



Albrecht Weinert

Tutorial

Subversion —
mit Windows und Active Directory



Stand: 05.03.2010





Albrecht Weinert

Labor für Medien und verteilte Anwendungen (MEVA-Lab)

Fachbereich Informatik der Hochschule Bochum

Subversion — mit Windows und Active Directory

V01.01, 28.10.2008 : neu (ersetzt [15] als Nachfolger)
V01.02, 07.11.2008 : erste abgeschlossene Version
V01.04, 10.11.2008 : kleinere Ergänzungen (u.a.) zu keywords
V01.06, 12.11.2008 : ergänzt (6.4) "multi file / directory" - Operationen
V01.07, 17.11.2008 : kleine Ergänzung zu Auth-Problemen
V01.08, 21.10.2009 : Korrekturen, Ergänzungen, Aktualisierungen
V01.09, 04.03.2010 : kleine Korrekturen, Ergänzungen

Version: V1.08

Zuletzt geändert von A. Weinert am 05.03.2010

Copyright © 2008 Albrecht Weinert. All rights reserved. [a-weinert.de](http://www.a-weinert.de)

Hinweis: Wesentliche Listen, Tabellen, Listings, Bilder etc. sind gemeinsam durchnummeriert.

Hinweis: Die URL dieses Dokuments ist

<http://www.a-weinert.de/weinert/pub/svn-win-de.pdf> .

Die dort zu findende Version könnte neuer sein, als das Vorliegende (falls Sie Papier oder eine gecachte / kopyierte Version vor sich haben).

Inhalt

1. Zweck, Voraussetzungen	3
1.1 Ziel	3
1.2 Voraussetzungen	4
2. Client - Grundinstallation	5
2.1 SVN – Installation	5
2.2 Troubles – Behind a firewall ?	6
2.3 Eclipse - Client	8
3. Server - Grundinstallation	10
3.1 Voraussetzungen	10
3.2 SVN – Installation	10
3.3 Apache – Grundkonfiguration	12
4. HTTPS — SSL und Nutzerauthentifizierung.....	15
4.1 Vorbemerkungen.....	15
4.2 https und ssl, Tomcat vs. Apache, Hintergrundinformation.....	15
4.3 https und ssl nur für Apache mit openSSL.....	16
4.4 Erzwingen von https (und von Nutzer-Authentifizierung).....	19
5. Hinweise zur Active Directory Integration.....	21
6. Wenige Hinweise zum Betrieb	22
6.1 Repository anlegen	22
6.2 Repositories sichern, kopieren und löschen	22
6.3 keyword substitution und "deploy without keywords"	22
6.4 "multi file / directory" - Operationen	23
7. Resümee	24
7.1 Erreicht mit CollabNet Subversion	24
7.2 Warum SVN? (statt cvs[NT])	24
7.3 Summa summarum — doch eine gute Wahl.....	25
Anhang	26
A1 SVN – Server Konfigurationsdatei	26
A2 Autorisierung als getrennte Konfiguration	36
A3 XSL-Transformator für Verzeichnislisten	37
A4 XML - Listing und XSL-Anzeige Ergebnis	43
A5 Zur Ablehnung ohne Authentifizierungsabfrage	44
A6 Abkürzungen	45
A7 Literatur	48

1. Zweck, Voraussetzungen

Das im Folgenden beschriebene Vorgehen zeigt Ihnen, wie Sie SVN ^{*1)}, also den

- Versionskontroll-Dienst Subversion

als Dienst auf einem Windows-Server installieren.

Die reine SVN-Client-Installation wird als grundsätzliche Arbeits- und Testvoraussetzung vorangestellt.

Es geht hier um die Installation „ab Null“, aber eine Vertrautheit mit den Konzepten der Versionsverwaltung, von Windows-Domains und deren Administration wird vorausgesetzt.

1.1 Ziel

Das Vorliegende beschreibt die Installation und Inbetriebnahme eines

- SVN – Servers ($\geq 1.5.6$)

mit einem bzw. auf Basis eines

- Apache- ^{*2)} Webservers (≥ 2.2)
- in einem Windows-Umfeld.

Windows-Umfeld meint hier vor Allem eine größere

- Domain mit Active Directory (AD)

meist (und im Beispiel und Test) auf Basis

- Windows Server 2003 enterprise edition.

In einem solchen Domain-Umfeld ist eine Authentifizierung gegen AD-Gruppenkonten "spielentscheidend" ^{*3)}. In einer Firma oder Organisation mit i.A. Tausenden von Benutzerkonten und Dutzenden von Gruppen werden die die Firmen-, Organisations- und (Entwicklungs-) Projektstrukturen eh im AD nachgebildet und aktuell gepflegt. Rollen und Rechte auf Ressourcen werden darauf aufbauend i.A. durch Gruppenzugehörigkeiten gepflegt. In AD können Gruppen wiederum Gruppen als Mitglieder angehören. Dies ist ein wesentliches Mittel zur Beherrschung komplexerer Organisationsstrukturen.

1*): Das Einführen jeder der zahlreichen Abkürzungen beim ersten Auftreten im Text stört den Lesefluss dessen, der sie schon kennt — und nützt andernfalls bei erneutem Auftreten weiter unten nichts. Also bitte ggf. im Abkürzungsverzeichnis (Anhang A6, ab Seite 45) schauen.

2*): Dies mag überraschen, aber Apache bietet im Gegensatz zu SVNserve die Möglichkeit einer AD-Authentifizierung. Und mehr Basis-Server-Auswahl (entweder Apache oder SVNserver) gibt es de facto nicht

3*): Windows-Administratoren hassen Anwendungen, die eine eigene Nutzer-, Gruppen oder gar Passwortverwaltung einführen — und aus gutem Grund.

21.10.2009	14:04	19.264.320	
			CollabNetSubversion-server-1.6.5-6.win32.exe
17.10.2008	15:41	39.674	mod_auth_sspi-1.0.4-2.2.2.zip
21.10.2009	15:18	10.585.405	
			CollabNetSubversion-client-1.6.5-6.win32.exe
26.08.2009	14:55	77.117.208	jdk-6u16-windows-i586.exe
04.09.2009	11:29	11.919.033	erg.zip
29.08.2009	08:55	198.520.136	eclipse-jee-galileo-win32.zip
21.10.2009	16:31	21.779.237	
			CollabNet_Desktop_e3.4-1.7.0.zip

Liste 1: Bereitzustellende Installationsdateien; Stand Oktober 2009
(<http://www.collab.net/>, <http://www.a-weinert.de/frame4j/downloads/> etc.).

1.2 Voraussetzungen

- a.) Sie haben die Installationsdateien aus der Liste 1 oder neuere bzw. Sie haben die betreffenden Anwendungen und Ergänzungen bereits installiert.
- b.) Sie haben einen geeigneten Server für die Installation von Apache und Subversion, vorzugsweise mit Windows Server 2003 (W2K3 oder W2K8) und RAID.
Der Server wird in den folgenden Beispielen PD331S genannt; er gehört der AD-Domain (als member computer) an. Er hat jederzeit (für Authentifizierungen) Zugriff auf mindestens einem Domain-Controller (besser auf 2). Er ist aber selbst kein DC.
Für die Client-Installation steht Ihnen eine weitere Windows-Workstation zur Verfügung. W2K3 und Domain-Mitgliedschaft erleichtert das professionelle Arbeiten. Ein XP-PC tut es aber auch.
- c.) Der Server (PD331S) ist im Netzwerk für alle beabsichtigten Client-Rechner (der Abteilung, der Firma, weltweit) vor Allem mit http(s) zugänglich. Er hat genügend Plattenplatz für Alles, was Sie da an versionsverwalteten Projekten anbieten wollen. Der Platzbedarf übersteigt deutlich die Gesamtgröße aller eingebrachten Dateien.
- d.) Sie haben administrativen Zugriff auf den Server. Ein remote-Zugriff genügt vollauf (Einstellungen -> Systemsteuerung -> System -> remote).
- e.) Auf dem Client sollte ein zeitgemäßes Eclipse (>= 3.4) auf Basis eines entsprechenden JDK (>= 1.6.0_18) mit allen sinnvollen Ergänzungen vorhanden sein.
- f.) Auf dem Client benötigen Sie einen Browser, wie Firefox (>= 3.5.8).

Anmerkung zu a) und e): Das Vorliegende geht, ohne es jedes mal zu erwähnen, von JDK1.6 und von Eclipse 3.5 (Galileo) etc. aus.

Nehmen Sie ohne Not nichts Älteres mehr für den Neueinstieg oder eine Renovierung.

2. Client - Grundinstallation

Eine JDK- (als Basis), Eclipse- und Browser-Grundinstallation wird für das Folgende auf allen Client-Workstations vorausgesetzt, auf denen Sie einen SVN-Client haben wollen. Ohne dies fehlt einfach zu viel an sinnvollem Benutzer-Komfort.

2.1 SVN – Installation

Die einfachere Client-Installation ist eine "Vorübung" und eine Testvoraussetzung für die Server-Installation.

Hinweis: Die Server-Installation umfasst die Client-Installation. Wenn Sie also auf Ihrem Büro- / bzw. Labor-Rechner einen lokalen (Spiel-) SVN-Server haben wollen, installieren Sie nur diesen — gleich im folgenden Schritt 1.

1. Schritt: Lassen Sie (mit administrativen Rechten)

```
CollabNetSubversion-client-1.6.5-6.win32.exe
```

laufen. Ändern Sie den Installationspfad in

```
C:\programme\Subversion.
```

Die Installation läuft durch.

Sie haben eine saubere Suchpfad-Strategie, die diesen (PATH) auf folgendes beschränkt:

```
PATH=C:\bat;C:\Programme\util;C:\programme\jdk\bin;  
C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem
```

Dies lassen Sie sich auch nicht mit jeder Neuinstallation verderben.

2. Schritt: Gehen Sie in die Systemverwaltung und entfernen Sie das ergänzte

```
;C:\Programme\Subversion
```

(und was Sie sonst noch Unnötiges finden) vom Suchpfad.

Hinweis: Bei neueren Serverinstallation geht das evtl. so nicht, da der Apache dann mit "mod_dav_svn.so not found" nicht startet. Tatsächlich findet er aber seine .dll nicht.

3. Schritt: Fügen Sie eine geeignete Script-Datei svn.bat (Listing 2) in C:\bat ein.

4. Schritt: Für ggf. spätere Scripte die svn.exe benötigen, ergänzen Sie vorausschauend diese Systemvariable

```
SVNEXE=c:\Programme\Subversion\svn.exe
```

```
@Echo.  
@Echo off  
@Echo svn.bat V0.01, 16.10.2008, A. Weinert  
@Echo.  
@echo svn %*  
@c:\Programme\Subversion\svn.exe %*  
@Echo.
```

Listing 2: Aufruf-Script für SVN

Nun sollte bereits das Folgende testweise funktionieren. Gehen Sie dazu in ein leeres Spielverzeichnis, das sich leicht aufräumen bzw. beseitigen lässt:

```
MD D:\temp\playground
CD /D D:\temp\playground
svn checkout  $\uparrow$ 
    http://svn.apache.org/repos/asf/tomcat/tc6.0.x/trunk/  $\uparrow$ 
    tomcat6trunk

svn checkout  $\uparrow$ 
    http://ai2t.de/svn/frame4j/src/trunk frame4j_trunk
```

(siehe auch unten zu Eclipse)

```
svn checkout http://svn.collab.net/repos/svn/trunk subversion
```

Falls Sie bereits (siehe unten) den Server PD331S mit einem Test-Repository installiert haben geht auch dieses:

```
svn checkout http://pd331s/svn/projSandbox projSandbox
```

Geht davon etwas (sonst funktionierendes) nicht, dann

2.2 Troubles – Behind a firewall ?

Wenn Sie nicht nach außen kommen, ändern Sie in der Datei **)

C:\Dokumente und Einstellungen\sinngemäß folgendes:

[global]

```
http-proxy-exceptions = *.fh-bochum.de, PD331S, pd331s
http-proxy-host = 193.175.85.214
http-proxy-port = 8080
```

Die Angaben müssen natürlich zu Ihren Proxy-Einstellungen und -Ausnahmen (sprich "diesseits" liegenden SVN-Servern) passen. Nun sollten die externen Zugriffe gehen.

Hinweis **): An der Stelle <name> kann Ihr Kontenname, aber auch "all users" stehen.
Dateien von "all users" werden von svn.exe beim ersten mal zum Profil des angemeldeten Nutzers kopiert.

Falls Sie bereits, wie weiter unten beschrieben, den Server haben und die Zugriffe à la

```
svn checkout http://pd331s/svn/projSandbox projSandbox
```

trotzdem nicht gehen, probieren Sie die betreffende URL mit Mozilla und dem Eclipse-Client (siehe unten). Falls letztere gehen und nur der Kommandozeilen-Client (svn.exe) und auch Eclipse (s.u.) nicht, liegt häufig einer der folgenden beiden Fälle vor:

Fall 1: Die Fehlermeldung

Server sent unexpected return value (403 Forbidden) in response to OPTIONS erfolgt, ohne dass dies im Server (Apache-Logs) als Zugriff bzw. als Fehler vermerkt wird.

