Der Director

Einführung in den Director 4

Thomas Aeby, Phinex Informatik AG
28. Oktober 2004

phinex
informatik ag

Inhalt

1. Die Benutzeroberfläche

Login

Die Web-Oberfläche zum Director lässt sich über die URL

```
http://<server>/bigswaf/BigClerk/
```

ansprechen. Am Login-Prompt ist ein gültiges Login/Passwort, das für die Directorbenutzung freigegeben wurde („root") anzugeben.

Navigation und Bedienungselemente

Einmal eingestiegen findet man sich wieder in einer hierarchischen, an typischen Dateibrowsern orientierten Ansicht der LDAP-Datenbank und daran angelegter Navigation:

- Die linke Spalte zeigt die Organisationsstruktur („Ordner") an
- auf der rechten Seite wird der Inhalt des ausgewählten Ordners angezeigt (LDAP-Objekte)
- Per Click auf einen Ordner wird dieser ausgewählt und zum aktiven Ordner
- Per Click auf das „+"-Symbol linkerhand eines Ordners wird seine Unterordner-Struktur geöffnet
- Per Click auf ein Objekt im rechten Fensterteil wird ein Kontext-Menü geöffnet, in welchem eine darauf anzuwendende Funktion angewählt werden kann
- Mehrere Objekte lassen sich auswählen und gemeinsam manipulieren, indem die Ansicht per Menü **View • View Mode • List** auf Listenansicht gestellt wird und anschliessend mit den Checkboxen am rechten Rand selektiert wird. Im Menü Edit finden sich Operationen, die auf ganze Selektionen anwendbar sind.
- Neue Objekte lassen sich am Einfachsten durch Clicken auf „Add Object" und darauf folgende Auswahl des Objekttyps erstellen

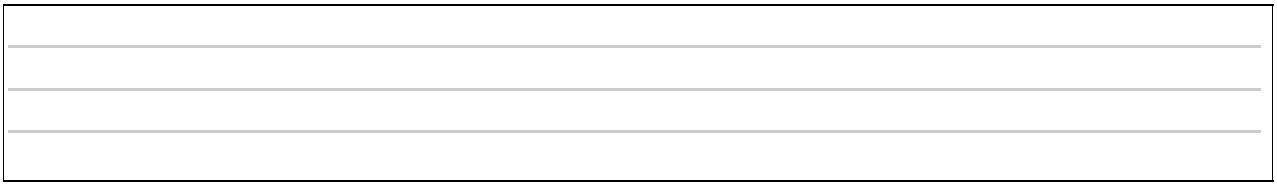
Aktionen und Events

Ein grundlegendes Konzept des Directors, das sich in der Oberfläche widerspiegelt, ist der sog. Event-Mechanismus. Jede Manipulation eines Objektes (Anlegen, Aendern, Löschen) löst ein *Event* aus, dem in der Director-Konfiguration eine/mehrere *Aktionen* zugeordnet sind, die die Manipulation in der Datenbank auf die Zielsysteme übertragen – z.B. beim Anlegen eines Benutzers im Director automatisch ein Home-Verzeichnis einrichten.

Events können auch manuell über das Kontextmenü eines Objektes ausgelöst werden, z.B. wird per **Event • Create** auf einem Benutzerobjekt das Home-Verzeichnis des Benutzers neu angelegt.

Datenbank-Struktur

Die Ordner der obersten Hierarchiestufe (Applications, Classes, DS-Rules, ...) haben eine spezielle Bedeutung. So können beispielsweise zwar prinzipiell unter „Hosts"



Benutzer angelegt werden, jedoch werden diese nicht überall korrekt ansprechbar.

Das Kontextmenü, das beim Anwählen von „Add Object" erscheint, zeigt in der oberen Sektion die Objekte an, die an der jeweiligen Stelle Sinn machen.

Unterhalb der obersten Hierarchiestufe darf frei eine Ordnerstruktur angelegt werden. Diese hat keinerlei Einfluss auf die Funktionen des Directors.

Benutzereinstellungen

Jeder Benutzer kann einige persönliche Einstellungen der Oberfläche vornehmen. Diese finden sich unter **Edit • Preferences**.

Die Spracheinstellung zeigt im Moment keine Wirkung – Uebersetzungen sind schlicht noch nicht vorhanden.

Nützlich ist die Einstellung unter **Director • Features**: Hier wählen Sie aus, welche speziellen Attribute/Objekte der Director Ihnen präsentieren soll. Wählen Sie beispielsweise „Windows" ab, so erscheinen beim Benutzereditieren die Windows-spezifischen Einstellungen nicht mehr.

