

# Univention Directory Reports

Thema:	Generierung von Reports aus dem Verzeichnisdienst, bspw. von Benutzerdaten und -berechtigungen.
Datum:	15. Dezember 2009
Seitenzahl:	<a href="#">10</a>
Versionsnummer:	4532
Autoren:	Univention GmbH   <a href="mailto:feedback@univention.de">feedback@univention.de</a>

## Inhaltsverzeichnis

<b>1</b>	<b>Übersicht</b>	<b>3</b>
<b>2</b>	<b>Registrierung von Report-Vorlagen</b>	<b>3</b>
<b>3</b>	<b>Erstellen von Reports</b>	<b>4</b>
3.1	Univention Directory Manager . . . . .	5
3.2	Kommandozeile . . . . .	6
<b>4</b>	<b>Erstellen von Vorlagen</b>	<b>6</b>
4.1	Definition . . . . .	7
4.2	Einfache Elemente . . . . .	7
4.3	Umgebungen . . . . .	8
4.4	Berichtsanfang und -ende . . . . .	9

## 1 Übersicht

Univention Directory Reports bietet die Möglichkeit vordefinierte Reports zu beliebigen im Verzeichnisdienst verwalteten Objekten zu erstellen.



Abbildung 1: Beispiel eines PDF-Reports

Die Struktur der Reports wird dabei durch Vorlagen definiert. Die dafür entwickelte Beschreibungssprache ermöglicht die Verwendung von Platzhaltern, die durch die Werte aus dem LDAP-Verzeichnis ersetzt werden. Über Univention Configuration Registry kann für jedes Univention Directory Manager Modul eine beliebige Anzahl von Reportvorlagen vorgegeben werden, die bei der Erstellung zur Auswahl stehen. So können beispielsweise für Benutzer wahlweise sehr detaillierte Reports oder nur einfache Adresslisten erstellt werden.

Für die Generierung der Reports kann entweder das Kommandozeilenprogramm `univention-directory-reports` oder das Web-Interface von Univention Directory Manager verwendet werden. Weitere Details dazu sind im Abschnitt [2](#) beschrieben.

## 2 Registrierung von Report-Vorlagen

Im Auslieferungszustand werden sechs Reportvorlagen von Univention Directory Reports bereitgestellt, die für Benutzer, Gruppen und Rechner verwendet werden können. Drei Vorlagen erzeugen PDF-Dokumente und drei CSV-Dateien, die als Import-Quelle für andere Programme verwendet werden können. Weitere Vorlagen können erstellt und registriert werden.

Schon vorhandene Reports können direkt mit den Voreinstellungen erstellt werden. Einige Voreinstellungen können mittels Univention Configuration Registry angepasst werden. Beispielsweise ist es möglich das Logo, das in der Kopfzeile jeder Seite eines PDF-Reports angezeigt wird zu ersetzen. Dafür kann der Wert der Univention Configuration Registry-Variable `directory/reports/logo` den Namen einer Bilddatei enthalten. Dabei können gängigen Bildformate wie JPEG, PNG oder GIF verwendet werden. Das Bild wird automatisch auf eine feste Breite von 5.0 cm angepasst.

Neben dem Logo kann auch der Inhalt der Reports angepasst werden indem neue Reportvorlagen definiert werden. Diese Vorlagen sind in einer Beschreibungssprache definiert, die in Abschnitt 4 detailliert beschrieben wird.