Es gibt die gleiche Fehlermeldung auch, wenn Sie statt korrekt "PD331S" (in unserem Beispiel) einen nicht vorhandenen Server (//pusemukel/) eintippen.

Dann liegt dies im Allgemeinen (immer noch) an einer fehlerhaften Proxy-Konfiguration. Schauen Sie sich in diesen Fällen nochmal die oben genannte Datei `servers` in

```
C:\ . . . . . \Anwendungsdaten\Subversion\
```

an und experimentieren Sie ggf. mit dieser.

Die (Proxy-) Verhalten von Mozilla, `svn.exe` und Eclipse unterscheiden sich hier.

Fall2 : Die Fehlermeldung à la

```
"unexpected return value (401 Authorization Required) in response to OPTIONS"
```

erscheint mit `svn.exe` (und mit Eclipse), ohne dass Sie nach Name und Passwort gefragt wurden, während der Zugriff mit Mozilla nach einer entsprechenden Frage und Antwort geht.

Wenn der betreffende Rechner kein Domain-Mitglied ist, könnte die im Anhang A5, Seite 44, vollständig zitierte und kommentierte Erklärung zutreffen:

```
"The reason for this is the automatic authentication with SSPI which is activated in version 1.4.x. That means TortoiseSVN now tries to authenticate automatically with the credentials of the user logged on to the Windows domain controller."
```

Solange dieses hier beschriebene Verhalten zutrifft, hat man an einem Domain-Rechner (member computer) und (!) eingeloggt als der Nutzer, der SVN-Zugriff hat, einen Komfortgewinn. Und dies mag das Motiv für die zitierte 1.4.x-Änderung sein.

Unter folgenden Bedingungen aber ist das "tödlich":

1. Sie sitzen an einem Nicht-Domain-Rechner mit lokaler Anmeldung
2. Ihr Domain-Anmeldekonto ist ein anderes als das Domain-Konto, das Ihnen SVN-Zugriff gibt. (Das mag dem einen oder Anderen seltsam erscheinen, ist aber eine vielfach sinnvolle, und dann auch häufige Situation.)

Der eigentliche Fehler liegt in der Vorgehensweise "now tries to authenticate automatically with the credentials of the user logged on to the Windows domain controller". Dies ist unter auch nur einer der eben genannten Bedingungen immer (immer!) falsch.

Man kann dieses Bug-Verhalten abstellen, indem man in `httpd.conf` an entsprechender Stelle dies hinzufügt:

```
SSPIOfferBasic On # Required for TortoiseSVN, svn.exe ...  
SSLRequireSSL
```

Hinweis: Dies ist ein Vorgriff auf die Serverinstallation weiter unten.
Die Beispieldatei `httpd.conf` im Anhang hat diese Einstellungen.

Nun fragen `svn.exe` (und Eclipse) beim ersten Zugriff immer nach Name und Passwort, wofür man bei gerade bei Eclipse nach Fehlerfensterchen nicht gleich aufgeben, sondern den Versuch auch mal beharrlich wiederholen muss. Man verliert als an einem member-computer schon "richtig" eingeloggter Nutzer etwas an Komfort, weil man u.U. seine Angaben nochmal machen muss.

Eine Einstellung, die die Nutzung der Windows-Anmeldung versucht und in allen anderen Fällen trotzdem zuverlässig fragt, wurde (noch ?) nicht gefunden.

Bei der Gelegenheit: Außer in wirklich abgeschlossenen reinen Domain-Umgebungen ist es eine recht gute Idee SSL zu verwenden, nicht nur weil dann Windows- / AD-Konten-

namen und -Passworte via basic authentication über die Leitung gehen.
Die oben angegebene zweite Zeile verlangt das.

2.3 Eclipse - Client

Es wird ein zeitgemäßes Eclipse (>= Gallileo) installiert und funktionsfähig vorausgesetzt.
Packen Sie die Datei

```
16.10.2008 13:42 21.587.020 CollabNet_Desktop_e3.4-1.7.0.zip
```

in ein temporäres Verzeichnis aus:

```
MD D:\temp\CollabEcli
CD /D D:\temp\CollabEcli
jar xfv W:o\immer\es\ist\CollabNet_Desktop_e3.4-1.6.0.zip
```

Starten Sie Eclipse. Schließen Sie vielleicht besser alle Projekte und machen eine Sicherheitskopie Ihrer Eclipse-Installation (C:\Programme\Eclipse\...).

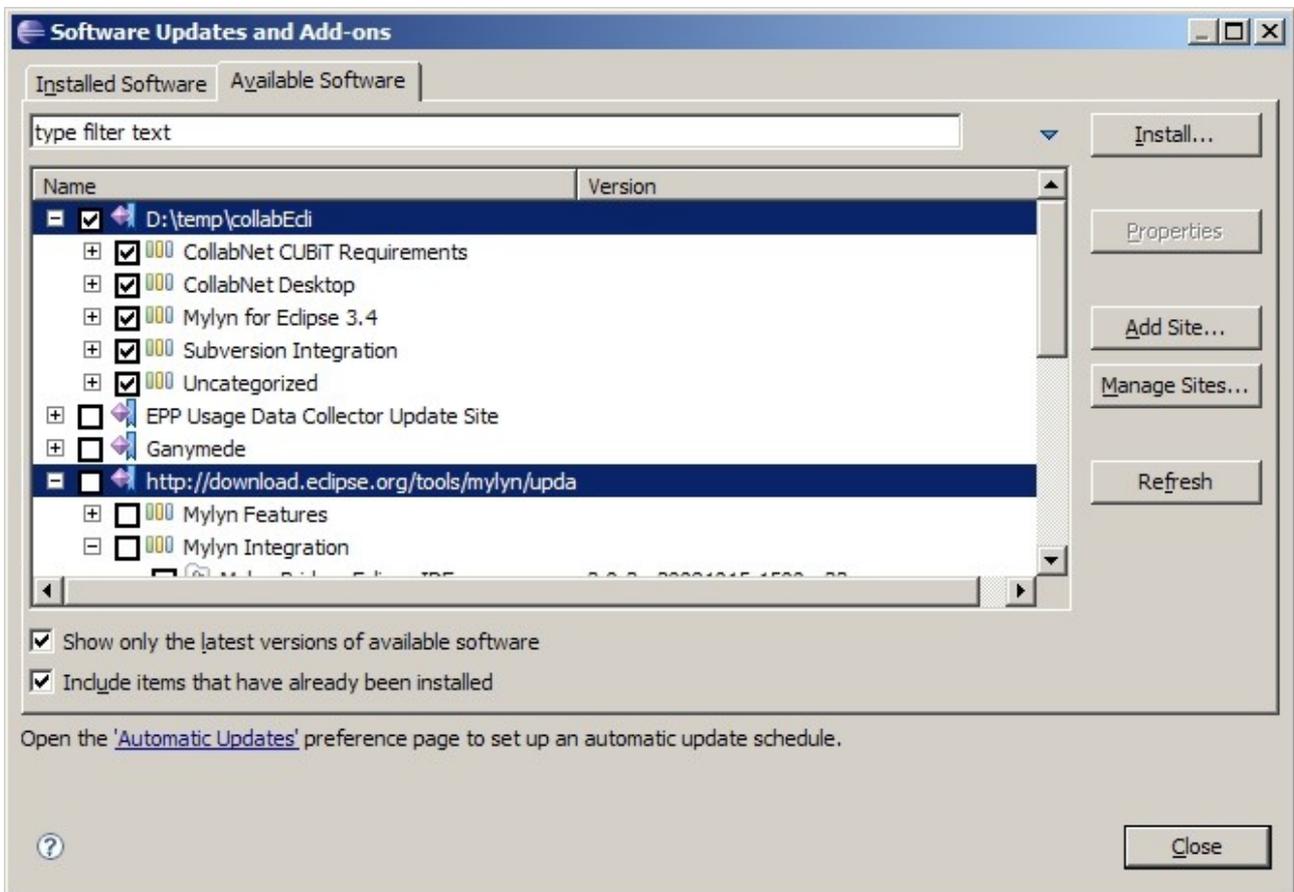


Bild 3: Lokale "site" für CollabNet-Desktop-Installation

Gehen Sie in Help->Software-Updates und dort in "add site" (Bild 3) Ihr Entpackverzeichnis an. Drücken Sie auf "install". Akzeptieren Sie die angegebenen Modifikationen (Bild 4) gegebenenfalls und drücken Sie "next".

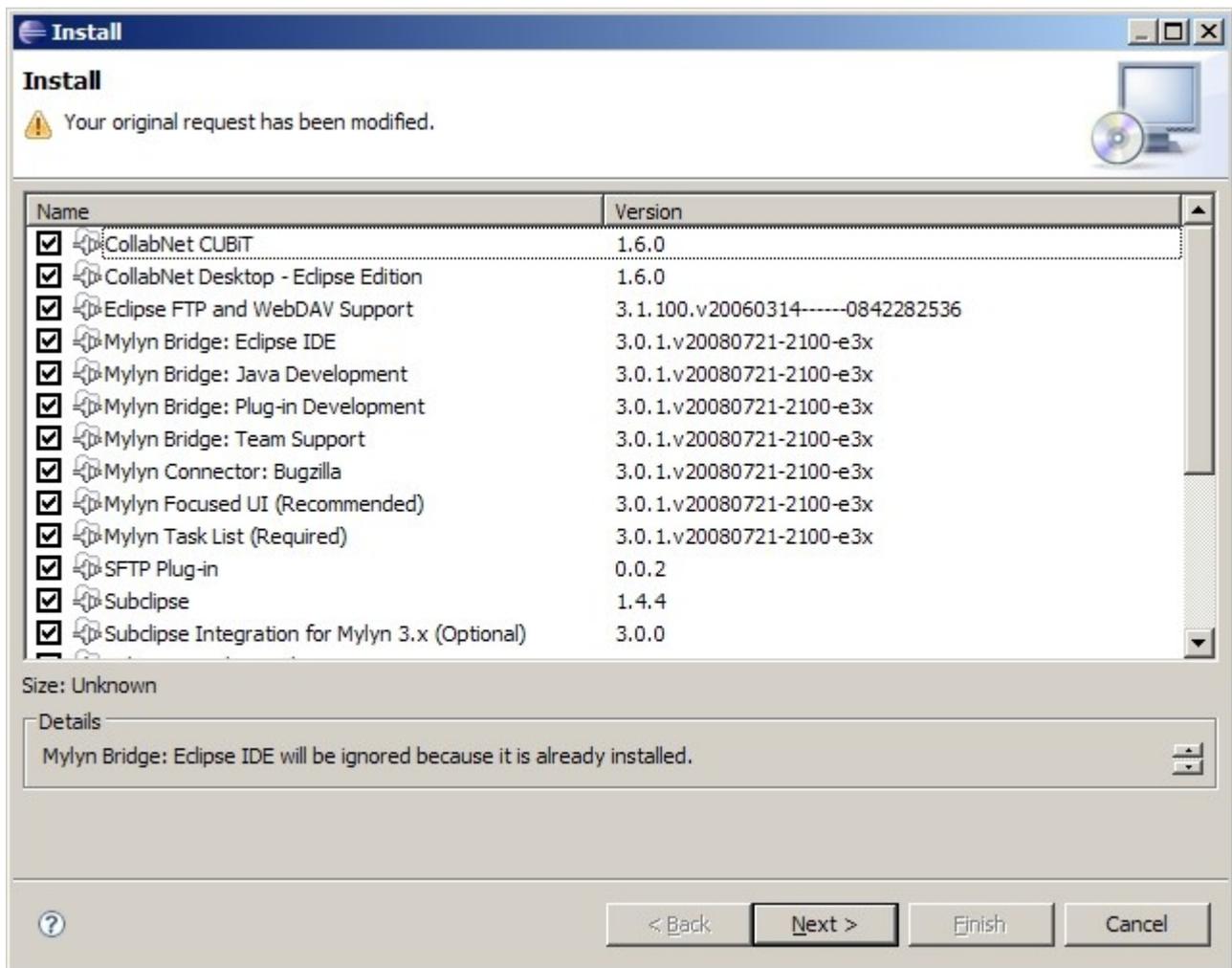


Bild 4: CollabNet-Desktop-Installation, Modifikationen

Akzeptieren Sie die Lizenzbedingungen und im Verlauf der so gestarteten Installation auch das absolut komische, leere Zertifikat. Wer sich nur wegen Apache (siehe weiter unten) je mit openssl herumgequält hat (anstatt die bisherigen Zertifikate etc. einfach weiter verwenden zu können), hat Verständnis für so was. Um das (leere) Zertifikat zu akzeptieren müssen Sie es ausdrücklich selektieren. Ohne dies "Zertifikat" bricht die Installation ab.

Nach der Installation starten Sie Eclipse neu.

Zum Test erzeugen Sie ein oder mehrere "checkout from SVN"-Projekte. Verwenden Sie dazu die (erprobten) URLs aus dem letzten Kapitel.

Probleme: Da beide,

CollabNet_Desktop_e3.4-1.7.0.zip und
CollabNetSubversion-client-1.6.5-6.win32.exe

offenbar dieselbe Software zum Zugriff auf den svn-Server verwenden haben, bekommen Eclipse und die Kommandozeile (svn.exe) im Allgemeinen jeweils die gleichen Probleme. Anders ausgedrückt:

Was mit svn.exe nicht funktioniert, wird i.A. auch dem Eclipse-Client nicht gelingen.