Rolle / Berechtigungen in der Director-Oberfläche

Vordefinierte Rollen im Director sind „admin" und keine. Wer „admin" ist, hat automatisch Zugriff zu allen Objekten in der Datenbank und zu allen ihren Attributen. Wer keine Rolle zugewiesen hat, der hat keinerlei Zugriff zum Director. Die Rolle wird eingestellt mit:

- Unter „People" den passenden Benutzer finden
- Per Linksclick editieren
- Im Register Director als „BigClerk Access Role" „admin" oder nichts eintragen
- Speichern

Weitere Rollen lassen sich definieren – z.B. eine Rolle „Sekretariat", deren Benutzer Zugriff zu rein administrativen Attributen von Benutzern wie Telefonnummer oder Postadresse erhalten. Eine solche Definition sprengt allerdings den Rahmen dieser Einführung.

Die Kommandozeile

Detailliert wird in dieser Einführung nicht auf die Kommandozeile eingegangen, trotzdem werden ein paar Beispiele benutzt. Man sollte wissen:

- Alle Funktionen des Directors sind in einem Kommando zusammengefasst:
  `sfidirector <unterfunktion> <optionen>`
- Eine Liste der Unterfunktionen erhalten Sie durch das Kommando
  `sfidirector help`
- Das sfidirector-Kommando muss auf dem Adminserver ausgeführt werden – nur dort erhält der Director LDAP-Zugang

Logging

- Auf der Director-Oberfläche finden sich unter **Tools • Job Status Monitor** Informationen über die zuletzt ausgeführten Aktionen
- Director-Server und -Agents benutzen jedoch Syslog
- Ein `/etc/init.d/sfidirector status` auf dem Adminserver zeigt an, was der Director-Server gerade in diesem Augenblick tut

2. Benutzerverwaltung

Benutzer anlegen

- Im Browser das passende Unterverzeichnis unter dem Ast People anzeigen
- Per **Add Object • Person** eine Eingabemaske für einen neuen Benutzer öffnen
- In der ersten Maske Namen und Vornamen, Klasse („Users") und Passwort angeben, dann auf Weiter klicken
- In der folgenden Maske alle Eingabefelder auf Richtigkeit prüfen, gegebenenfalls korrigieren und dann per Klick auf „Save" den Benutzer anlegen

Im Hintergrund passiert nun folgendes:

- In der LDAP-Datenbank wird ein Objekt mit den eben erfassten Daten abgelegt
- Ein Homeverzeichnis für den Benutzer wird angelegt (falls nicht schon existent)
- Nach einer Wartefrist von ca. 1 Minute werden Einträge in sekundären Namensdiensten (z.B. smbpasswd) aufdatiert

Benutzer löschen

- Den zu löschenden Benutzer im Browser finden
- Kontextmenü mit Links-Klick öffnen, dort „Delete" wählen und bestätigen

Im Hintergrund passiert folgendes:

- Das Benutzerobjekt wird aus der LDAP-Datenbank gelöscht
- Das Homeverzeichnis des Benutzers wird ins Unterverzeichnis BACKUP des Homeverzeichnis-Volumes verschoben
- Nach einer Wartefrist von ca. 1 Minute wird die Aenderung an sekundäre Namensdienste weitergereicht

Nach Erstellen oder Löschen eines Objektes wird nur die aktuelle Browseransicht aufdatiert, sind mehrere Fenster offen, dann erscheinen/verschwinden die Objekte in der Ansicht erst beim weiteren Navigieren.

Was tun bei Problemen

Homeverzeichnis wird nicht erstellt

Haben Sie die richtige Klasse zugeordnet? Mit **Kontextmenü • Event • Create** wird der Director einen neuen Versuch starten, das Homeverzeichnis zu erstellen. Falls schon ein Verzeichnis existiert, aber den falschen Inhalt oder die falschen Berechtigungen hat, dann überschreibt **Kontextmenü • Event • Force (re-)initialize home** das Verzeichnis (löscht aber Benutzerdaten nicht).

Namensdienste (smbpasswd, u.ä.) werden nicht automatisch aufdatiert

Sie können manuell ein Update erzwingen per **Tools • Update Directory Services**.

Benutzergruppen anlegen/löschen

Dies funktioniert analog zum Anlegen / Löschen von Benutzern. Um Benutzer einer Gruppe zuzuordnen editieren Sie das Gruppenobjekt.