Eine neu erstellte Vorlage muss noch registriert werden, damit Univention Directory Reports diese Vorlage verwenden kann. Dafür müssen vorher folgende Parameter definiert werden:

**Univention Directory Manager Modul** Jede Vorlage ist für genau ein Univention Directory Manager Modul zu definieren. Beispiel: `users/user`

**Name** Der Name dient zur Identifikation und wird bei der Auswahl in Univention Directory Manager angezeigt

**Dateiname** Der absolute Pfad zu der Vorlage wird benötigt. Anhand der Endung des Dateinamen wird erkannt welches Ausgabeformat für den Bericht verwendet werden soll (PDF oder CSV). Für PDF-Reports werden Vorlagen in  $\LaTeX$  definiert und haben daher die Endung `“.tex“`

Sind alle diese Parameter bekannt, kann die Vorlage mit dem folgenden Kommando registriert werden:

```
univention-config-registry set directory/reports/templates/meinePDFVorlage1=  
  'users/user "Meine PDF Vorlage 1" /etc/vorlagen/vorlage1.tex'  
univention-config-registry set directory/reports/templates/meineCSVVorlage1=  
  'users/user "Meine CSV Vorlage 1" /etc/vorlagen/vorlage1.csv'
```

Danach können die Vorlage in Univention Directory Manager unter dem Namen `“Meine PDF Vorlage 1“` sowie `“Meine CSV Vorlage 1“` im Benutzer-Assistenten ausgewählt werden.

Dabei muss beachtet werden, dass es bisher keine weitere Univention Configuration Registry-Variable `directory/reports/templates/meinePDFVorlage1` existiert. `meinePDFVorlage1` kann durch eine beliebige Zeichenkette (ohne Leerzeichen) ersetzt werden, wichtig ist nur, dass die Präfix der Variable `directory/reports/templates/` ist.

### 3 Erstellen von Reports

Reports können über ein Kommandozeilenprogramm oder über das Web-Interface von Univention Directory Manager erstellt werden.

### 3.1 Univention Directory Manager

Die Funktion zur Erstellung von Reports ist in die Assistenten des Univention Directory Manager integriert. Im Untermenü **Suchen** gibt es eine Schaltfläche **[Report erstellen]**, die angezeigt wird, wenn für die Univention Directory Manager Module, die durch den Assistenten verwaltet werden, Reportvorlagen definiert sind (siehe Abbildung 2).

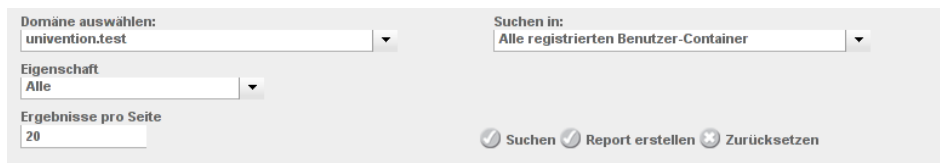


Abbildung 2: Erweiterter Suchdialog

Um Reports nur für eine bestimmte Auswahl von Objekten zu erstellen, kann mittels der gegebenen Suchkriterien eine Einschränkung vorgenommen werden.

Ist die gewünschte Auswahl von Objekten definiert, wird durch die Betätigung der Schaltfläche **[Report erstellen]** die Auswahl der zu verwendenden Vorlage angeboten (siehe Abbildung 3).

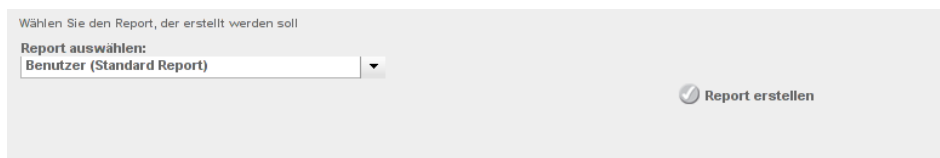


Abbildung 3: Auswahl der Vorlage

In der Auswahlliste stehen alle Vorlagen zur Verfügung, die für die Univention Directory Manager Module dieses Assistenten definiert wurden. Ist die gewünschte Vorlage ausgewählt, kann mit der Schaltfläche **[Report erstellen]** der Report erzeugt werden. Je nach Anzahl ausgewählter Objekte kann dieser Vorgang einige Zeit in Anspruch nehmen. Ist der Report erfolgreich erstellt worden, wird dieser als Link zum Herunterladen angezeigt (siehe Abbildung 4).