Ach ja: Und spätestens wenn all das funktioniert, installieren sie noch Tortoise, den Komfortbringer für Windows- (Explorer-) Nutzer.

3. Server - Grundinstallation

3.1 Voraussetzungen

Für ein gegen AD authentifizierenden SVN-Server haben Sie einen geeigneten Application-Server, hier beispielhaft PD331S genannt, vorzugsweise mit Windows Server 2003 (oder inzwischen auch 8) enterprise edition, der Mitglied der betreffenden Domain ist.

Auf diesem Application-Server sollten — außer evtl. Zusatzaufgaben für den mit installierten Apache — keine weiteren (80/443) Webdienste laufen müssen (J2EE, Tomcat, IIS, etc.). Dann kann für den (Web-Dav-) Zugriff auf SVN via Apache sinnvollerweise Port 80 / 443 genommen werden. Und etwas statischen Webinhalt etc. pp. erledigt der mit SVN installierte Apache ggf. gleich mit.

Ferner haben Sie die Installationsdateien aus der Liste 1 (Seite 4), und die übrigen dort genannten Voraussetzungen sind erfüllt.

Hinweis: Durch das immer unsinnigere hinein Spielen von teilweise unbeeinflussbaren Internet-Explorer-Einstellungen in alles Mögliche trieb Microsoft schon Viele zur Verzweiflung — und manche treue Seele zu alternativen Systemen.
Wenn also die unten genannte ".exe" nicht laufen darf, schauen Sie hier in den Blog: <http://blog.a-weinert.de/betriebssysteme/win-verweigert-exe/> .

3.2 SVN – Installation

Zum Installieren des Subversion-Servers lassen Sie zunächst mit administrativen Rechten

```
CollabNetSubversion-server-1.6.5-6.win32.exe
```

auf dem Ziel-Server, hier PD331S genannt, laufen. Wählen Sie

```
C:\Programme\Subversion\
```

als Ziel und unbedingt Apache als unterlagerten Server. Nur mit Apache geht die AD-Authentifizierung [sic!].

Entpacken Sie in ein temporäres Verzeichnis

```
mod_auth_sspi-1.0.4-2.2.2.zip
```

Kopieren Sie von dort nur die Datei

```
21.10.2008 16:13 28.672 mod_auth_sspi.so
```

nach

```
C:\Programme\Subversion\httpd\modules
```

Bevor Sie etwas starten, müssen Sie sich um folgende Dateien kümmern:

```
C:\Programme\Subversion\httpd\conf\httpd.conf
```

```
C:\Programme\Subversion\httpd\conf\extra\httpd-ssl.conf
```

```
.. .C:\Dokumente und .....\Anwendungsdaten\Subversion\servers
```

sowie ggf. auch um

```
C:\Programme\Subversion\httpd\conf\authz.conf
```

```
.. D:\www\serv-intra\conf\svnindex.css
```

```
.. D:\www\serv-intra\conf\svnindex.xsl
```

```
.. D:\www\serv-intra\index.html
```

Die Datei **httpd.conf** ist die Apache-Grundkonfigurationsdatei des darauf basierenden SVN-Servers. In ihr binden Sie auch die Grund-Authentifizierung der Nutzer an das AD. **httpd-ssl.conf** regelt den SSL- / https-Zugriff.

Mit der Datei **servers** regeln Sie den lokalen SVN-Zugriff von dem SVN-Server (PD331S) aus. Im Allgemeinen werden Sie diesen client-Zugriff von dort selten bis nie brauchen. Aber gehen sollte er im Zweifelsfalle. Siehe dazu das letzte Kapitel zur client-Installation.

Mit **authz.conf** können Sie die Zugriffsmöglichkeiten der Nutzer feingranular regeln.

Mit **svnindex.css** und **svnindex.xsl** sollten Sie die Browser-Darstellung Ihrer Repository-Listings ansprechend gestalten.

Und mit **index.html** sollten Sie eine Startdatei für statischen Web-Auftritt (http://PD331S) des unterlagerten Apache bereitstellen, welche auf die SVN-Repositories verweist.

Für diese Dateien gibt es im Anhang vollständige und teilweise erläuterte Beispiele.

```
@echo.
@chcp 28591
@echo Lokalen Apache + Subversion stoppen, Logs löschen, starten.
@echo.
net stop Apache2.2
@if /I %1X==--apStopX      goto :ende
@Echo Lokale Apache-Logs löschen
@echo.
java Update "C:\Programme\Subversion\httpd\logs"
              "E:\Arch\rettKannWeg%date%\apachLogs" -v
java Del "C:\Programme\Subversion\httpd\logs\+."
@Echo Lokalen Apache = Subversion starten
@echo.
net start Apache2.2
:ende
@echo.
```

Listing 5: Script, lokalen Apache stoppen, Logs weg und starten (u.A. für Tests).

Anmerkung zu Konfigurationsänderungen:

Am Anfang werden Sie mit der Konfiguration etwas experimentieren müssen. Änderungen an **httpd.conf**, **httpd-ssl.conf** und an **servers** erfordern immer einen Neustart, wie man ihn sich mit etwas wie Listing 5 bequem macht. Aber auch für andere Tests ist ein Neustart mit leeren logs gelegentlich sinnvoll. Sobald der produktive Betrieb mit allen Nutzern läuft, ist man mit Neustarts eher sparsam.

Wenn Apache bzw. SVN nicht starten und (nach Listing 5) das log-Verzeichnis C:\Programme\Subversion\httpd\logs\ leer bleibt, schaue man in die Windows-Ereignisse. Achtung der Dienst kann auch anders heißen, wie z.B. "CollabNet Subversion Apache".

3.3 Apache – Grundkonfiguration

Die Datei **httpd.conf** (vollständiges Beispiel im Anhang) ist die Grundkonfigurationsdatei. Listing 6 zeigt als Auszug die wesentlichen Einstellungen.

```
ServerRoot "C:/Programme/Subversion/httpd"
Listen 80
# Listen 443 see include below (is in the ssl include)

LoadModule ssl_module          modules/mod_ssl.so
LoadModule dav_module          modules/mod_dav.so
LoadModule dav_svn_module      modules/mod_dav_svn.so
LoadModule authz_svn_module    modules/mod_authz_svn.so
LoadModule sspi_auth_module    modules/mod_auth_sspi.so

ServerAdmin svn.31s.admin@meva-lab.de
ServerName pd331s
DocumentRoot "D:\www\serv-intra"

<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    Satisfy all
</Directory>

<Directory D:/www/serv-intra/>
    AllowOverride None
    Order allow,deny
    Allow from all
    Options Indexes FollowSymLinks
</Directory>

Include conf/extra/httpd-ssl.conf

<Location /svn/>
    DAV svn
    SVNParentPath E:/svn_repository
    SVNListParentPath on # may be considered as security-critical
    SVNIndexXSLT /conf/svnindex.xsl
    AuthzSVNAccessFile conf/authz.conf
```

```

AuthName "Subversion"
AuthType SSPI
SSPIAuth On
SSPIAuthoritative On
SSPIDomain FB3-MEVA
SSLRequireSSL
SSPIOmitDomain On      # omit domain name in userid string
SSPIOfferBasic On     # Required for TortoiseSVN, svn.exe, Eclipse etc.
SSPIUsernameCase lower
# require valid-user
<LimitExcept PROPFIND >
    require group "fb3-meva\\kernteam" "fb3-meva\\cax"
</LimitExcept>
</Location>

```

Listing 6: Apache Konfiguration für SVN (Auszug; siehe auch Anhang)

Hinweis: Die Syntax der Apache-Konfigurationsdateien ist an sich selbst erklärend. Für den erstmaligen Betrachter verwirrend ist das halbherzige "halt doch kein" XML.

Der Auszug Listing 6 aus **httpd.conf** zeigt nur die für SVN wesentlichen und teilweise an Ihre Verhältnisse anzupassenden Teile.

"**Listen 80**" und innerhalb **httpd-ssl.conf** "**Listen 443**" setzen die normalen http[s]-Ports für Zugriffe auf SVN und den übrigen auch statischen Content.

DocumentRoot "D:\www\serv-intra" bestimmt das genannte Verzeichnis als Wurzel für statischen Web-Content. Dorthin gehört eine Begrüßungs-index.html des Servers und auf dieses Verzeichnis beziehen sich alle Angaben für Stylesheets (css) und Transformationen (xsl), auch die vom SVN-Server genutzten.

Der Abschnitt **<Directory />...</Directory>** verbietet erst mal alle Zugriffe.

Der Abschnitt **<Directory D:\www\serv-intra/>...</Directory>** erlaubt alle lesenden Zugriffe auf das Wurzelverzeichnis für statischen Content.

Include conf/extra/httpd-ssl.conf benennt die gesonderte Datei zur Konfiguration des SSL-Zugriffs.

Der längere Abschnitt **<Location /svn/> DAV svn . <Location /svn/>..** macht für alle Zugriffe, die mit http[s]://PD331D/svn/ beginnen, das SVN-Modul zuständig.

Ohne **SSLRequireSSL** ist https optional, wenn die zugehörige SSL-Konfiguration OK ist. Mit **SSLRequireSSL** ist https für SVN zwingen. Dann muss auch die SSL-Konfiguration (siehe nächstes Kapitel) funktionieren. Zur Inbetriebnahme kommentiert man das auch schon mal aus.

SVNParentPath E:/svn_repository macht das genannte Verzeichnis zur Wurzel von (dort) beliebig vielen SVN-Repositories. Dies ist dem Nennen einzelner Repositories in jeweils eigenen Abschnitten der **httpd.conf** i.A. vorzuziehen, hauptsächlich weil dann jede (dynamische) Änderung der Repository-Struktur komplexere Konfigurationsänderungen und (!) Neustart erfordert.

Mit **SVNListParentPath on** erlauben Sie das Auflisten auch der Repositories. Ohne dies muss man die Repository-Namen wissen. Und mit **SVNIndexXSLT /conf/svnindex.xsl** machen sie alle Listings auf dem Browser beliebig schön. Der Browser muss (clientseitige) XSLT beherrschen. "/conf/svnindex.xsl" bezieht sich auf die Wurzel des statischen Web-Contents.

Mit den Eintragungen **SSPIxyz** "hängen Sie die Grund-Authentifizierung des ganzen SVN an das Active Directory der Domain FB3-MEVA (im Beispiel).

Mit **require group "fb3-meva\kernteam" "fb3-meva\cax"** lassen Sie nur Domain-Nutzer ran, deren Konten zu einer der genannten AD-Gruppen gehören. Das SSPI-Modul ist clever genug, indirekte Mitgliedschaften sowie beliebige ou und cn für die Mitglieds- und Gruppen-Konten zu können.

Anmerkung: Letzteres sollte für jede Software, die Authentifizierung gegen Windows-Domain-Konten anbietet selbstverständlich sein. Wie man allerlei Foren (zu Tomcat u.A.) entnehmen kann, wird derlei gerne "vergessen". Das dies hier Alles geht, muss vor diesem Hintergrund also lobend erwähnt werden. Auch hieran erkennt man den professionellen Ansatz von CollabNet .

Würde man stattdessen **require valid-user** einsetzen, wäre jeder gegen die Domain authentifizierte Nutzer erst mal zugelassen.

Mit **AuthzSVNAccessFile conf/authz.conf** bestimmt man, dass in der genannten Datei zusätzliche feingranularere Einschränkungen, i.A. repository-spezifisch, gemacht werden. Näheres dazu siehe im Anhang A2 ab Seite 36, insbesondere zur im AuthzSVNAccess-File gegebenen enttäuschenden Einschränkung bezüglich Windows-Gruppen.

Eine wesentliche Erleichterung für solche weiteren Festlegungen zur Autorisierung stellen die Anweisungen **SSPIomitDomain On** und **SSPIUsernameCase lower** dar. Ohne diese kommt man in allerlei Schreibweisenprobleme à la

"fb3-meva\weinert" versus "FB3-MEVA/Weinert".

Mit diesen Anweisungen wird dieser beispielhafte Nutzer nach Authentifizierung gegen die Domain nur noch als "weinert" geführt unabhängig davon wie die Schreibweise im AD ist und unabhängig davon, wie der Benutzer seinen Namen ggf. bei einer Authentifizierungsabfrage eintippte.

Der Eintrag **SSPIOfferBasic On** ist ein eher düsterer, oben bereits erläuterter Punkt; Näheres und Hintergrund hierzu siehe nochmal dort im Kapitel 2.2 ab Seite 6..

4. HTTPS — SSL und Nutzerauthentifizierung

4.1 Vorbemerkungen

Die gesicherte, verschlüsselte Übertragung von Informationen mit einem entsprechenden Protokoll (https z.B.) über ein Netzwerk ist eine Fähigkeit, die der dem SVN unterlagerte Apache-Server aus vielen Gründen und für viele Zwecke eh haben sollte. Die in Kapitel 2.2 oben ab Seite 6 geschilderten Authentifizierungsprobleme und ihre "Lösung" mit "SSPIOfferBasic On" allein machen SSL/HTTPS schon häufig zum "Muss".