Wird ein Benutzer gelöscht, so werden seine Gruppenzugehörigkeiten nicht automatisch auch entfernt.

Benutzerklassen

Funktion

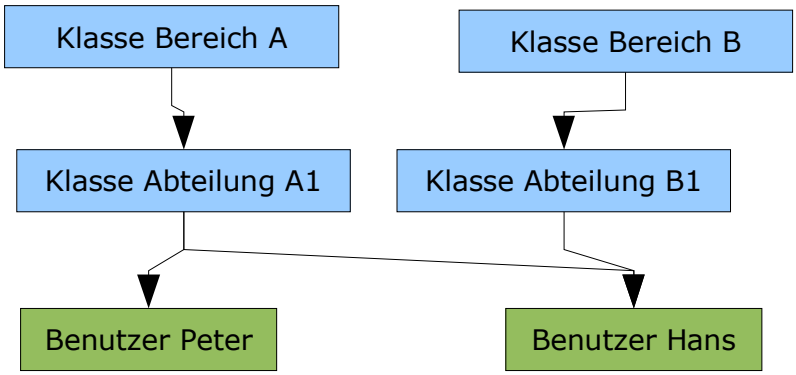
Jeder Benutzer kann einer oder mehreren Benutzerklassen zugeordnet sein. Die Klassenzugehörigkeit bewirkt folgendes:

- In der Benutzerklasse sind *Vorgaben* („Templates") definiert, die beim Erstellen eines neuen Benutzers dazu dienen, auf Grund von Vornamen/Nachnamen automatisch Vorschläge für die restlichen Felder zu machen
- Die Benutzerklasse enthält *Einstellungen*, die auf alle Benutzer in dieser Klasse vererbt werden

Ein wichtiger Unterschied zwischen Einstellungen und Vorgaben ist, dass Vorgaben nur zum Zeitpunkt der Neuerstellung eines Objektes wirksam sind und ihre Resultate im neu erstellten LDAP-Objekt mit abgelegt werden. Einstellungen hingegen sind im LDAP-Objekt nicht repräsentiert und werden bei jedem Zugriff durch den Director aufgelöst. Nachträgliche Aenderungen von Vorgaben haben auf bestehende Objekte keinen Einfluss, Aenderungen von Einstellungen hingegen schon. Für andere LDAP-Clients als den Director sind Einstelllungen unsichtbar.

Klassenhierarchie

Klassen können selbst wieder von anderen *Oberklassen* abgeleitet sein und erben sowohl Einstellungen wie Vorgaben von ihren Oberklassen. Einstellungen in einer tieferen Hierarchiestufe haben Vorrang vor Einstellungen in der Oberklasse, leere Einstellungen in einer Unterklasse bedeuten, dass die jeweilige Einstellung von der Oberklasse geerbt wird.



Ein Objekt kann auch von mehreren Klassen abgeleitet sein.

Einstellungen

In der Eingabemaske für Benutzerklassen sind alle Einträge ausser die in den Registern „UI" und „General" Einstellungen. Am sinnvollsten sind die Einstellungen in den Register „Unix" und „Home Dir" - sie besagen, wo sich das Homeverzeichnis eines Benutzers befindet, mit welchen Berechtigungen es eingerichtet wird und welches Skeleton-Verzeichnis zur Erstellung benutzt wird.

- Home File Server: Der Server, auf dem sich das Homeverzeichnis befindet
- Phys. Home Dir.: Der Homeverzeichnispfad auf dem Home File Server
- Template Server: Der Server, auf dem sich das Skeleton befindet
- Template Path: Das Verzeichnis auf dem Template Server, in dem sich das Skeleton befindet
- Access perms. und Dir Access perms: Berechtigungen, mit denen Dateien und Verzeichnisse im Homeverzeichnis angelegt werden – leer bedeutet: Die Berechtigungen aus dem Skeleton übernehmen

Vorgaben

Vorgaben finden sich im Register „UI". Jeder Eintrag erfolgt in der Form

```
attributname=Wert_oder_Ausdruck
```

Als Attributnamen werden diejenigen des Objekteditors (das Ding, das für die Anzeige von Eingabemasken verantwortlich ist) verwendet und diese entsprechen meist den LDAP-Attributen. Eine vollständige Liste der erkannten Attribute findet sich in `/usr/share/sfidirector/schema`, in der Datei Person und allen Dateien, die per „import"-Anweisung referenziert werden.

Eine Eingabehilfe für Vorgaben, die ein Nachschlagen der Attributnamen überflüssig macht, ist in Planung.

