



Subversion

Quellcodeverwaltung a la Opensource

Matthias Fischer

Technical Expert .NET
Tieto, JOSI / Microsoft
Matthias.Fischer@tieto.com

Agenda

- Grundlagen
 - Installation
 - Administration / Konfiguration
- Verwendung
 - Struktur für Projekte
 - Lokal / Remote
- Advanced
 - Keywords
 - Hooks
 - Backup
 - Branches ,Tags und Merging



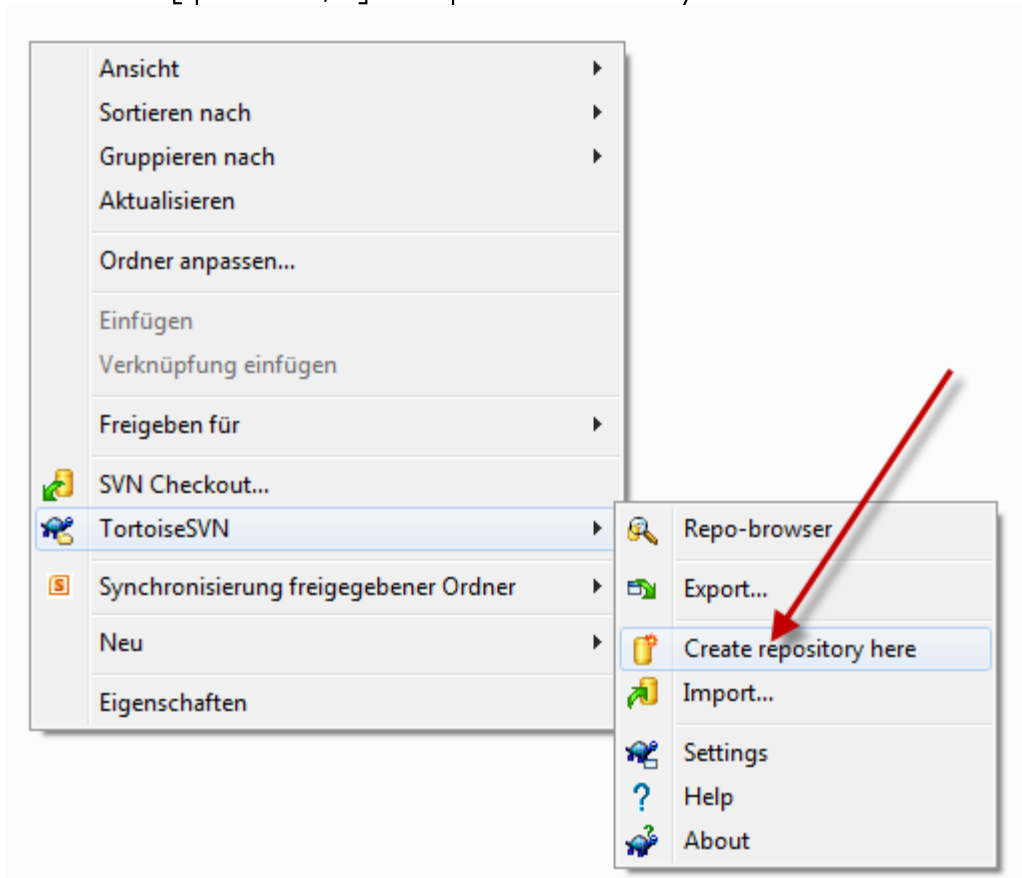
Übersicht

- Clients
 - Kommandozeile ([svn.exe](#), [svnadmin.exe](#), [SilkSvn...](#))
 - Windows GUI ([tortoisesvn](#), [rapidsvn](#), ...)
 - Andere ([AnkhSvn](#), [VisualSvn](#) (\$),...)
- Server
 - Standalone
 - Leichtgewicht, mit verbindungsorientiertem Protokoll, TCP/IP
 - tunnel mode (z.B. SSH)
 - Kommunizieren durch `svn://` oder `svn+ssh://` Schemata
 - Apache modul
 - WebDAV-basiert
 - Skalierbar
 - Sicherer