4.2 https und ssl, Tomcat vs. Apache, Hintergrundinformation

Als Hintergrundinformation und Lösungshinweis mögen auch die entsprechenden Kapitel aus [13] dienen. Wer auf diese oder ähnliche Weise unter Windows HTTP und SSL für Tomcat, JMX und andere (Java-) Anwendungen quer über die Domain und hinaus am Laufen hat mag meinen, dass auch Apache dann all die (Server-) Zertifikate, Keystores, Truststores etc. pp. verwenden kann.

Konkret:

Einträge in der Datei

```
conf/extra/httpd-ssl.conf
```

(siehe Listing 7, Seite 18 weiter unten) in der Form

```
SSLCertificateFile "C:\Programme\localhost.crt"  
SSLCertificateKeyFile "C:\Programme\.keystore"
```

versuchen von allen bisherigen Anwendungen verwendete Zertifikate weiter zu nutzen.

Zur Verdeutlichung:

localhost.crt ist hier das anderen Clients bereits bekannte Zertifikat des genutzten Servers PD331S (im Beispiel). Dieses nicht zu verwenden, ist in vielerlei Hinsicht verkehrt. Aber alle solchen Versuche endeten mit Fehlermeldungen bzw. Erläuterungen der Art (Zitat):

```
"SSLPassPhraseDialog builtin is not supported on Win32. Unfortunately Apache on  
Windows does not support encrypted private keys. To resolve this you will need to  
remove the private key password."
```

Das muss man schon zweimal lesen und es sich "auf der Zunge zergehen" lassen.

Nochmal zur Erinnerung: der Grund für Apache

Apache wurde in diesem SVN-Server für Windows vor Allem (wenn auch nicht nur) genommen, weil nur mit Apache die Grund-Authentifizierung gegen eine Windows-AD-Domain funktioniert. Und genau dieses wegen Windows genommene Apache erfordert nun bei https Windows-widrige und vor Allem zu vorhandenem und bereits prima laufendem quer stehende, zusätzliche Ansätze. Das ist irgendwie widersprüchlich.

Die schlechte Nachricht dazu

Das Passwort des private key entfernen oder diesen im Klartext exportieren, geht mit dem (JDK-) keytool einfach nicht. (Das ist eigentlich eine gute Nachricht.) Also muss nur (!) für Apache das Ganze irgendwie mit openssl nochmal aufgezogen werden.

Und openssl unter Windows ist eher ein trübes Kapitel. Viele bisher ausprobierten Versionen von `openssl.exe` scheiterten kläglich mit einer Fehlermeldung:

can't open config file: /usr/local/ssl/openssl.cnf

Ja [sic!] in einer nach Windows portierten Version und im Allgemeinen dann auch durch dokumentierte Konfigurationseingriffe unreparierbar.

Die einzig funktionierende openssl.exe fand sich schließlich in der CollabNet Subversion Installation. Nehmen Sie unter Windows am besten keine anderen.

Die gute Nachricht dazu

Es gibt eigentlich keine richtig gute. Beide unten aufgezeigten Lösungswege können jeweils nicht voll befriedigen; der erste schon gar nicht, der zweite wohl eher.

Wenn Sie allerdings ein neues fremd unterzeichnetes Zertifikat (Ihres Providers) gehen Sie nach dessen der der CA Dokumentation für Apache vor. Dies ist i.A. recht einfach; allerdings landet meist der private Key in einer Klartextdatei auf dem betreffenden Server:

```
SSLCertificateKeyFile C:/.../private/private.key
```

4.3 https und ssl nur für Apache mit openSSL

Beide im Folgenden geschilderten Vorgehensweisen benötigen openssl.exe. Suchen Sie nun nicht im Internet nach frustrierenden (s.o.) Varianten, sondern nehmen Sie die mit der (CollabNet-) SVN-Server-Installation zugelaufene Implementierung.

Vorgehen 1

Im Verzeichnis C:\Programme\Subversion\httpd\conf\ssl führt man folgende Kommandos

```
openssl.exe req -config openssl.cnf -new -out my-server.csr
openssl.exe rsa -in privkey.pem -out my-server.key
openssl.exe x509 -in my-server.csr -out my-server.cert
               -req -signkey my-server.key -days 9365
```

aus. In openssl.cnf trägt man am besten vorher die zu den eigenen Anforderungen passenden defaults ein. Nach Beantworten einiger Fragen und dem Durchlaufen der aufgeführten Kommandos hat man in etwa folgende Dateien:

```
23.10.2008 12:06 1.024 .rnd
23.10.2008 13:32 603 my-server.cert
23.10.2008 12:06 525 my-server.csr
23.10.2008 12:15 887 my-server.key
23.10.2008 12:05 2.864 openssl.cnf
23.10.2008 12:06 963 privkey.pem
```

Hiervon benötigt man letztlich nur noch my-server.cert (Serverzertifikat) und my-server.key (private key) im Verzeichnis von C:\Programme\Subversion\httpd\conf\ssl .

Beide werden an entsprechender Stelle (siehe unten, Listing 7 Seite 18) in

```
C:\Programme\Subversion\httpd\conf\extra\httpd-ssl.conf
```

eingetragen.

Mit diesem Vorgehen 1 wurde mit openssl für den benutzten Application-Server "von Null an" ein Schlüsselpaar und ein Server-Zertifikat generiert. openssl.exe wurde genommen, damit es Apache passt. Wenn es bereits ein Zertifikat für diesen Server — in der Domain und darüber hinaus bekannt und vertraut — gab, so hat dieser Server nun ein total anders zweites Zertifikat mit allen vorhersehbaren Problemen.

Vorgehen 2

Dieses Vorgehen geht davon aus, dass der für "SVN on Apache" genutzte Application-Server bereits ein mit (Java-) keytool generiertes Server-Zertifikat plus keystore hat und dass dieses bekannte und vertraute Zertifikat weiterverwendet werden soll:

```
03.02.2006 13:02 1.353 pd331s-keystore
03.02.2006 13:02 585 pd331s.cer
```

Mit dem vorhandenen keystore und dem enthaltenen (verschlüsselten) private key kommt Apache, wie oben dargestellt, nicht klar. Wenn man das Zertifikat auch für Apache/SVN verwenden will muss man genau den zugehörigen private key (unverschlüsselt mit openssl) in einer Apache gängigen Form verpacken.

Hinweis: Ob man den vom Server und anderen Anwendungen bisher verwendeten private key in einer solchen Form zusätzlich speichern (exponieren) will, muss unter den security-Aspekten der bisher laufenden Anwendungen entschieden werden. Wird das Folgende ordentlich gemacht und werden die Spuren beseitigt, wird der Vorteil eines einheitlichen Server-Zertifikats die möglichen Nachteile zusätzlichen Gefahren überwiegen.

Das JDK-Werkzeug keytool wird den private key aus einem keystore nicht unverschlüsselt ausgeben oder exportieren. Nun muss es aber sein, und das Tool PKextr aus dem Framework frame4j kann (nur wegen Apache&OpenSSL) auch den private key exportieren mit:

```
> java de.frame4j.PKextr -ks pd331s-keystore PD331S
-v -pass denkste prkeyout -exportPrivate
.....sIJrrLjV8kUPu8Jaru7kcoxFNkOPr52PclU93Ha5Ugt+7ICdFXK
-----END PRIVATE KEY-----
private key in Datei "prkeyout" schreiben.
```

geschieht genau dies. "-ks pd331s-keystore PD331S" spezifizieren den keystore und den Alias des keys. "-pass denkste" gibt das Passwort an; ohne diese Angabe würde gefragt. "prkeyout" benennt die openssl-gängige Ausgabedatei, in die der key exportiert wird. "-exportPrivate" schließlich sagt, dass der private anstelle des public key exportiert werden soll. Mit "prkeyout" (pem) kann man nun — wie oben bei Vorgehen 1 — mit openssl ein Apache-gängige private key-file erzeugen:

```
> openssl.exe rsa -in prkeyout -out the-server.key
```

```
openssl rsa -in prkeyout -out the-server.key
writing RSA key
```

```

Listen 443
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl .crl
SSLSessionCache
"shmcb:D:/vinithra/build/svn_1.5.3/subversion/../../httpd/tmp/httpd/A
pache/logs/ssl_scache(512000)"
SSLSessionCacheTimeout 300
SSLMutex default
<VirtualHost _default_:443>
  DocumentRoot "D:/www/serv-intra"
  ServerName pd331s
  ServerAdmin svn.31s.admin@meva-lab.de
  SSLEngine on
  SSLCipherSuite
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL

  SSLCertificateFile
    "C:/Programme/Subversion/httpd/conf/ssl/my-server.cert"
    #! geht nicht mit "C:\Programme\localhost.cert"

  SSLCertificateKeyFile
    "C:/Programme/Subversion/httpd/conf/ssl/my-server.key"
    #! geht nicht mit "C:\Programme\keystore"

  <FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
  </FilesMatch>

  <Directory "D:/www/serv-intra">
    SSLOptions +StdEnvVars
  </Directory>

  BrowserMatch ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
  CustomLog "logs/ssl_request.log" \
    "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>

```

Listing 7: Apache Konfiguration für SSL — Auszug der Datei ...\\extra\\httpd-ssl.conf
(Diese Datei wird mit include praktisch ein Teil der httd-conf.)

Nach diesen Schritten haben Sie folgende Dateien (am Besten gleich) im Verzeichnis

```
C:\Programme\Subversion\httpd\conf\ssl\..
03.11.2008 16:02 1.353 pd331s-keystore
03.11.2008 16:02 585 pd331s.cer
06.11.2008 13:47 921 prkeyout
06.11.2008 13:53 887 the-server.key
```

Löschen Sie nun sofort die Zwischendatei `prkeyout`, welche den private key des Servers im Klartext enthält! Beseitigen Sie Spuren und "wiederbelebte" Dateien.

Beschränken Sie die Zugriffsrechte auf die (unverschlüsselte OpenSSL-) Datei `the-server.key` im Wesentlichen auf Leserechte für das lokale System-Konto, unter dem Apache2.2 und damit SVN als Dienst läuft.

Setzen Sie nun in

```
C:\Programme\Subversion\httpd\conf\extra\httpd-ssl.conf
```

(Listing 7 unten, Seite 18) die folgenden zwei Einträge:

```
SSLCertificateFile "C:/Programme/Subversion/httpd/conf/ssl/pd331s.cer"
```

```
# Server Private Key:
```

```
SSLCertificateKeyFile "C:/Programme/Subversion/httpd/conf/ssl/the-server.key"
```

Nun nutzen Apache & SVN dasselbe (mit `keytool` generierte) Serverzertifikat wie alle anderen (Java-) Anwendungen (Tomcat, JMX etc.) auf diesem Server. Vorhandene Vertrauensstellungen, `truststores` etc. können weiterverwendet werden; siehe auch [13].

4.4 Erzwingen von https (und von Nutzer-Authentifizierung)

Nun hat der Nutzer die Wahl, `http` oder `https` zu nehmen. I.A. möchte man für bestimmte (kritische) Teile des SVN-Bereichs die Verwendung von `https` erzwingen. Hierfür reicht nun in der `http.conf`:

```
<Location /svn/>
    SSLRequireSSL
```

Mit dieser einfachen Variante sind nur noch `https`-Zugriffe auf alle Repositories erlaubt und `http` wird schlicht abgelehnt.

Diese Variante allein mag für viele Anwendungen (vor Allem in-house) genügen. Alle Links auf den SVN-Server müssen dann einfach "`https://pd331s/svn/...`" (im Beispiel hier) sein. Dies gilt dann selbstverständlich auch für entsprechende "locations" von Eclipse.

Zusätzlich möchte man oft automatisch `https` nutzen, auch wenn der Client eine Seite dieses Teilbereichs mit nur `http://` anfordert. Dies ist für menschliche Browser-Nutzer komfortabler und bringt Suchmaschinen und Lesezeichen eher auf den richtigen Weg.

Eine solche http-nach-https-Automatik ist bei Apache viel komplexer und fehlerträchtiger als bei Tomcat. Im übrigen wimmelt das (hierzu gegoogelte) Web von Problembestätigungen und Lösungsvorschlägen

Der (noch einfache und richtige) Anfang überwiegend aller Lösungen ist folgender zusätzlicher Eintrag an der betreffenden Stelle in http.conf :

```
LoadModule rewrite_module modules/mod_rewrite.so
```

Darüber hinaus haben alle (und das waren sehr viele) durchprobierten Lösungsvorschläge in unserem SVN-Zusammenhang kläglich versagt. Damit sei nichts Negatives über die Brauchbarkeit der angebotenen Hinweise in anderen Apache-Zusammenhängen gesagt. Hier (DAV svn; AuthType SSPI) warren deren häufigste (Neben-) Wirkungen: "dicht Machen", "gar nichts Tun", zweimalige Login-Abfrage mit einmal Klartextübertragung und div. andere Missheligkeiten mehr.

Hinweis: Zu Einwände der Art "Mein viel besserer / einfacherer /schnellerer Ansatz xyz geht aber auch" gibt es zwei mögliche Antworten:

- a) kann sein;
dann wurde er bis zur Zeit des Schreibens nicht gefunden bzw. übersehen.
- b) Eine Lösung gilt hier nur dann als akzeptabel, wenn sie unter allen Kombinationen der folgenden Umstände läuft: member-computer oder nicht, (Windows-/ Domain-) eingeloggt als berechtigter SVN-client oder nicht, Zugriffe mit svn.exe bzw. mit Eclipse oder über einen Browser sowie Zugriffe auch von Nicht-Windows-Maschinen. Die notorischen "bei mir geht aber"-Einwender übersehen üblicherweise 90% der möglichen Bedingungen.