Ausdrücke in Einstellungen und Vorgaben

Neben einfachen Werten dürfen in Einstellungen und Vorgaben andere Attribute referenziert werden und ganze Berechnungen durchgeführt werden, z.B. wird eine

Vorgabe wie

```
uid=${sn}${givenname}${<:lowercase:asciify:substr.0.7:uniquelogin}
```

dafür sorgen, dass als Login-Name eine Kombination aus Nachnamen (sn) und Vornamen (givenname), klein geschrieben (lowercase), mit Sonderzeichen in US-ASCII-Zeichen umgewandelt (asciify), nur die ersten 7 Zeichen (substr.0.7) verwendet wird. Auch wird dafür gesorgt, dass der Login-Name durch optionales Hintanstellen einer Laufnummer eindeutig gehalten wird (uniquelogin).

In einem Ausdruck dürfen verwendet werden:

- Attributreferenzen: ${attributname}
- spezielle Attributreferenzen sind ${<} (die Zeichenkette, die links von dieser Referenz steht) und ${} (leer)
- Modifier: ${attributname:modifier1:modifier2:....}, darf man auch als Aufruf einer Funktion verstehen, die den Inhalt des Attributes nimmt, ihn verändert und an der Stelle einfügt, wo die Referenz steht
- Modifier mit Argumenten: ${attributname:modifier1.argument1.argument2}

Bekannte Modifier sind:

- lowercase/uppercase: Kleinschreibung/Grossschreibung
- finduid.lower.upper: Sucht im Zahlenbereich von lower bis upper nach einer noch nicht belegten UID
- findgid.lower.upper: analog zu finduid, aber für GIDs
- substr.start.end: Teilstring von Start bis Ende (beginnend bei 0)
- *.faktor: Interpretiert den Wert als Zahl und multipliziert sie mit faktor
- /.divisor, +.summand, -.subtrahend: Analog zur Multiplikation
- masterhost: Gibt den Namen des Director-Servers zurück
- firsthost.klassenname: Gibt den Namen des „ersten" Rechners in einer Rechnerklasse zurück
- uniquelogin: Ergänzt den Wert falls notwendig um eine Laufnummer um ein einen eindeutigen Login-Namen zu erhalten
- asciify: Schreibt einige Sonderzeichen wie Umlaute, Accents, etc. in eine reine ASCII-Repräsentation um
- shell.cmd.arg: Startet das Shell-Kommando cmd, schickt den Ausgangswert auf STDIN des Kommandos und akzeptiert Ausgaben an STDOUT als Resultat.

Beim Auflösen von Attributreferenzen in Einstellungen werden jeweils nur die Attribute der nächst-niedrigen Hierarchiestufe verwendet, d.h. z.B. kann eine Oberklasse einer Klasse nicht direkt ein Attribut eines Benutzerobjektes referenzieren. Dies darf man als künftig zu behebendes Fehlverhalten betrachten.

Auflösen von Einstellungen

Wie ein Objekt nach der Auflösung der durch Klassen vererbten Einstellungen für den Director aussieht, lässt sich nur auf der Kommandozeile herausfinden durch das Director-Kommando

```
sfidirector list –c People
```

3. Rechnerverwaltung

Die Rechnerverwaltung funktioniert analog zur Benutzerverwaltung, indem unter dem Ast „Hosts" passende Einträge gemacht werden. Ebenso wie Benutzer sind Rechner in einer Klassenhierarchie angelegt. Anders als bei Benutzerklassen sind bei Rechnerklassen allerdings nur Einstellungen verfügbar.

Fixe Adressen und DHCP

Ist im Rechnerobjekt sowohl die MAC-Adresse des Rechners erfasst, so wird in der DHCP-Konfiguration automatisch ein entsprechender Eintrag erstellt und der Rechner erhält eine fixe IP-Adresse per DHCP – sofern ein durch den Director verwalteter DHCP-Server eingerichtet ist (s. Kapitel Directory- und Namensdienste)

Rechner unter Director-Verwaltung stellen

Damit ein Rechner unter Director-Verwaltung steht muss

```
/root/.ssh/authorized_keys
```

auf dem Zielrechner den Public-Key des Root-Benutzers auf dem Adminserver enthalten. Ein

```
ssh zielrechner date
```

muss ohne Passwortabfrage vom Adminserver aus funktionieren.

Ob der Director einen Rechner verwalten kann, kann man auf der Kommandozeile mit

```
sfidirector –v event ping Hosts:rechnername
```

prüfen.