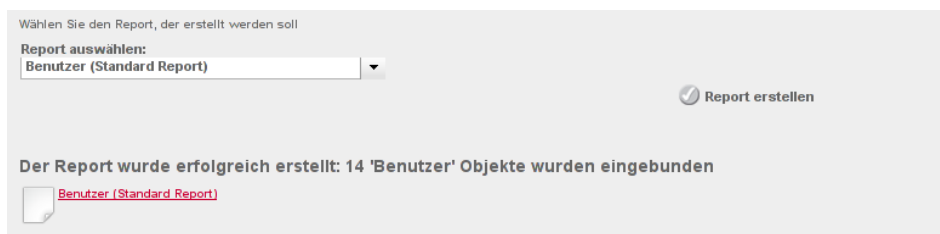


Abbildung 4: Ergebnisanzeige

Die über Univention Directory Manager erzeugten Reports werden für 12 Stunden aufbewahrt und danach durch einen Cron-Job entfernt. Die Einstellungen, wann dieser Cron-Job laufen soll und wie lange diese Reports aufbewahrt werden sollen, kann über zwei Univention Configuration Registry-Variablen definiert werden:

`directory/reports/cleanup/cron` definiert den Zeitpunkt zu dem der Cron-Job ausgeführt werden soll.

`directory/reports/cleanup/age` bestimmt das maximale Alter eines Report-Dokumentes in Sekunden bevor es gelöscht wird.

## 3.2 Kommandozeile

Reports können auch über die Kommandozeile mit dem Programm `univention-directory-reports` erstellt werden. Dies ist gerade für den Test neuer Reportvorlagen hilfreich, kann aber auch bei der Integration in eigene Programme genutzt werden.

Informationen zur Verwendung des Programm können über die Option `-help` abgefragt werden. Einfache und häufig genutzte Varianten werden hier kurz aufgelistet.

Mit dem folgenden Befehl können die verfügbaren Reportvorlagen zu einem Univention Directory Manager Modul aufgelistet werden:

```
univention-directory-reports -m users/user -l
```

Ist der gewünschte Report vorhanden, kann mit dem folgenden Kommando der Report für den Benutzer **user1** erstellt werden:

```
univention-directory-reports -m users/user -r "Standard Report" \  
uid=user1,cn=users,dc=univention,dc=qa
```

Es können beliebig viele DNs angegeben werden. Der Report wird dann alle angegebenen Objekte umfassen, die von dem Modul abgebildet werden können. Ist beispielsweise unter den DNs ein Rechner-Objekt, so kann dieses nicht mit in den Report für Benutzer aufgenommen werden.

## 4 Erstellen von Vorlagen

Reportvorlagen sind zunächst normale  $\text{\LaTeX}$ -Dokumente für die PDF-Reports bzw. eine einfache Textdatei für die CSV-Reports. Um die Attributwerte der Objekte einzufügen, werden an den entsprechenden Stellen Platzhalter eingebaut, die von Univention Directory Reports bei der Erstellung eines Reports ersetzt werden. Dabei gibt es nicht nur die Möglichkeit einfache Ersetzungen durchzuführen, sondern auch auf weiterführende Objekte zu verweisen, die über ihre DN mit dem aktuellen Objekt verknüpft sind.

Im Folgenden werden die vier Platzhalterelemente mit ihren möglichen Optionen vorgestellt und anhand von Beispielen die genaue Verwendung erläutert.

## 4.1 Definition

Die Syntax der Sprachelemente ist an SGML bzw. XML angelehnt. Sie besteht aus Elementen, umschlossen durch `<@ELEMENTNAME ... @>`, denen Optionen übergeben werden können. Anhand des Elementnames wird eine von vier Funktionen spezifiziert, die im folgenden aufgelistet sind. Dabei kann zwischen einfachen Elementen, die auf Attributen des aktuellen Objektes operieren und Umgebungen, die eine neue Menge von Objekten generieren, unterschieden werden. Diese Varianten können beliebig ineinander verschachtelt werden.