Die svn-via-https-Automatik lässt sich in drei Schritten etablieren.

1. Das rewrite-Modul des Apache muss geladen sein (siehe oben).
2. Die gesamte SVN-Konfiguration zwischen <Location /svn/> und </Location> wird unverändert vor das </VirtualHost> am Ende der Datei httpd-ssl.conf verschoben (nicht kopiert); vgl. Listing 7 Seite 18. Dies macht den SVN-Server zum Teil des <VirtualHost _default_:443>.
3. Genau an die von der SVN-Konfiguration "verlassene" Stelle in der Datei httpd.conf kommen die Anweisungen aus Listing 8.

Hinweis: Die beiden letzten Punkte sind in den übrigen Beispieldateien und -konfigurationen, vor Allem im Anhang, nicht enthalten.

```
<VirtualHost _default_:80>
  <Location /svn/>
    RewriteEngine on
    RewriteCond %{SERVER_PORT} !443
    RewriteRule ^(.*)$ https://pd331s%{REQUEST_URI} [R=301,L]
  </Location>
</VirtualHost>
```

Listing 8: Apache Konfiguration für SVN (Auszug; siehe auch Anhang)

5. Hinweise zur Active Directory Integration

Die Grundeinbindung von AD in "SVN on Apache" wurde schon in Kapitel 3.3 ab Seite 12 vollständig geschildert. Sie ist viel einfacher zu bewerkstelligen als bei Tomcat. Die Lösung trägt dann leider aber viel weniger weit.

Wie in Kapitel 3.3 dargelegt hat man in der Datei `http.conf` folgende Möglichkeiten:

Mit **`require group "fb3-meval\kernteam" "fb3-meval\cax"`** lassen Sie an SVN nur Domain-Nutzer ran, deren Konten zu mindestens einer der genannten AD-Gruppen gehören. Das SSPI-Modul ist clever genug, indirekte Mitgliedschaften und beliebige `ou` und `cn` für die Mitglieds- und Gruppen-Konten zu können.

Würde man, wie oben schon gesagt, stattdessen **`require valid-user`** einsetzen, wäre jeder gegen die Domain authentifizierte Nutzer erst mal zugelassen.

Darauf aufbauend kann man mit **`AuthzSVNAccessFile conf/authz.conf`** festlegen, dass in der genannten Datei zusätzliche feingranularere Einschränkungen, i.A. nur repository-spezifisch sinnvoll, gemacht werden. Näheres dazu siehe im Anhang A2 ab Seite 36.

Dort finden Sie auch Näheres zu den dabei gegebenen enttäuschenden Einschränkung bezüglich Windows-Gruppen. Diese Einschränkungen besagen, dass

- a) diese feingranulareren Festlegungen von SVN ohne Apache's-Hilfe und -Kenntnisse ausgewertet werden und dass
- b) SVN von AD Gruppen absolut nichts weiß [sic!].

Hierher werden lediglich die Namen des authentifizierten AD-Kontos gemeldet und nur darauf aufbauend werden Rechte vergeben.

Dort definierte Gruppen haben mit AD-Gruppen nichts zu tun!

Zur Zeit hat man wohl keine andere Wahl als diesen Missstand hinzunehmen.

Unter diesen Umständen kann es oft es sinnvoller scheinen, auch in der Grund-Authentifizierung von SVN gegen AD einfach nur **`require valid-user`** zu nehmen und nun, als zwangsläufige Konsequenz, alle (alle!) Einschränkungen im `AuthzSVNAccessFile` zu regeln. Da dies wie gesagt nur Domain-Nutzer-bezogen geht, wird dies bei jeweils großen Nutzerkreisen einzelner Repositories leider beliebig aufwändig.

6. Wenige Hinweise zum Betrieb

6.1 Repository anlegen

Zum Anlegen eines neuen Repositories ist man mit entsprechenden Rechten (remote) am betreffenden Server (PD331S) angemeldet.

Insbesondere, wenn man die SVN-Standard-Gewohnheit mit den drei Repository-Bereichen "trunk, branches und tags" annehmen möchte, folgt man hier einfach dem Rezept im SVN-Buch ([17]).

Nach dem Anlegen mit "svn admin create ..." können Sie das leere repository gleich mit Eclipse (Neues Projekt ...) auschecken und dort alle Verzeichnisse und Grunddateien anlegen. Dies gilt auch für das Anlegen eines "repository template's" (s.u.). So passiert es jedenfalls nicht, dass versehentlich (reine) Betriebssystemfunktionen verwendet werden, von denen dann SVN nichts mitbekommt.

6.2 Repositories sichern, kopieren und löschen

Wenn man (standardmäßig) das SVN-eigene FSFS als Datenbank für den SVN-Server nimmt, wird jedes Repository ein eigener Verzeichnisbaum, dessen innere Struktur man allerdings in Ruhe lassen sollte. Der Name des Wurzelverzeichnisses dieses Baums ist der Repository-Name.

Dies gesagt ist klar, dass man mit Windows-Mitteln Repositories komplett und auf einfachste Weise

- umbenennen
- kopieren
- sichern und
- löschen

kann. Löschen (oder den Baum weg Bewegen) in diesem Sinne heißt dann auch, dass aus SVN-Serversicht das Repository komplett und spurlos weg ist. Das ist natürlich etwas ganz anderes als ein Löschen oder "roll back" mit SVN-Mitteln: Ein Versionsverwaltungssystem vergisst auf diese Weise nicht wirklich irgend etwas.

Durch diese FSFS-Möglichkeiten kann man nun das etwas umständliche administrative Herstellen eines neuen leeren Repositories rationalisieren:

Man macht das einmal (gründlich und ordentlich) und hält sich dieses "Werk" ab da unverändert als Kopiervorlage.

6.3 keyword substitution und "deploy without keywords"

Sie muss bei SVN (Datei-weise) via einer property eingeschaltet werden. Wenn sie nur Textdateien (.txt, .asm, .bat, .cmd, .java, .html, .xml, .xsl, .css etc. pp.) bestellen Sie die keyword substitution für alle Dateien gleich so:

```
propset -R svn:keywords "Date Author Revision" path/projX
```

Das geht auch mit Eclipse (Team > set property).

Wenn Sie eine (ausgecheckte, "committete") Version als Kopie (!) verbreiten (deployen) wollen machen Sie mit

```
java de.frame4j.CVSkeys -v D:\deploy\SVNtheCopy\
```

wie schon mit cvsNT die Substitutionen geschönt keyword-frei.

Von cvs[NT] zu SVN haben sich zwar interne Handhabung und Bedeutung der keywords geändert, das Werkzeug cvsKeys funktioniert aber ja für Beides:

a-weinert.de/java/docs/frame4j/de/frame4j/CVSkeys.html

6.4 "multi file / directory" - Operationen

Obgleich viele SVN-Kommandos file-Angaben mit Wildcards sowie optional den rekursiven Abstieg (-R) erlauben, machen sie beides i.A. nicht wirklich gut.

Die Mängel sind:

- nur eine Dateiangebe (keine Liste von Dateien oder Wildcard-Ausdrücken)
- Wildcard-Expansion wird Windows überlassen ¹⁾
- SVN-eigene versteckte Dateien und Verzeichnisse (.svn) werden besucht ²⁾.

Alle diese Mängel lassen sich mit etwas wie Listing 9 umgehen.

```
> set svnCVStextTypes=*.java *.html *.properties *.mf *.bat
*.xml *.txt *.asm *.cmd *.htm *.xsl *.xsd *.dtd
*.ent *.cpp *.php *.ini *.psf *.js

> set svnKeywords=Author Revision Date Id HeadURL

> for /r %i in (%svnCVStextTypes%) do
    %svnexe% propset svn:keywords "%svnKeywords%" %i
```

Listing 9: Auch rekursives Besuchen auch mehrerer Dateien mit SVN-Kommandos.

Hinweis: Listing 9 ist als konkretes Beispiel auch gleich der Trick, um die sonst nicht funktionierende "keyword expansion" (vgl. letztes Kapitel) für die Nicht-Binär-Dateien einzuschalten. Aktuelles Verzeichnis muss die Wurzel des (ausgecheckten) betreffenden Bereichs sein.

-
- 1) Dies bedeutet a) "*.txt" funktioniert gar nicht und b) *.txt funktioniert rekursiv (-R) nur für die erste im Grundverzeichnis gefundene Datei (also eigentlich auch gar nicht).
 - 2) Selbst falls dadurch nichts Schlimmes passieren sollte, wird der Anwender durch kryptische Meldungen à la "blah blah ends in a reserved name" genervt oder verunsichert.

7. Resumme

7.1 Erreicht mit CollabNet Subversion

Dieses Tutorial zeigte von Null an, also auch für Ersteinsteiger, das Aufsetzen eines (CollabNet-) Subversion in einem Windows- / Active-Directory- / Domain-Umfeld. Domain-Nutzer und -Gruppenkonten werden für Authentifizierungszwecke verwendet.

Dies setzt Apache statt svnServe als unterlagerten Server voraus — eine Entscheidung die über den dann möglichen komfortabel gestaltbaren Browser-Zugriff auf den SVN-Server nur zusätzliche Vorteile bietet.

Der Wermutstropfen ist, dass bei der (feingranularen) Repository-Autorisierung hier der AD-Weg nur halb gegangen wurde. Das verwendete Apache kann Windows-AD und der SVN-Server selbst ist da total ahnungslos.

7.2 Warum SVN? (statt cvs[NT])

Das Folgende soll nichts zur teilweise schon mit religiösen Zügen geführte CVS-/SVN-Diskussion beitragen, sondern lediglich ein paar Fakten als Entscheidungs- und Umstiegs-hinweise nennen.

Inzwischen hat sich SVN zu dem Standard bei Software-Projekten in-house und open source entwickelt (und mit GIT wächst im in Teilbereichen ein Konkurrent nach).

Die Features und das Verhalten des Tools (als Versionsverwaltungssystem) sind weitgehend gut und bei atomic commit (mehrerer Dateien) Umbenennen und Verschieben besser als CVS.

Man kann auch sagen, dass CVS beim letzten Aspekt so schlecht ist, das es manches notwendige re-factoring in Java-Projekten effektiv verhindert. Die Werkzeug-Unterstützung ist umfassend. Nach jetzigen Stand des Geschriebenen bleiben im wesentlichen die oben schon genannten zwei offenen Punkte.

Viele Menschen sprechen und korrespondieren inzwischen, ohne das aber zu sagen oder zu erläutern, einfach in "SVN-speak" bzw. meist in "Linux-SVN-speak". Schon um da mitdenken zu können, muss man also SVN ganz gut kennen:

Wer eine pure Aufforderung "Gib mit ein Patch gegen trunk" nicht versteht (und vielleicht an ein Nikotin-Pflaster gegen Alkoholsucht denkt) gilt bei manchem Absender solcher Botschaften als abgrundtief blöd. (Fragt sich natürlich, wer da geistig beschränkter ist.)

Wo Licht ist, ...Die Schattenseiten von SVN

Die folgenden Nachteile muss nun (leider) insbesondere der Umsteiger von CVS bedenken.

Da SVN den Anspruch des (besseren) CVS-Nachfolgers erhebt, muss man sogar von entfallenen Features sprechen.

- Die keyword-substitution ist bei SVN viel umständlicher in der Handhabung, und das bis zu einem schon ärgerlichen Grade, welcher die (bisherige) Verwendbarkeit ernsthaft einschränkt.
- SVN kann und hat keine Tags im CVS-Sinne. (Die SVN-Tags sind einfach Kopien und damit was total anderes.) Das entsprechende keyword "\$Name: Kunden3V0\$" ist dementsprechend entfallen.

- SVN kann / hat einfach keine Versionierung von Dateien. Punkt. Dementsprechend ist auch die ursprüngliche (CVS-) Bedeutung von \$Revision: 99\$ schlicht weg.

Die sattsam bekannten SVN-Gegenargumente insbesondere zum letzten Punkt sind sinngemäß weitgehend:

"Ist / war eh Blödsinn; das brauchen Sie gar nicht."

Nun ist es einerseits per se falsch, wenn der Lieferant eines Werkzeugnachfolgemodells dem Kunden oder Nutzer ins Gesicht sagt

"Linkslauf der Bohrmaschine haben Sie nie gebraucht
(und wenn Sie nicht so blöd wären, wüssten Sie's selbst)".

Datei-Versionierung ist einfach für viele Anwendungsfälle von Versionsverwaltungssystemen (abseits der Softwareentwicklung) sinnvoll und wichtig. Auch insofern sind solche Argumente schlicht und einfach falsch. Wer's also brauchte und braucht, ist mit SVN schlecht bedient.

Was bei der SVN-Architekturfindung ursprünglich wohl passiert sein mag, ist dass man "atomic commit mehrerer Dateien" und "Versionierung jeder einzelnen Datei" diskussionshalber in einen Topf warf. (Dafür spricht, dass man Entsprechendes immer noch liest und hört.) Und vermutlich aufgrund dieses Denkfehlers hat man dann mit der Einführung des Einen das Andere über Bord geworfen.