Die Methode, mit der der Director auf einen Rechner zuzugreifen versucht, wird durch das Register Invoker im Abschnitt Director im Rechnerobjekt oder seiner Oberklasse festgelegt. Tragen Sie „ssh" ein um dem Director mitzuteilen, er solle per ssh auf den Rechner zugreifen.

4. Anwendungsverteilung

Paketmechanismen

Anders als der Director 3 unterstützt der Director 4 direkt die Verteilung von Originalpaketen des Systems, also z.B. RPMs oder Debian-Paketen. Der Paketerstellungsmechanismus des Director 3 wurde klar von der eigentlichen Anwendungsverteilung separiert.

Repositories

Ein Paket – egal ob es sich um ein Originalpaket oder ein selbsterstelltes Director3-Paket handelt – muss in einem Paketrepository abgelegt werden, bevor es mit dem Director verwaltet werden kann. Ein Repository hat einerseits wie gewohnt einen LDAP-Eintrag unter dem Ast Applications sowie einen zugeordneten Server/Pfad, unter dem die Pakete physikalisch zu finden sind.

Die Defaultinstallation kennt die folgenden beiden Repositories:

- DebianRepository: Auf dem Adminserver unter `/export/admin/debian`, vorgesehen für Debian-Pakete
- Director Applications: Auf dem Adminserver unter `/export/admin/reldirs/` vorgesehen für Director-3-Pakete

Stellt man Pakete manuell in eines dieser Repositories ein, so muss dieses im Director mittels **Kontextmenü • Event • Scan Repository** neu indexiert werden. Als Ergebnis der Indexierung wird für jedes Paket ein eigener LDAP-Eintrag unter dem Applications-Ast erstellt.

Pakete zuweisen, verteilen, löschen

Pakete, die sich bereits in einem Repository befinden, können so verteilt werden:

- Im Register Applications des Rechners oder einer Klasse unter Subscribed den Applikationsnamen eintragen, Objekt speichern
- Klasse oder Rechner in der Oberfläche öffnen, dann im Tools-Menü „Install/Remove Applications" ausführen
- Alternativ auf der Kommandozeile:

  ```
  sfidirector –v app_dist –r –h rechnername
  sfidirector –v app_dist –r –c klassenname
  ```

  ausführen

Auf ähnliche Weise (Remove statt Subscribed) können Pakete auch wieder entfernt werden.

Pakete, die zwar im Register Applications eingetragen sind, aber in keinem Repository zu finden sind, werden ignoriert, also weder installiert noch gelöscht.

Paketgruppen

Mehrere Pakete lassen sich zu ganzen Paketgruppen zusammenfassen, die man dann an Stelle der Einzelapplikationen zuweisen lassen. Dazu geht man wie folgt vor:

- Im Ast „Applications" ein Objekt „Application Group" einrichten
- Dort die gewünschten Applikationen zuordnen
- Statt die Namen von Einzelpaketen den Namen der Applikationsgruppe einem Rechner oder einer Rechnerklasse zuweisen

Herausfinden, was die Anwendungsverteilung anrichten wird

Um unangenehme Ueberraschungen zu vermeiden, ist es bei grösseren Aenderungen oft hilfreich, vor dem eigentlichen Verteilen zu sehen, was der Director installieren/löschen möchte. Dies geht auf der Komandozeile mit app_dist:

```
sfidirector –v app_dist –n –r –h rechnername
```

Repositories und Paketarchitektur zuweisen

In einer Director-Umgebung werden typischerweise mehrere Paketrepositories existieren. Jedem Rechner müssen deshalb – am einfachsten über eine Rechnerklasse – die für ihn passenden Repositories zugewiesen werden. Das geschieht im Register Director, Attribut Repositories.

Ebenso können sich in einzelnen Repositories Pakete für verschiedene Rechnerarchitekturen befinden. Deshalb können unter dem Register Director auch Paketarchitekturen (z.B. i386) und Paketmechanismen (z.B. Debian, RPM) angegeben werden. Werden die Felder leer gelassen, so kommen alle Architekturen und Mechanismen in Frage.

Pakete in ein Repository stellen

Binärpakete

Binärpakete in einem Director-unterstützten Format (z.B. Debian, RPM, Sun-PKG) können direkt in das passende Verzeichnis kopiert und dann per „Scan Repository" neu indexiert werden.