**attribute** soll aus dem aktuellen Objekt den Wert des angegebenen Attributs anzeigen.

**policy** kann zu einem definierten Richtlinien-Modul ausgeben, ob vom aktuellen Objekt ererbt wird oder direkt verbunden ist.

**resolve** Liest aus dem angegebenen Attribut den Wert, der einer LDAP-DN entsprechen muss und macht das Objekt zum aktuellen Objekt, d.h. alle Elemente innerhalb dieser Umgebung verwenden dieses Objekt bei der Auflösung. Das angegebene Attribut kann ein Multi-Value-Attribut sein.

**query** Ähnlich wie `resolve` kann diese Umgebung das aktuelle Objekt ersetzen. Mit dieser Umgebung ist es zusätzlich möglich eine Rekursion zu definieren. Dies kann beispielsweise für die Auflösung von verschachtelten Gruppen genutzt werden.

## 4.2 Einfache Elemente

Die beiden einfachen Elemente `attribute` und `policy` werten Attribute des aktuellen Objektes aus. Wird beispielsweise ein Benutzer-Report erstellt, kann mit dem folgenden Sprachelement der Nachname in das Dokument eingefügt werden:

```
<@attribute name="lastname"@>
```

Um eine Kombination aus Vor- und Nachname einzufügen kann folgendes Segment verwendet werden:

```
<@attribute name="lastname"@>, <@attribute name="firstname"@>
```

Das würde dann im Report folgendermaßen aussehen

```
Mustermann, Max
```

Da das Attribut `firstname` kein Pflichtfeld darstellt kann es leer sein. In diesem Fall wäre das Komma nach dem Nachnamen nicht erwünscht. Um dies zu unterdrücken gibt es die folgende Möglichkeit:

```
<@attribute name="lastname"@><@attribute name="firstname" prepend=", "@>
```

Die Option `prepend` wird bei gesetztem Attribut ausgewertet, wenn das Attribut gesetzt ist. In diesem Fall wird es dann vor den Wert des Attributes gehängt. Äquivalent dazu gibt es noch die Option `append`. Um bei einem nicht gesetztem oder leerem Attribut trotzdem etwas in den Report einzufügen, kann die Option `default` wie folgt verwendet werden:

```
<@attribute name="firstname" default="nicht angegeben"@>
```

Das einfache Element `policy` kann nur ein "Ja" oder "Nein" in den Report einfügen und damit die Anfrage nach einer ererbten oder direkt gesetzten Richtlinie beantworten.

```
<@policy module="policies/pwdhistory" inherited=""@>
```

In dem obigen Beispiel wird ein "Ja" in den Text eingefügt, wenn das aktuelle Benutzer-Objekt die Passwort-Richtlinie ererbt und anderenfalls ein "Nein". Die Anfrage kann auch andersherum definiert werden, indem statt der Option `inherited` die Option `direct` angegeben wird.

```
<@policy module="policies/pwdhistory" direct=""@>
```

### 4.3 Umgebungen

Die Umgebung `resolve` kann basierend auf Attributwerten, die eine DN enthalten, weitere Objekte auflösen. Innerhalb dieser Umgebung kann dann auf Attribute des aufgelösten Objekts zugegriffen werden.

Der folgende Ausschnitt ist die Zeile einer  $\text{\LaTeX}$ -Tabelle, die den Vorgesetzten eines Benutzers anzeigen soll. Da der Vorgesetzte eines Benutzers als LDAP-DN an diesem Objekt gespeichert wird, muss das Objekt erst geöffnet werden, bevor auf die Attribute Vor- und Nachname des Vorgesetzten zugegriffen werden kann.

```
Vorgesetzter &  
<@resolve module="users/user" dn-attribute="secretary" alternative=""@>  
  <@attribute name="lastname"@><@attribute name="firstname" prepend=", "@>  
</resolve@>
```

Dabei muss angegeben werden, welches Univention Directory Manager-Modul das Objekt mit der LDAP-DN verwalten kann (in diesem Fall `users/user`). Wenn das Attribut **`secretary`** gesetzt ist, gilt für die `attribute`-Elemente das Benutzer-Objekt des Vorgesetzten als aktuelles Objekt, so dass hier dessen Vor- und Nachname eingesetzt wird. Ist das Attribut nicht gesetzt, kann mit der Option `alternative` ein Text definiert werden, der stattdessen angezeigt werden soll.