Ein ganz harter (von den Entwicklern mit Argumenten obiger Qualität als feature verteidigter) Bug ist

- SVN macht das ursprüngliche Datei-Modifikationsdatum unwiderruflich beim ersten (ursprünglichen) check in kaputt.

Wenn Sie also Hunderte von Bildern, Unterlagen, Fotos etc., aus denen Sie andere Dokumente, Webauftritte etc. generieren über 10 und mehr Jahre gesammelt haben und sie diese endlich(!) unter Versionsverwaltung stellen wollen,

- ist dies eine gute Absicht, aber
- SVN hierfür eigentlich total ungeeignet. (Alle Ihre Photos und Dokumente sind ab da am selben Tag des ersten check in entstanden.)

Reine Softwareentwickler, insbesondere die von SVN selbst, vermögen nicht, den letzten Punkt als harten Fehler zu erkennen. Aber auch sie verlieren u.v.a. das Datum des letzten check in, wenn sie wegen Server-/Provider-Absturz z.B. ein repository aus lokalen Daten neu aufsetzen.

7.3 Summa summarum — doch eine gute Wahl

Von allen erwähnten Misshelligkeiten und Bugs abgesehen, ist für Softwareprojekte SVN heute der sinnvolle Standard bei Software-Projekten aller Größen. Wenn man nicht bei größeren Projekten mit den (wenigen) bei cvs und insbesondere cvsNT besseren features fest "verheiratet" ist, gibt es keinen Grund, nicht SVN zu nehmen oder dorthin zu wechseln.

Dies gilt, gerade in einem Windows-Umfeld, wenn man die professionelle und dennoch lizenzgebührenfreie SVN-Zusammenstellung von CollabNet nimmt.

Hierzu kommt gerade bei Windows "Tortoise" als geniale und (meist) keine Probleme machende Integration in den Explorer — für den Windows-SVN-Nutzer (als client) fast schon ein "must have".

Anhang

A1 SVN – Server Konfigurationsdatei

Das folgende vollständige `httpd.conf` – Beispiel zeigt die Einstellungen für die SVN-Autentifizierung gegen AD-Gruppen. Es ist tatsächlich eine Konfigurationsdatei des dem SVN unterlagerten Apache.

Wer so was zum ersten mal sieht, wundert sich mit Recht. Ja es ist ein solches halbherziges Nicht-XML. (Ja, warum den nicht?)

Mit XML-Tools rangehen, analysieren, modifizieren etc. hätte was, geht aber nicht.

```
----- C:\Programme\Subversion\httpd\conf\httpd.conf -----

# Subversion 1.5.3 on Apache 2.2
# 17:20 25.10.2008
# Albrecht weinert

# This is the main Apache HTTP server configuration file.
# See <URL:http://httpd.apache.org/docs/2.2/> for detailed information, in
# particular <URL:http://httpd.apache.org/docs/2.2/mod/directives.html>

# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32),
# the server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/foo.log"
# with ServerRoot set to "C:\Programme\Subversion/httpd" will be
# interpreted by theserver as "C:\Programme\Subversion/httpd/logs/foo.log".

# NOTE: Where filenames are specified, you must use forward slashes
# instead of backslashes (e.g., "c:/apache" instead of "c:\apache").
# If a drive letter is omitted, the drive on which Apache.exe is located
# will be used by default.  It is recommended that you always supply
# an explicit drive letter in absolute paths, however, to avoid
# confusion.

# ThreadsPerChild: constant number of worker threads in the server process
# MaxRequestsPerChild: maximum number of requests a server process serves
ThreadsPerChild 250
MaxRequestsPerChild 0

# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
# Do not add a slash at the end of the directory path.  If you point
# ServerRoot at a non-local disk, be sure to point the LockFile directive
# at a local disk.  If you wish to share the same ServerRoot for multiple
# httpd daemons, you will need to change at least LockFile and PidFile.
```

```

ServerRoot "C:/Programme/Subversion/httpd"

# : Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
#Listen 12.34.56.78:80
Listen 80
# Listen 443 see include below (this is in the ssl include)

# Dynamic Shared Object (DSO) Support
# To be able to use the functionality of a module which was built as a DSO
# you have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by `httpd -l') do not need
# to be loaded here. Example:
# LoadModule foo_module modules/mod_foo.so

LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
#LoadModule authn_anon_module modules/mod_authn_anon.so
#LoadModule authn_dbm_module modules/mod_authn_dbm.so
LoadModule authn_default_module modules/mod_authn_default.so
LoadModule authn_file_module modules/mod_authn_file.so
#LoadModule authz_dbm_module modules/mod_authz_dbm.so
LoadModule authz_default_module modules/mod_authz_default.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule autoindex_module modules/mod_autoindex.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule cgi_module modules/mod_cgi.so
#LoadModule dav_module modules/mod_dav.so
#LoadModule dav_fs_module modules/mod_dav_fs.so
#LoadModule deflate_module modules/mod_deflate.so
LoadModule dir_module modules/mod_dir.so
LoadModule env_module modules/mod_env.so
#LoadModule expires_module modules/mod_expires.so
#LoadModule file_cache_module modules/mod_file_cache.so
#LoadModule headers_module modules/mod_headers.so
LoadModule imagemap_module modules/mod_imagemap.so
LoadModule include_module modules/mod_include.so

```

```

#LoadModule info_module modules/mod_info.so
LoadModule isapi_module modules/mod_isapi.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
#LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
#LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
#LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule negotiation_module modules/mod_negotiation.so
#LoadModule rewrite_module modules/mod_rewrite.so
LoadModule setenvif_module modules/mod_setenvif.so
#LoadModule spelling_module modules/mod_spelling.so
#LoadModule status_module modules/mod_status.so
#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule userdir_module modules/mod_userdir.so
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so
LoadModule ssl_module modules/mod_ssl.so
LoadModule dav_module      modules/mod_dav.so
LoadModule dav_svn_module  modules/mod_dav_svn.so

LoadModule authz_svn_module  modules/mod_authz_svn.so
LoadModule sspi_auth_module modules/mod_auth_sspi.so

# 'Main' server configuration
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition.  These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.

# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.  e.g. admin@your-domain.com
ServerAdmin svn.31S.admin@meva-lab.de

# ServerName gives the name and port that the server uses to identify
# itself.  This can often be determined automatically, but we recommend
# you specify it explicitly to prevent problems during startup.  If your
# host doesn't have a registered DNS name, enter its IP address here.
ServerName pd331s

```

```

#ServerName pd331s.fh-bochum.de

# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
DocumentRoot "D:\www\serv-intra"

# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
# First, we configure the "default" to be a very restrictive set of
# features.
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    Satisfy all
</Directory>

# Note that from this point forward you must specifically allow particular
# features to be enabled - so if something's not working as you might
# expect, make sure that you have specifically enabled it below.
# This should be changed to whatever you set DocumentRoot to.
# ex "C:\Programme\Subversion\httpd\htdocs"

# Allgemeine statische Inhalte (Apache als Webserver)
<Directory D:/www/serv-intra/>
    # Possible values for the Options directive are "None", "All",
    # or any combination of: Indexes Includes FollowSymLinks
    # SymLinksifOwnerMatch ExecCGI MultiViews
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    # The Options directive is both complicated and important. Please see
    # http://httpd.apache.org/docs/2.2/mod/core.html#options
    # for more information.
    Options Indexes FollowSymLinks

    # AllowOverride controls what directives may be placed
    # in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
    # Options FileInfo AuthConfig Limit
    AllowOverride None

    # Controls who can get stuff from this server.
    Order allow,deny
    Allow from all

```

```

</Directory>

# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
<FilesMatch "^\.ht">
    Order allow,deny
    Deny from all
</FilesMatch>

# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
ErrorLog logs/error.log

# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

<IfModule log_config_module>
    #
    # The following directives define some format nicknames for use with
    # a CustomLog directive (see below).
    #
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common
    <IfModule logio_module>
        # You need to enable mod_logio.c to use %I and %O
        LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O" combinedio
    </IfModule>

    # The location and format of the access logfile (Common Logfile Format).
    # If you do not define any access logfiles within a <VirtualHost>
    # container, they will be logged here.  Contrariwise, if you *do*
    # define per-<VirtualHost> access logfiles, transactions will be
    # logged therein and *not* in this file.
    CustomLog logs/access.log common

    # If you prefer a logfile with access, agent, and referer information
    # (Combined Logfile Format) you can use the following directive.

```

```

    #CustomLog logs/access.log combined
</IfModule>

<IfModule alias_module>
    # Redirect: Allows you to tell clients about documents that used to
    # exist in your server's namespace, but do not anymore. The client
    # will make a new request for the document at its new location.
    # Example:
    # Redirect permanent /foo http://www.example.com/bar

    # Alias: Maps web paths into filesystem paths and is used to
    # access content that does not live under the DocumentRoot.
    # Example:
    # Alias /webpath /full/filesystem/path

    # If you include a trailing / on /webpath then the server will
    # require it to be present in the URL. You will also likely
    # need to provide a <Directory> section to allow access to
    # the filesystem path.

    #
    # ScriptAlias: This controls which directories contain server scripts.
    # ScriptAliases are essentially the same as Aliases, except that
    # documents in the target directory are treated as applications and
    # run by the server when requested rather than as documents sent to the
    # client. The same rules about trailing "/" apply to ScriptAlias
    # directives as to Alias.
    ScriptAlias /cgi-bin/ "C:\Programme\Subversion\httpd/cgi-bin/"
</IfModule>

# "C:\Programme\Subversion\httpd/cgi-bin" should be changed to whatever
# your ScriptAliased CGI directory exists, if you have that configured.
<Directory "C:\Programme\Subversion\httpd/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

# Apache parses all CGI scripts for the shebang line by default.
# This comment line, the first line of the script, consists of the symbols
# pound (#) and exclamation (!) followed by the path of the program that
# can execute this specific script. For a perl script, with perl.exe in
# the C:\Program Files\Perl directory, the shebang line should be:

#!c:/program files/perl/perl

```

```

# Note you must not indent the actual shebang line, and it must be the
# first line of the file. Of course, CGI processing must be enabled by
# the appropriate ScriptAlias or Options ExecCGI directives for the files
# or directory in question.

# However, Apache on Windows allows either the Unix behavior above, or can
# use the Registry to match files by extension. The command to execute
# a file of this type is retrieved from the registry by the same method as
# the Windows Explorer would use to handle double-clicking on a file.
# These script actions can be configured from the Windows Explorer View
# menu, 'Folder Options', and reviewing the 'File Types' tab. Clicking
# the Edit button allows you to modify the Actions, of which Apache 1.3
# attempts to perform the 'Open' Action, and failing that it will try the
# shebang line. This behavior is subject to change in Apache release 2.0.

# To enable the this Windows specific behavior (and therefore disable the
# equivalent Unix behavior), uncomment the following directive:

#ScriptInterpreterSource registry

# The directive above can be placed in individual <Directory> blocks or the
# .htaccess file, with either the 'registry' (Windows behavior) or 'script'
# (Unix behavior) option, and will override this server default option.

# DefaultType: the default MIME type the server will use for a document
# if it cannot otherwise determine one, such as from filename extensions.
# If your server contains mostly text or HTML documents, "text/plain" is
# a good value. If most of your content is binary, such as applications
# or images, you may want to use "application/octet-stream" instead to
# keep browsers from trying to display binary files as though they are
# text.
DefaultType text/plain

<IfModule mime_module>
    # TypesConfig points to the file containing the list of mappings from
    # filename extension to MIME-type.
    TypesConfig conf/mime.types

    # AddType allows you to add to or override the MIME configuration
    # file specified in TypesConfig for specific file types.

    #AddType application/x-gzip .tgz

    # AddEncoding allows you to have certain browsers uncompress
    # information on the fly. Note: Not all browsers support this.

    #AddEncoding x-compress .Z

```

```

#AddEncoding x-gzip .gz .tgz

# If the AddEncoding directives above are commented-out, then you
# probably should define those extensions to indicate media types:

AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz

# AddHandler allows you to map certain file extensions to "handlers":
# actions unrelated to filetype. These can be either built into the
# server or added with the Action directive (see below)

# To use CGI scripts outside of ScriptAliased directories:
# (You will also need to add "ExecCGI" to the "Options" directive.)

#AddHandler cgi-script .cgi

# For type maps (negotiated resources):
#AddHandler type-map var

# Filters allow you to process content before it is sent to the client.
# To parse .shtml files for server-side includes (SSI):
# (You will also need to add "Includes" to the "Options" directive.)

#AddType text/html .shtml
#AddOutputFilter INCLUDES .shtml
</IfModule>

# The mod_mime_magic module allows the server to use various hints from the
# contents of the file itself to determine its type. The MIMEMagicFile
# directive tells the module where the hint definitions are located.

#MIMEMagicFile conf/magic

# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects

# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html

# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall is used to deliver
# files. This usually improves server performance, but must
# be turned off when serving from networked-mounted