APT-Pakete

Möchte man von den Möglichkeiten von Debians APT Gebrauch machen – besonders hübsch ist die automatische Auflösung von Abhängigkeiten – geht man wie folgt vor:

- Eine Datei mit der Endung .apt im Repository-Verzeichnis erstellen
- In dieser Datei eine Liste der Paketnamen eintragen, die man per Director verwalten möchte
- Das Repository per „Scan Repository" neu indexieren

Der Director wird jetzt die APT-Kommandos verwenden, um die Pakete zu indexieren, zu installieren und zu löschen. Dies verlangt allerdings, dass auf dem Director-Server

und den Zielrechnern identische APT-Konfigurationen vorhanden sind. Ausserdem sollte mit „dpkg-reconfigure debconf" die Priorität für Konfigurationsabfragen auf „critical" gestellt werden oder das Interface für Konfigurationsabfragen auf „Noninteractive".

***Vorsicht**: Das Löschen von Paketen per APT kann durch die Auflösung von Abhängigkeiten dazu führen, das überraschend viele Pakete deinstalliert werden. Schlimmstenfalls lässt sich eine komplette Linuxinstallation durch Deinstallation eines einzigen Basis-Paketes komplett entfernen.*

Eigene Pakete erstellen

Um ein eigenes Director-3-Paket zu erstellen geht man wie folgt vor:

- Unter dem Ast „Applications" ein Objekt „Packable Application" erstellen
- Dabei berücksichtigen:
  - Release-Dir `/export/admin/reldirs/rel<appname>`
  - Release-Host: `adminserver`
  - Application Repository: `Director Applications`
  - Installation Directory: `/export/opt/paketname` oder `/export/apl/paketname`
- Das Release-Verzeichnis sollte nun automatisch (leer) erstellt werden
- Im Release-Verzeichnis zu verteilende Datein, Installskripte, usw. erstellen
- In der Oberfläche **Kontextmenü • Event • Freeze** wählen.
- Das Paket ist nun zum Verteilen bereit

Director-3-Pakete können nur installiert, aber nicht gelöscht werden. Künftige Versionen des Directors werden eigene Pakete auch im nativen Format (z.B. Debian-Paket) des Systems erzeugen können, die dann natürlich auch deinstallierbar sind.

5. Konfigurationsverwaltung

Oft möchte ein Systemadministrator dafür sorgen, dass Dienste auf mehreren Rechnern gleichartig konfiguriert sind. Director 3 hat dafür noch die Applikationsverteilung verwendet. Director 4 führt dazu die wesentlich „leichtfüssigere" Konfigurationsverteilung ein. Sie verteilt lediglich einzelne oder wenige Konfigurationsdateien ohne einen Paketmechanismus einzubeziehen.

Konfigurationsdateien zuordnen

Wie gewohnt können Konfigurationen einzelnen Rechnern oder ganzen Rechnerklassen zugeordnet werden. Dazu machen Sie im Register System Config einen entsprechenden Eintrag in der Liste Configuration Template.

Jeder Eintrag in der Configuration-Template-Liste hat die Form:

```
paketname://servername/verzeichnispfad
```

Der Paketname definiert dabei, welche Konfigurationsdateien verteilt werden. Ein Paket kann mehrere Dateien enthalten. Eine Liste von bekannten Konfigurationsdateien und Paketnamen erhält man auf der Kommandozeile per

```
sfidirector listconfig
```

Servername und Verzeichnispfad sagen dem Director, aus welcher Quelle er Konfigurationsdateien verteilen soll, z.B.

```
//adminserver/export/configsrc/std
```

Unter diesem Verzeichnis ist eine „normale" Konfigurationsdateien-Struktur vorhanden. Konfigurationsdateien, die neu unter Verteilung zu stellen sind, müssen also dort eingepflegt werden.

Der Director verwendet keine speziellen Template-Anweisungen im den Quellen der Konfigurationsdateien. Um die aktive Konfiguration eines Rechners rechnerA auf einen rechnerB zu replizieren können Sie also in der Configuration-Template-Liste für rechnerB ganz einfach

```
paketname://rechnerA/
```

angeben.

Konfigurationsdateien verteilen

Verteilen lassen sich Konfigurationsdateien so:

- In der Oberfläche das Rechnerobjekt oder die Klasse öffnen
- Im Menü **Tools • Update System Configuration** ausführen

Falls eine Konfigurationsdatei durch die Konfigurationsverteilung verändert wird, wird automatisch der zugehörige Dienst neu gestartet.

Konfigurationseinstellungen

Oft sollen zwar Konfigurationsdateien gleichartig, aber nicht völlig identisch auf ver-

schiedenen Systemen existieren, z.B. soll ein Samba-Server zwar auf Maschine A und B gleich konfiguriert sein, aber unterschiedliche Workgroups bedienen.