Zwei weitere Optionen, die `resolve` unterstützt, sind `header` und `footer`. Der Wert dieser Optionen wird jeweils vor bzw. hinter der `resolve`-Umgebung eingesetzt, wenn Objekte gefunden wurden. In dem folgenden Beispiel wird eine komplette Tabellenzeile erstellt, d.h. es wird die erste Spalte mit dem Wert "Vorgesetzter &" und der Zeilenumbruch eingefügt.

```
Vorgesetzter &  
<@resolve module="users/user" dn-attribute="secretary"  
  header="Vorgesetzter &" footer=""@>  
  <@attribute name="lastname"@><@attribute name="firstname" prepend=", "@>  
</resolve@>
```

Die query funktioniert sehr ähnlich zu der Umgebung resolve, nur dass sie Objekte auch über mehrere Ebenen auflösen kann. Zusätzlich kann für dieses Kommando noch die Option separator angegeben werden. Ist dieses Attribut gesetzt wird zwischen die einzelnen Wert die angegebene Zeichenkette eingetragen. So ist es beispielsweise möglich eine Komma-separierte Liste zu erstellen.

Das folgende Beispiel demonstriert die rekursive Auflösung von Gruppen eines Benutzers, die in einer description-Umgebung aufgelistet werden:

```
<@query module="groups/group" start="groups" next="memberOf"
      header="\begin{description}" footer="\end{description}"
      alternative="Der Benutzer ist in keiner Gruppe enthalten"@>
  \item[<@attribute name="name"@>] (<@attribute name="description"@>)
</query@>
```

Die innerhalb dieser Umgebung aufzulösenden Objekte können durch einen Mustervergleich eingeschränkt werden. Mit der Option pattern kann dazu ein solcher angegeben werden. Der Wert der Option besteht dabei aus dem Namen eines Attributes und dem Muster, getrennt durch ein =-Zeichen. Nur wenn das Muster in dem Attribut existiert, wird das Objekt in die Liste der aktuellen Objekte mit aufgenommen.

Die zweite Option ist regex und funktioniert vergleichbar, verwendet aber reguläre Ausdrücke als Muster.

```
<@query module="groups/group" regex="name=~[(s\-)([a\-)(i\-)(h\-)]"...
<@query module="groups/group" pattern="name=a-*"...
```

## 4.4 Berichtsanzug und -ende

Normalerweise wird für die Berichterstellung immer der gleiche Anfang und das gleiche Ende verwendet. Diese werden aus einer Datei ausgelesen, die in der Konfiguration angegeben ist. Für die Berichte im CSV-Format wurde die Möglichkeit hinzugefügt diese beiden Teile mit in der Vorlage zu definieren. Dafür wurde die beiden neuen Umgebungen header und footer aufgenommen. Diese sind optional und können für jede Art von Vorlage verwendet werden.

```
<@header@>
Name      Nachname      Benutzername
</header@>
<@attribute name="name"@>      <@attribute name="lastname"@>
<@attribute name="username"@>
<@footer@>
-----
</footer@>
```

In dem Beispiel wird ein einfacher Bericht definiert, der mit der Zeile innerhalb der header-Umgebung beginnt und mit der Linie aus der footer-Umgebung endet. Der Inhalt des Berichts setzt sich aus jeweils einer Zeile pro Datensatz zusammen, die zwischen den beiden Umgebungen definiert ist.

Wird in einer Vorlage eine dieser Umgebungen angegeben, werden die globalen Einstellungen für Berichtsanfang bzw. -ende ignoriert.