Erstellen eines Repositories

- Kommandozeile
 - `svnadmin create <[path/]repositoryname>`
- TortoiseSVN



Aufbau Repository

- **conf**
 - Enthält die Konfigurationsdateien.
- **dav**
 - Enthält ggf. Dateien welche vom apache-dav Modul benötigt werden.
- **db**
 - Enthält die Berkeley DB oder das Subversion eigenen Filesystem (welches die versionierten Verzeichnisse und Dateien enthält).
- **format**
 - Enthält einfache Integer-Werte welche die Version des Repositories wieder geben.
- **hooks**
 - Enthält Skripte für diverse Aktionen .
- **locks**
 - wird für Semaphoren verwendet



Basis-Konfiguration

- **svnserve.conf**

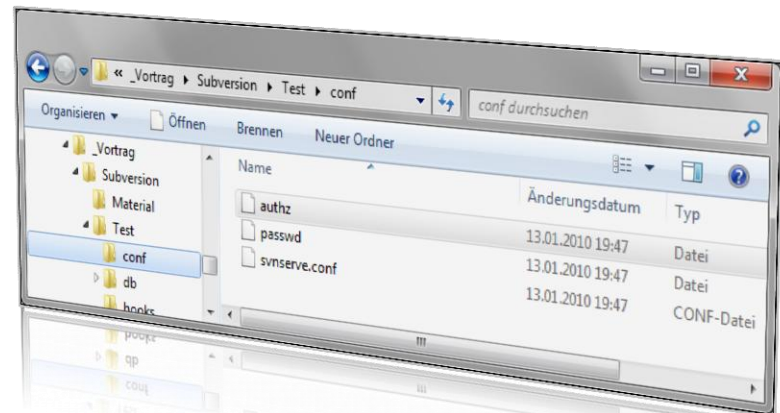
```
[general]
anon-access = read
auth-access = write
password-db = passwd
authz-db = authz
realm = Test Repository .NET
UGBerlin
```

- **passwd**

```
[users]
fischmat = geheim
schwewer = xxx
```

- **authz**

```
[/foo/bar]
fischmat = rw
schwewer = r
* =
```



Basis-Konfiguration

- authz

```
aliases]
```

```
joe = /C=XZ/ST=Dessert/L=Snake City/O=Snake Oil, ↵  
Ltd./OU=Research Institute/CN=Joe Average
```

```
[groups]
```

```
harry_and_sally = harry,sally
```