In solchen fällen helfen Konfigurationseinstellungen dabei, die Konfigurationsdateien während ihrer Verteilung geringfügig ans Zielsystem anzupassen.

Einstellungen zuordnen

Um einem Rechner oder einer Rechnerklasse Konfigurationseinstellungen zuzuordnen, werden im Register System Config im Feld Package Configuration Einträge der Form

```
prefix.name=Wert
```

gemacht. Den jeweiligen Prefix kann man dem Output von

```
sfidirector listconfig
```

entnehmen. Erkannte Namen und die Bedeutung der Werte hängen stark von der jeweiligen Konfigurationsdatei ab, z.B. wird eine Samba-Workgroup in smb.conf mit

```
samba.global.workgroup=WORKGROUPNAME
```

gesetzt.

Einige bekannte Konfigurationseinstellungen sind

Name	***Bedeutung***
samba.global.*	Einstellung in der [global]-Section von smb.conf
ntp.keys.*	Schlüssel mit Nummer '*' in NTP-Keydatei setzen
ntp.tickers.server	Servereintrag in ntp step-tickers Datei
ntp.conf.peer	Liste der Peers in ntp.conf (Leerzeichen-getrennt)
ntp.conf.server	Liste der Server in ntp.conf
ntp.conf.*	beliebiger Eintrag in ntp.conf
squid.*	beliebige Einstellung in squid.conf, sofern nur einmal in der Konfigurationsdatei vorkommend (keine Einstellung von komplexen Optionen wie acls)
sendmail.cw.names	Liste der Namen in sendmail.cw-Datei
sendmail.masquerade.-domains	Liste von Namen in masquerade-domains Datei
sendmail.conf.smarthost	DS-Eintrag in sendmail.cf
sendmail.conf.hub	DH-Eintrag in sendmail.cf
sendmail.conf.local	Namen in C-Klasse (lokale Namen)
sendmail.access.*	Eintrag in sendmail access-Datei
uucp.call.*	User und Passwort für einen Eintrag in der UUCP-Calls-Datei
uucp.dialcode.*	Wähl-Prefix in UUCP-Dialcode-Datei
inittab.*	inittab-Eintrag mit Label '*'
resolvconf.nameserver	Liste der Nameserver in /etc/resolv.conf
resolvconf.domain	Domain in /etc/resolv.conf
logindefs.*	Logindefs-Eintrag

Die Konfigurationsverteilung lässt explizit auch eine Verteilung auf sich selbst zu, also z.B. eine Verteilung samba://localhost/.

6. Directory- und Namensdienste

Der Director legt seine Daten soweit möglich in Standardformaten im LDAP-Verzeichnis ab. Auf dieses können also andere Clients direkt zugreifen. Zusätzlich bietet der Director allerdings die Möglichkeit, diese Daten auch in andere Verzeichnis- oder Namensdienste zu propagieren. Dies dient dazu

- Dienste, die keine direkte LDAP-Anbindung bieten
- Dienste, die man aus Gründen der (Ausfall-)Sicherheit nicht direkt ans LDAP koppeln möchte
- Rechner, die aus Netzwerktopologischen Gründen (z.B. Firewall, keine permanente Netzwerkverbindung, etc.) nicht an den/die LDAP-Server koppelbar sind

mit Kopien der verwalteten Daten zu versorgen.

Der Begriff Verzeichnis- und Namensdienst ist dabei recht weit gefasst. Darunter fällt so ziemlich jeder Dienst, der rechner- oder benutzerspezifische, oder sonstwie in LDAP gepflegte Informationen benötigt, also z.B. DHCP, DNS, sendmail, etc.

Unterstützte Namensdienste

Unter dem Ast DS-Rules befinden sich Definitionen für die unterstützten Namensdienste. Darauf einzugehen, wie eigene Definitionen zu ergänzen sind übersteigt den Rahmen dieser Einführung. Einfache Beispiele findet man allerdings bei den Definitionen für DHCP, SambaPWD oder – etwas aufwändiger – Local Users Files.