```

```
# filesystems or if support for these functions is otherwise
# broken on your system.

#EnableMMAP off
#EnableSendfile off

# Supplemental configuration

# The configuration files in the conf/extra/ directory can be
# included to add extra features or to modify the default configuration of
# the server, or you may simply copy their contents here and change as
# necessary.

# Server-pool management (MPM specific)
#Include conf/extra/httpd-mpm.conf

# Multi-language error messages
#Include conf/extra/httpd-multilang-errordoc.conf

# Fancy directory listings
#Include conf/extra/httpd-autoindex.conf

# Language settings
#Include conf/extra/httpd-languages.conf

# User home directories
#Include conf/extra/httpd-userdir.conf

# Real-time info on requests and configuration
#Include conf/extra/httpd-info.conf

# Virtual hosts
#Include conf/extra/httpd-vhosts.conf

# Local access to the Apache HTTP Server Manual
#Include conf/extra/httpd-manual.conf

# Distributed authoring and versioning (WebDAV)
#Include conf/extra/httpd-dav.conf

# Various default settings
#Include conf/extra/httpd-default.conf

# Secure (SSL/TLS) connections
Include conf/extra/httpd-ssl.conf
#
# Note: The following must must be present to support
```

```

#         starting without SSL on platforms with no /dev/random equivalent
#         but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>

## Subversion - Konfiguration
## Subversion Parent-Dir of all repositories

<Location /svn/>
    DAV svn
    SVNParentPath E:/svn_repository
    # allow listing of repositories (u.U. security-kritisch)
    SVNListParentPath on
    SVNIndexXSLT /conf/svnindex.xsl

    # our access control policy (just for repository level)
    AuthzSVNAccessFile conf/authz.conf

    AuthName "Subversion"
    AuthType SSPI
    SSPIAuth On
    SSPIAuthoritative On
    SSPIDomain FB3-MEVA
    SSLRequireSSL

    SSPIOmitDomain On          # omit domain name in userid string
    SSPIOfferBasic On          # Required for TortoiseSVN, svn.exe, Eclipse etc.
                                # to get at repository, if on a non-Domain-PC
# SSPIBasicPreferred Off # should basic authentication have higher priority
    SSPIUsernameCase lower

    # require valid-user
    #<LimitExcept GET PROPFIND OPTIONS REPORT>
    <LimitExcept PROPFIND >
        require group "fb3-meva\\kernteam" "fb3-meva\\cax"
    </LimitExcept> # require group LDAPreader

</Location>

----- httpd.conf (Ende) -----

```

A2 Autorisierung als getrennte Konfiguration

Für die repository-spezifische Zugriffs-Autorisierung kann bzw. muss praktisch eine im folgenden Beispiel gezeigte SVN-eigene Konfigurationsdatei verwendet werden.

Eine solche Datei erlaubt auch wesentlich feingranularere Einstellungen bis hin zu verzeichnis- und dateispezifischen Rechten. Dies ist im Zusammenhang mit Versionsverwaltung und Teamarbeit aber selten sinnvoll oder notwendig. Und es ist auf Dauer viel schwerer handhabbar als in "normalen" Dateisystemen, da Rechte auch für alte gelöschte und umbenannte Dateien und Verzeichnisse weiterhin zutreffend verwaltet werden müssten.

Die absolute schlechte Nachricht ist, dass man in der authz.conf keine AD-Gruppen verwenden kann. Das ist ein eklanter Widerspruch zur SSPI-Authentifizierung (s.o. in `<Location>..</Location>`).

Unter bestimmten (Firmen-) Rahmenbedingungen ist dies ein "show stopper".

Zitat zum Hintergrund: "It boils down to Subversion, at present, can authenticate but not authorise against an external source. Subversion's authz component has its own concept of group that is unrelated to whatever is in the external source."

```
----      C:\Programme\Subversion\httpd\conf\authz.conf      -----  
  
## 30.10.2008 weiteres Repo  
# Subversion authorisation  
# Server PD331S  
# nur für repository-spezifische Einstellungen  
  
[groups]  
frameWcommitter = weinert, dfb30017  
frameWreader = weinert, dfb30017, seidel  
  
# allow all here as a start  
[/]  
* = rw  
  
## http://pd331s/svn/projSandbox/ # alle durch /-Erbe (s.o.)  
  
## http://pd331s/svn/weFramework/  
[weFramework:/]  
* =  
@frameWreader = r  
@frameWcommitter = rw  
  
[weWebSite:/]  
* =  
@frameWcommitter = rw  
  
-----      authz.conf      (Ende) -----
```

A3 XSL-Transformator für Verzeichnislisten

Die beiden folgenden style-sheets sorgen für eine gefälligere Darstellung von Repository-Listings. Ihre absolute Lageangaben, also hier /conf/svnindex.xsl bzw. /conf/svnindex.css beziehen sich auf Apaches "document root", hier D:\www\serv-intra\, sprich auf die Wurzel des statischen Webauftritts des SVN-Apache-Servers.

```
----- D:\www\serv-intra\conf\svnindex.xsl -----
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE xsl:stylesheet [ <!ENTITY nbsp "&#160;"> ]>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  version="2.0">

<!-- XML transformation style sheet
  for displaying the Subversion directory listing generated by
  mod_dav_svn if "SVNIndexXSLT" directive is used.
  31.10.2008: A. Weinert
-->

<xsl:template match="*" />

<xsl:template match="svn">
  <html><head>
    <title>
      <xsl:if test="string-length(index/@base) != 0">
        <xsl:value-of select="index/@base"/>
        <xsl:text>: </xsl:text>
      </xsl:if>
      <xsl:value-of select="index/@path"/>
      <xsl:text> &#160; &#x2014; &#160;</xsl:text>
    </title>
  <link rel="stylesheet" type="text/css" href="/conf/svnindex.css"/>
  <script type="text/javascript">

function getLoadDate() {
  var now = new Date();
  var Std = now.getHours();
  var Min = now.getMinutes();
  var StdF = ((Std &lt; 10) ? "0" + Std : Std);
  var MinF = ((Min &lt; 10) ? "0" + Min : Min);
  var month = now.getMonth() + 1;
  var MonF = ((month &lt; 10) ? "0" + month : month);
  return now.getDate() + "." + MonF + "."
}
```

```

        + (now.getYear() + 1900) + " " + StdF + ":" + MinF ;
    }

var nowDate = getLoadDate();

function aendere() {
    document.getElementById("whereDate").firstChild.nodeValue =
        "generated at " + nowDate;
}

</script>
</head>
<body>
    <div class="svn">
        <xsl:attribute name="onMouseOver">
            javascript:aendere()</xsl:attribute>
        <xsl:apply-templates/>
    </div>
    <div class="footer">
        <xsl:attribute name="onMouseOver">
            javascript:aendere()</xsl:attribute>
        <xsl:text>Powered by </xsl:text>
        <xsl:element name="a">
            <xsl:attribute name="href">
                http://www.meva-lab.de
            </xsl:attribute><i>MEVA</i>-Lab</xsl:element><xsl:text>'s&#160;
                </xsl:text>
            <xsl:element name="a">
                <xsl:attribute name="href">
                    <xsl:value-of select="@href"/>
                </xsl:attribute>
                <xsl:text>Subversion</xsl:text>
            </xsl:element>
            <xsl:text> &#160;</xsl:text>
            <xsl:value-of select="@version"/>
            <xsl:text> &#160;</xsl:text>
            <xsl:element name="span">
                <xsl:attribute name="id">whereDate</xsl:attribute>
                <xsl:attribute name="onMouseOver">
                    javascript:aendere()</xsl:attribute>
                generated at: tt.mm.jjjj hh:mm
            </xsl:element>
        </div>

```

```

    </body></html>
</xsl:template>

<xsl:template match="index">
  <div class="rev">
    <xsl:value-of select="@name"/>
    <xsl:if test="@base">
      <xsl:if test="@name">
        <xsl:text>: &#xA0; </xsl:text>
      </xsl:if>
      <xsl:value-of select="@base" />
    </xsl:if>

    <xsl:if test="@rev > 0">
      <xsl:if test="@base | @name">
        <xsl:text> &#160; &#x2014; &#160; </xsl:text>
      </xsl:if>
      <xsl:text>Revision </xsl:text>
      <xsl:value-of select="@rev"/>
    </xsl:if>
    <xsl:if test="@rev = 0">
      <xsl:element name="a">
        <xsl:attribute name="href">
          http://www.meva-lab.de
        </xsl:attribute>
        <i>MEVA</i>-Lab
      </xsl:element>
      <xsl:text> &#160; &#x2014; &#160; </xsl:text>
      <xsl:element name="a">
        <xsl:attribute name="href">
          <xsl:value-of select="@href"/>
        </xsl:attribute>
        <xsl:text>Subversion</xsl:text>
      </xsl:element>
      <xsl:text> </xsl:text>
    </xsl:if>
  </div>

  <div class="path">
    <xsl:element name="a">
      <xsl:attribute name="href">..</xsl:attribute>
      <xsl:attribute name="title">
        one diretcory up (if possible)</xsl:attribute>
      <xsl:value-of select="@path"/>
    </xsl:element>
  </div>

```

```

    </xsl:element>
</div>

<div class="updir">
  <xsl:text>.. </xsl:text>
  <xsl:choose>
    <xsl:when test="updir">
      <xsl:element name="a">
        <xsl:attribute name="href">..</xsl:attribute>
        <xsl:attribute name="title">
          to the parent directoy</xsl:attribute>
        <xsl:text>[one director</xsl:text>
      </xsl:element>
      <xsl:text>y up]</xsl:text>
    </xsl:when>
    <xsl:otherwise>
      <xsl:element name="a">
        <xsl:attribute name="href">..</xsl:attribute>
        <xsl:attribute name="title">
          to svn / server (if allowed)</xsl:attribute>
        <xsl:text>[parent director</xsl:text>
      </xsl:element>
      <xsl:text>y] (may be forbidden)</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
</div>

<xsl:apply-templates select="dir"/>
<xsl:apply-templates select="file"/>
</xsl:template>

<xsl:template match="dir">
  <div class="dir">
    <xsl:element name="a">
      <xsl:attribute name="href">
        <xsl:value-of select="@href"/>
      </xsl:attribute>
      <xsl:value-of select="@name"/>
      <xsl:text>/</xsl:text>
    </xsl:element>
  </div>
</xsl:template>

<xsl:template match="file">

```

```

<div class="file">
  <xsl:element name="a">
    <xsl:attribute name="href">
      <xsl:value-of select="@href"/>
    </xsl:attribute>
    <xsl:value-of select="@name"/>
  </xsl:element>
</div>
</xsl:template>
</xsl:stylesheet>

```

----- D:\www\serv-intra\conf\svnindex.xml ----- (Ende) -----

----- D:\www\serv-intra\conf\svnindex.css -----

```

/* style sheet for displaying the Subversion directory listing
   that is generated by mod_dav_svn and "svnindex.xml".
   modif. 1.10.2008: Albrecht Weinert
   MEVA-Lab Subversion (http://PD331S/svn/.....)
*/

```

```

body {
  margin: 0;
  padding: 0;
  font-family:arial, helvetica, sans-serif;
}

a {
  color: navy;
}

.footer {
  margin-top: 3em;
  padding: 0.5em 1em 0.5em;
  border: 1px solid;
  border-width: 1px 0;
  clear: both;
  border-color: rgb(30%,30%,50%) navy rgb(75%,80%,85%) navy;
  background: rgb(88%,90%,92%);
  font-size: 80%;
}

.svn { margin: 2em; }

```

```

.rev {
  margin-right: 3px;
  padding-left: 3px;
  text-align: left;
  font-size: 120%;
}
.rev a {
  text-decoration: none;
  color: blue;
}

.dir a, .file a, .path a {
  text-decoration: none;
  color: black;
}

.path {
  margin: 3px;
  padding: 3px;
  background: #FFCC66;
  font-size: 120%;
}

.updir {
  margin: 3px;
  padding: 3px;
  margin-left: 3em;
  background: #FFEEAA;
}

.file {
  margin: 3px;
  padding: 3px;
  margin-left: 3em;
  background: rgb(95%,95%,95%);
}

.file:hover, .updir:hover {
  margin: 3px;
  padding: 3px;
  margin-left: 3em;
  background: rgb(100%,100%,90%);
}

```

```
.dir {
  margin: 3px;
  padding: 3px;
  margin-left: 3em;
  background: rgb(85%,85%,85%);
}

.dir:hover {
  margin: 3px;
  padding: 3px;
  margin-left: 3em;
  background: rgb(100%,100%,80%);
}
```

---- D:\www\serv-intra\conf\svnindex.css ----- (Ende) -----

A4 XML - Listing und XSL-Anzeige Ergebnis

Zum Verständnis der Transformation des letzten Abschnittes hier ein (selbsterklärendes) Beispiel eine XML-Repository-Listings sowie das dann mit der client- (browser-) seitigen Transformation angezeigte Ergebnis (Bild 10).

