](#)

Nachdem die oben gemachten Einstellungen gespeichert wurden, muss das Active-Directory-Zertifikat für den Active Directory Connector hochgeladen werden.

 [Download des .msi-Pakets](#)

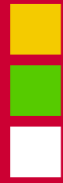
Abschließend müssen auf dem Active Directory Server einige Programme/Dateien für die korrekte



Agenda

- 1) Einführung, Anwendungsszenarien
- 2) Technischer Aufbau
- 3) Praxis
 - a) Basiseinrichtung
 - b) Betrieb: Überwachung, Fehleranalyse**
 - c) Sonderfälle: Neusynchronisation, erweiterte Konfiguration
- 4) Hinweise und Ausblick
- 5) Fragen/Diskussion



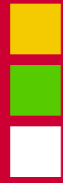


Statuskontrolle und Logdateien

- `/var/log/univention/connector-status.log`
 - Wird mit jedem Sync-Intervall neu geschrieben
 - Enthält die gerade durchgeführten und anstehenden Aktionen und die aktuell vorliegende Anzahl von Rejects

- `/var/log/univention/connector.log`
 - Typische Logdatei, Inhalt abhängig von den definierten Logleveln (UCR)
 - Enthält Statusmeldungen bei Rejects

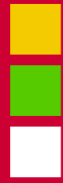
- `/var/log/univention/connector-tracebacks.log`
 - Erweitertes Debugging bei Rejects mit Code-Verweisen
 - Hinweise zur Auswertung finden sich u.a. im Univention Bugzilla



Agenda

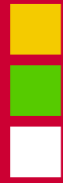
- 1) Einführung, Anwendungsszenarien
- 2) Technischer Aufbau
- 3) Praxis
 - a) Basiseinrichtung
 - b) Betrieb: Überwachung, Fehleranalyse
 - c) **Sonderfälle: Neusynchronisation, erweiterte Konfiguration**
- 4) Hinweise und Ausblick
- 5) Fragen/Diskussion





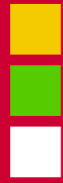
Erneute Synchronisation: AD → UCS

- Mapping und letzte ChangeID sind in der internal.cfg gespeichert
- Resync durch:
 - Beenden des Connectors
 - Löschen der Datei
 - Neustart des Connectors
- Ablauf Erstsynchronisation:
 - Connector generiert vollständige Liste der AD-Objekte, sortiert nach Create/Change-ID (in dieser Zeit keine Einträge in der connector-status.log)
 - Änderungen werden abgearbeitet



Erneute Synchronisation: UCS → AD

- Änderungen werden vom Listener-Plugin angestossen
- Ablauf des Resync:
 - Beenden des Connectors
 - Löschen der noch vorliegenden Dateien unter `/var/lib/univention-connector/ad` und dem tmp-Unterverzeichnis
 - Resync des Listener-Moduls:
„univention-directory-listener-ctrl resync ad-connector“
 - Der Resync läuft im Hintergrund ab!
 - Start des Connectors
- Der Connector arbeitet die Objekte in der vom Listener-Plugin vorgegebenen Reihenfolge ab, das Plugin wiederum in der vom LDAP-Server ausgegebenen Reihenfolge (entspricht i.d.R. der Reihenfolge des Anlegens im LDAP)



Erweiterte Konfiguration

■ „Post“-Funktionen

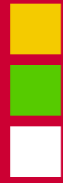
- Ausführen beliebiger Folgeaktionen nach Änderungen von Objekten
- Werden z.B. für die Passwort-Synchronisation genutzt

■ Mapping-Funktionen

- „Umschreiben“ von Attributwerten um Syntax-Anforderungen abzugleichen
- Wurden in Projekten z.B. für Telefonnummern genutzt

■ Multi-Instanz-Betrieb (seit UCS 2.3)

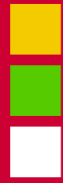
- Ermöglicht den Abgleich eines UCS-Servers mit mehrere AD-Domänen
- z.B. für den Betrieb als „Meta-Directory“ oder den Abgleich mit mehreren Domänen eines AD Forest
- Achtung: „Roundtrips“ und doppelte Benutzer/Gruppennamen bedenken



Agenda

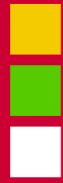
- 1) Einführung, Anwendungsszenarien
- 2) Technischer Aufbau
- 3) Praxis
 - a) Basiseinrichtung
 - b) Betrieb: Überwachung, Fehleranalyse
 - c) Sonderfälle: Neusynchronisation, erweiterte Konfiguration
- 4) Hinweise und Ausblick**
- 5) Fragen/Diskussion





Hinweise aus Projekten

- Testen, Testen, Testen....
- Backup erstellen und testen....
- Auch bei bidirektionaler Synchronisation einen führenden Verzeichnisdienst definieren, in dem vorrangig neue Objekte angelegt werden
- Umfangreiche Filterkonfigurationen bei bidirektionaler Synchronisation vermeiden
- Bei Windows 2000:
 - Längere Laufzeiten einplanen, Beispiel: bei ca. 2000 Usern kann die Erstsynchronisation >12h dauern, bisheriger „worst case“ waren >70h
 - Die Last der Erstsynchronisation kann die Administration des AD beeinträchtigen



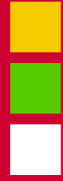
Ausblick

■ Weiterentwicklung AD-Connector

- Ausbau der Möglichkeiten in UMC
- Verbesserte Fehleranalyse
- Feedback und Patches von Partnern sind willkommen
- eDirectory und andere Verzeichnisdienste denkbar

■ Samba 4

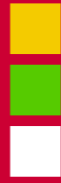
- Samba 4 unterstützt die AD-Replikation und Samba 4 kann OpenLDAP als Backend nutzen
→ es wäre als Alternative zum AD-Connector denkbar
- Aber: Samba 4 erstellt per Default Objekte wie in AD, Probleme mit vorhandenen Strukturen, Posix etc. sind zu erwarten
- Univention wird Samba 4 zum Release von UCS 3.0 vorbereiten



Agenda

- 1) Einführung, Anwendungsszenarien
- 2) Technischer Aufbau
- 3) Praxis
 - a) Basiseinrichtung
 - b) Betrieb: Überwachung, Fehleranalyse
 - c) Sonderfälle: Neusynchronisation, erweiterte Konfiguration
- 4) Hinweise und Ausblick
- 5) **Fragen/Diskussion**





Vielen Dank für Ihre Aufmerksamkeit!

 **univention**
linux for your business

Univention Partner Summit 2010

Ingo Steuer, steuer@univention.de