Namensdienste zuordnen

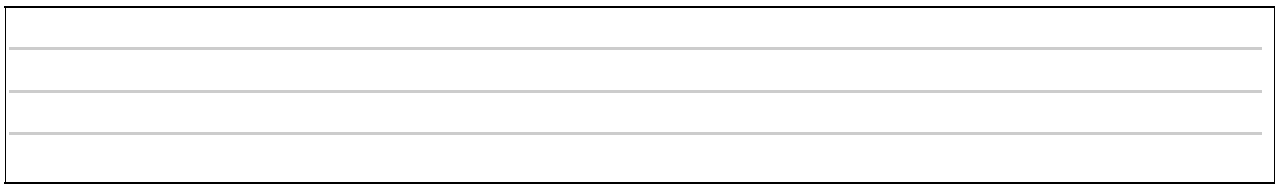
Damit ein Dienst auf einem spezifischen Rechner Daten aus dem LDAP erhält, muss eine Verbindung zwischen DS-Rule und Rechner (oder Rechnerklasse) hergestellt werden. Dazu geht man wie folgt vor:

- Rechner oder Rechnerklasse in der Oberfläche öffnen
- **Tools • Assigned Directory Services** wählen
- Es erscheint eine Tabelle mit den Zuordnungen des Rechners und sämtlicher Oberklassen zu den Namensdiensten
- Die Zuordnungen per Checkbox ergänzen
- Speichern

Dienst auf Zielrechner vorbereiten

Je nach Namensdienst ist es notwendig, die Konfiguration des Dienstes auf dem Zielrechner zu erstellen. Z.B. trägt die Definition für den DHCP-Dienst auf dem Zielrechner Rechnerinformation in der dhcpd.conf-Datei ein. Es ist also unumgänglich, dass diese erstens existiert und zweitens für den Director vorbereitet ist.

Im Falle von „DNS zones" und „DHCP" wird die Rechnerinformation direkt in die bestehende Konfigurationsdateien eingepflegt. Damit der Director weiss, an welcher Stelle er diese ablegen soll, ist ein entsprechender Abschnitt mit den Zeilen



```
##BEGIN director maintained section
##END director maintained section
```

zu begrenzen.

Verzeichnisdaten verteilen

Automatische Updates

Jedes Mal, wenn in LDAP über den Director eine Aenderung vorgenommen wird, prüft der Director, welche Namensdienste von der Aenderung betroffen sein könnten. Nach einer kurzen Wartezeit im Bereich von 1-2 Minuten – die dafür sorgt, dass mehrere kurz aufeinanderfolgende Aenderungen nicht zu unnötigen Updates führen – werden diese dann automatisch aufdatiert.

Der Director macht recht grosszügig automatische Updates. Um zu bestimmen, welche Dienste ein Update benötigen, berücksichtigt er lediglich die Objektklasse von modifizierten LDAP-Einträgen, ohne darauf zu achten, ob die Aenderung überhaupt einen konkreten Einfluss auf den jeweiligen Dienst hat.

Manuelle Updates

Sind die Daten eines Dienstes nicht aktuell, z.B. weil er eben erst eingerichtet wurde, so kann manuell ein Update gestartet werden.

Dazu öffnet man am einfachsten über **Tools • Update Directory Services** die entsprechende Maske, wählt die aufzudatierenden Dienste aus und klickt auf Ok.

Auf der Kommandozeile geht dies per

```
sfidirector –v event update DSDomains:<Name des Dienstes>
```

oder

```
sfidirectov –v event –a update DSDomains
```

Dienste-Restart

Nach erfolgtem Update wird der Director je nach Konfiguration (Feld Post Build Events im Register General einer DS-Rule) dafür sorgen, dass der Zieldienst das Update zur Kenntnis nimmt, z.B. indem der DHCP-Daemon neu gestartet wird.

Was wird aufdatiert

Was genau aufdatiert wird hängt von der jeweiligen Definition ab. Einige Beispiele:

- Samba Password File: Die Datei `/etc/samba/smbpasswd` wird neu geschrieben. Benutzer mit UIDs < 5000 oder >=50000 werden aus der Datei übernommen („lokale Benutzer"), Maschinenkonten werden ebenfalls aus der bestehenden Datei übernommen
- Local Users Files: `/etc/passwd`, `/etc/group` und `/etc/shadow` werden neu geschrieben. Benutzer mit UIDs < 5000 und mit UIDs >=50000 (lokale Benutzer) und Gruppen im selben GID-Raum bleiben erhalten. Einträge aus dem LDAP haben aber Vorrang, d.h. z.B. lässt sich das Root-Passwort so zentral verwalten.

- DHCP: `/etc/dhcp3/dhcpd.conf` wird überschrieben. Dabei werden alle Teile ausserhalb des für den Director markierten Abschnittes aus der lokalen Datei übernommen.