---- XML Repository-Listing (erklärt den Document Type) -----

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="/conf/svnindex.xsl"?>
<!DOCTYPE svn [
  <!ELEMENT svn (index)>
  <!ATTLIST svn version CDATA #REQUIRED
                href CDATA #REQUIRED>
  <!ELEMENT index (updir?, (file | dir)*)>
  <!ATTLIST index name CDATA #IMPLIED
                path CDATA #IMPLIED
                rev CDATA #IMPLIED
                base CDATA #IMPLIED>
  <!ELEMENT updir EMPTY>
  <!ELEMENT file EMPTY>
  <!ATTLIST file name CDATA #REQUIRED
                href CDATA #REQUIRED>
  <!ELEMENT dir EMPTY>
  <!ATTLIST dir name CDATA #REQUIRED
                href CDATA #REQUIRED>
]>
<svn version="1.5.3 (r33570)"
      href="http://subversion.tigris.org/">
  <index rev="1" path="/" base="weFramework">
    <dir name="branches" href="branches/" />
    <dir name="tags" href="tags/" />
    <dir name="trunk" href="trunk/" />
  </index>
</svn>
```

---- XML Repository-Listing (Ende) -----

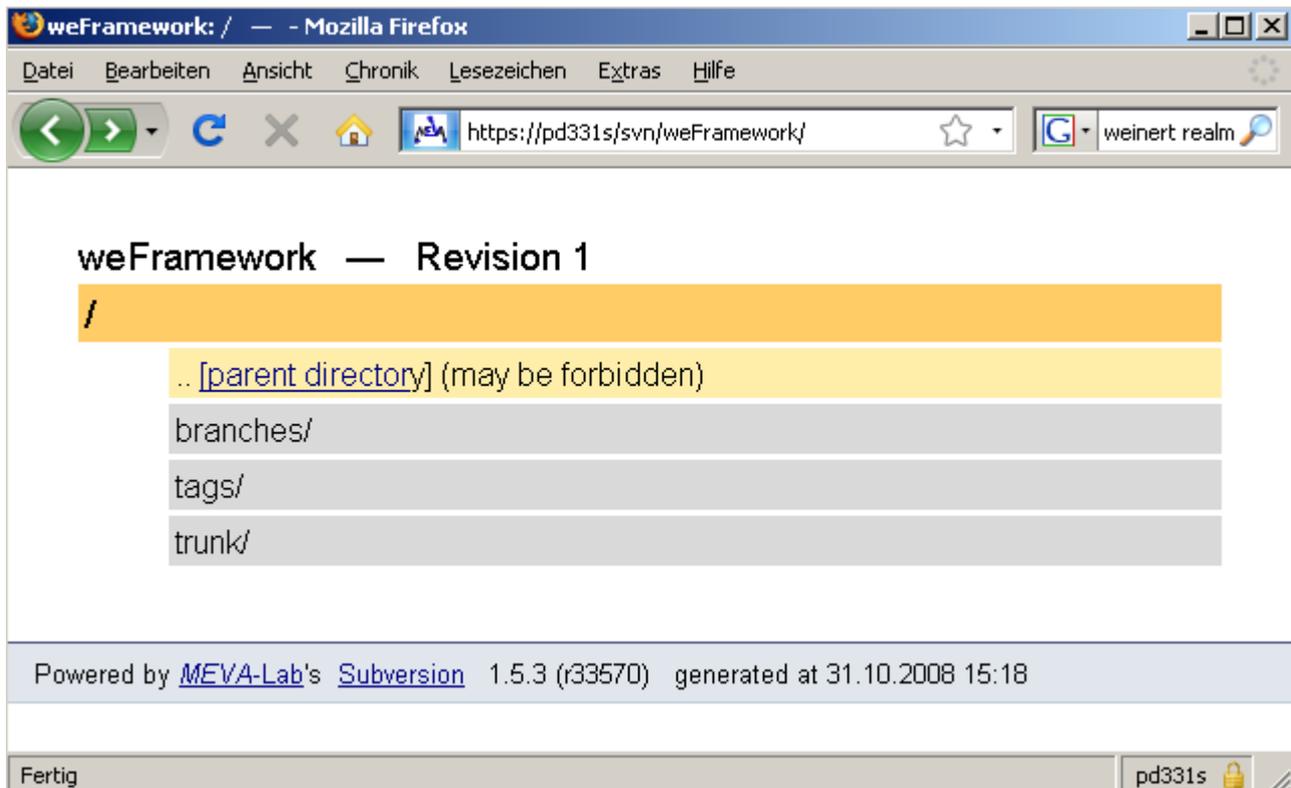


Bild 10: Transformierte Repository-Anzeige.

A5 Zur Ablehnung ohne Authentifizierungsabfrage

Zitat von <http://tortoisesvn.tigris.org/faq.html#four01>:

The reason for this is the automatic authentication with SSPI which is activated in version 1.4.x. That means TortoiseSVN now tries to authenticate automatically with the credentials of the user logged on to the Windows domain controller.

If you have set up your server to authenticate with SSPI against a domain controller, and the domain controller does not have the user account GUEST enabled, you should be fine. But if the user account GUEST is active, then all authentication succeeds with that user - and you usually won't give the user GUEST access to your repository. That's why the *authentication* succeeds, but the *authorization* fails.

Another reason why it can fail is if you have set up different accounts for the repository access than you use for logging in to your workstations (although then I wonder why you are using SSPI authentication in the first place).

To solve this issue you have the following options:

1. disable the GUEST account on the domain controller
2. use the same accounts for your workstations and access to the repository
3. disable SSPI authentication for the repository
4. Check the case of the usernames. Changing the usernames in the access files to lowercase also might solve this problem

(Zitat Ende)

Wenn dies der Fall sein sollte, hat man mit der betreffenden Version mit Nicht-Domain-Rechner und SVN mit SSPI-Authentifizierung gegen AD verloren. Dies wäre dann ein Bug; Mozilla zeigt ja, dass es in genau der Kombination prima geht.

Anmerkungen zu den Lösungsvorschlägen aus obigem Zitat:

1. ist in jeder halbwegs professionell geführten Domain eh der Fall. Außerdem wird nicht erklärt, wer denn nun das Konto `Gast` ins Spiel bringen sollte.
2. läuft darauf hinaus, den betreffenden Rechner in die Domain aufzunehmen. Das kommt häufig nicht in Frage, und es ist natürlich keine Lösung des eigentlichen Problems.
3. kann hier nicht ernst gemeint sein. Man macht AD-Authentifizierung, weil man den Verwaltungsaufwand ja eh hat. Wenn natürlich gemeint ist "Ohne jede Authentifizierung hat man damit auch keine Probleme", so stimmt das immer.
4. wie 1. Ist schon der Fall, kommt also als Ursache / Lösung nicht in Frage.

A6 Abkürzungen

ACL	access control list (Liste mit Zugriffsrechten auf ein Objekt)
AD	Active Directory (Microsofts Interpretation von LDAP)
AJAX	Asynchronous JavaScript + XML
API	Application Programme Interface
BuB	Bedienen und Beobachten (von Prozessen)
C/S	Client-Server
CA	Certification authority
CVS	Concurrent Versioning System
FAQ	Frequently Asked Questions (Hilfetexte in Frage-Antwort-Form)
FB	Fachbereich; insbesondere ...
FB3	Fachbereich Elektrotechnik und Informatik der Hochschule Bochum
FSFS	fast secure file system (neues Datenbanksystem von SVN)
GSS	Generic Security Service
GUID	Globally Unique Identifier
GWT	Google Webtoolkit, AJAX mit nur Java
HTML	Hypertext Markup Language [RFC 1866]
HTTP	Hypertext Transfer Protokoll. Internet-Protokoll zur Übertragung von Seiten.
HTTPS	HTTP über SSL. Abgesicherte Übertragung.

HW	Hardware
IIOp	Internet Inter-ORB Protocol
IP	Internet Protocol
J2EE	Java 2 Enterprise Edition
J2ME	Java 2 Micro Edition
J2SE	Java 2 Standard Edition
JAAS	Java Authentication and Authorization Service
JAF	JavaBeans Activation Framework
JAR	Java Archive. (.zip + Semantik)
JAXP	Java API for XML Parsing
JCA	Java Cryptography Architecture (der Java Security API)
JCE	Java Cryptography Extensions (zur JCA, Exportrestriktion wegen DAS, DES)
JDBC	Java Database Connectivity (Java Datenbankanschluss)
JDC	Java Developer Connection (Ein WWW-Service)
JDK	Java Development Kit; der Werkzeugsatz für die Entwicklung mit Java
JEB	Enterprise JavaBeans (ungleich JavaBeans)
JMX	Java Management Extensions
JNDI	Java Naming and Directory services Interface
JNI	Java Native Interface
JRE	Java Runtime Environment; JDK-Subset ohne Entwicklungswerkzeuge.
JSDK	Java Servlet Development Kit
JSF	Java Server Faces
JSP	Java Server Pages
JSSE	Java Secure Socket Extension (seit JDK1.4.x integriert)
JSTL	JavaServer Pages Standard Tag Library
JVM	Java virtual machine; der eigens für Java erfundene Prozessor. Er wird im Allgemeinen auf dem jeweiligen Zielsystem emuliert.
LAN	Local area network; Datennetz für mittlere Entfernungen
LDAP	Lightweight Directory Access Protocol
LGPL	Lesser GNU Public License
MBean	Managed Bean (JMX)
MEVA	Labor für Medien und verteilte Anwendungen

MS	Microsoft
NT	Betriebssystem Windows NT (MS)
OMG	Object Management Group
OS	Operating System
PAM	Pluggable Authentication Module
PC	Personal Computer
R&D	Research and Development
RAID	Redundant Array of inexpensive Disks
RDF	Resource Description Framework (W3C)
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SAAJ	SOAP with Attachments API for Java
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SQL	Structured query language, Datenbankbearbeitungssprache
SSL	Secure Socket Layer. Protokollschicht zu Absicherung.
SSO	Single Sign on; Authentifizierung vieler (n) Anwendungen gegen eine (1) "security realm".
SSPI	Security Support Provider Interface
SVN	Subversion
TCP	Transmission Control Protocol
TM	Trade Mark (Warenzeichen)
UML	Unified Modelling Language
URI	Uniform Resource Locator
W2K	Betriebssystem Windows 2000 (MS)
W2K3	Betriebssystem Windows Server 2003 (MS)
W3	Amerikanische Kurzform für WWW
W3C	World Wide Web Consortium
WebDAV	Web-based Distributed Authoring and Versioning
WS	Workstation
WSDL	Web Services Description Language
XML	eXtensible Markup Language

A7 Literatur

- [1] Ed Ort and Mark Basler, AJAX Design Strategies, SUN 2006
<http://java.sun.com/developer/technicalArticles/J2EE/AJAX/.../design-strategies.pdf>
- [2] Brett McLaughlin, Mastering Ajax, Part 1..4, IBM, 2005
<http://www-128.ibm.com/developerworks/web/library/wa-ajaxintro.html>
- [3] Albrecht Weinert, Zur Installation des JDK (Java Development Kit)
<http://a-weinert.de/weinert/pub/java-install.txt>
- [4] Albrecht Weinert, Java — Tipps und Tricks
<http://a-weinert.de/weinert/pub/java-tips.txt>
- [5] Albrecht Weinert, AJAX mit GWT — Tipps und Tricks
<http://a-weinert.de/weinert/pub/gwt-tips.pdf>
- [6] Albrecht Weinert, Tipps zu CVS für Windows — cvsNT
<http://a-weinert.de/weinert/pub/cvsnt-tipp.txt>
- [7] Google, Web-Toolkit, online-Dokumentation (nicht am Stück verfügbar)
<http://code.google.com/webtoolkit/documentation/>.
- [8] Albrecht Weinert, Tipps zu JMX mit SSL
<http://a-weinert.de/weinert/pub/jmx-ssl-tips.pdf>
- [9] Albrecht Weinert, Windows 2003 Domain Migration von NT4 mit Fremd-DNS
<http://www.a-weinert.de/weinert/pub/w2k3domain.pdf>
- [10] Albrecht Weinert, Windows Server 2003 — Domain FB3-MEVA Schulungsräume und Infrastruktur — Renovierung 2007
<http://www.a-weinert.de/weinert/pub/fb3-meva-domain2007.pdf>
- [11] Albrecht Weinert, Tipps zu Tomcat (5.x für Windows) ersetzt durch [13] ([13] stattdessen für Tomcat >= 6) <http://a-weinert.de/weinert/pub/tomcat-tips.pdf>
- [12] Albrecht Weinert, Windows Server 2003 — Domain FB3-MEVA Workstations und Server — Renovierung 2007
<http://www.a-weinert.de/weinert/pub/fb3-meva-workst2007.pdf>
- [13] Albrecht Weinert, Tomcat — mit Windows und Active Directory (ersetzt [11] als Nachfolger) <http://www.a-weinert.de/weinert/pub/tomcat-win-ad.pdf>
- [14] Albrecht Weinert, Tipps zu MySQL (mit Java, für Windows) (2006) <http://a-weinert.de/weinert/pub/mysqjawi-tipp.txt>
- [15] Albrecht Weinert, Tipps zu Subversion (2006; ersetzt durch [16])
<http://www.a-weinert.de/weinert/pub/subversion-install-tipp.txt>

<16> Albrecht Weinert, Subversion — mit Windows und Active Directory (2008)
(ersetzt [15]) <http://www.a-weinert.de/weinert/pub/subversion-win-de.pdf>

[17] Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato,
Version Control with Subversion; For Subversion 1.5;
als C:\Programme\Subversion\svn-book.pdf bei CollabNet-SVN mitinstalliert oder im Web

Hinweis: Aus Dateien „.../docu/*.txt“ könnten inzwischen teilweise
„.../weinert/pub/*.pdf“ geworden sein.