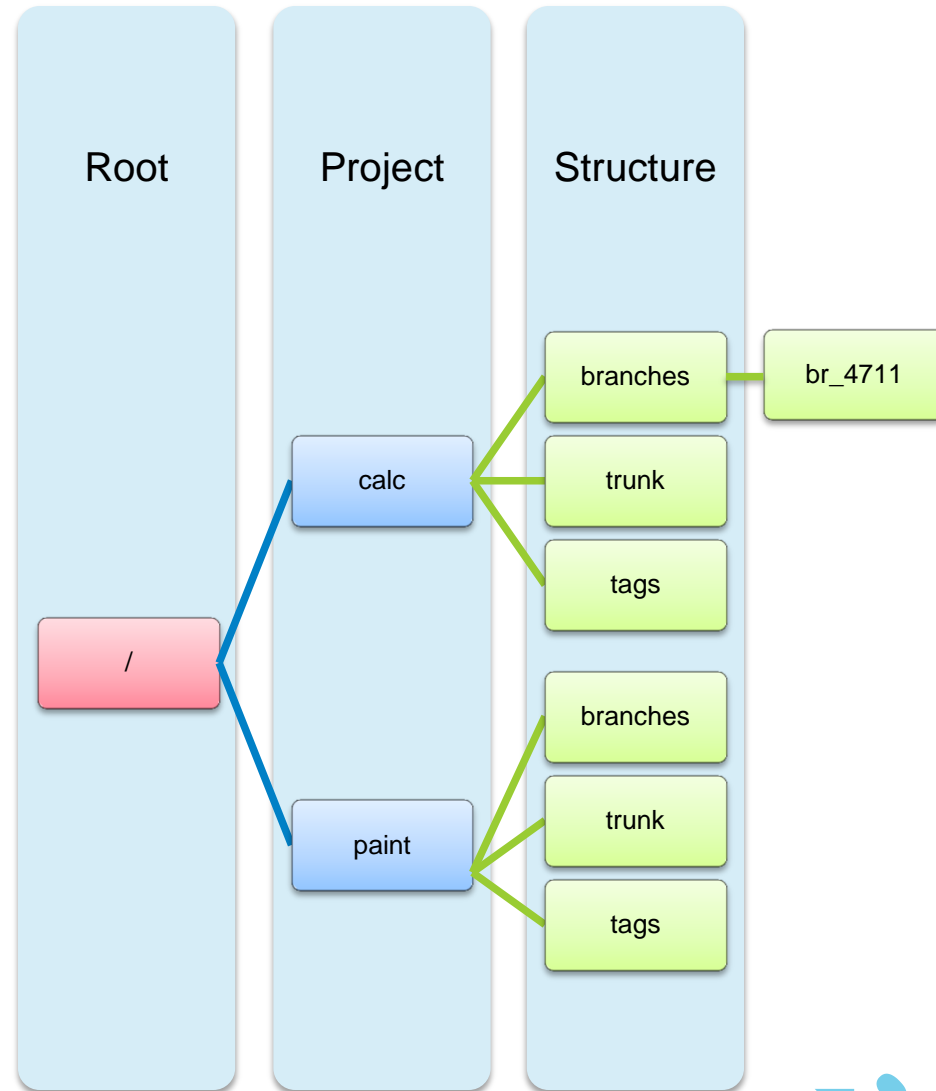
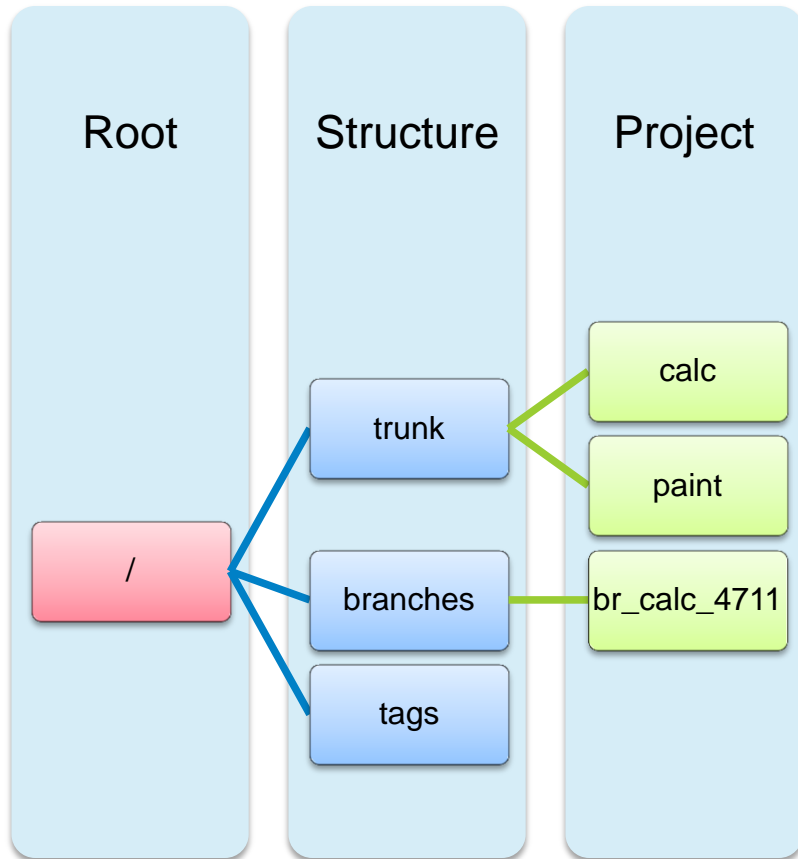
```
harry_sally_and_joe = harry,sally,&joe
```

Demo

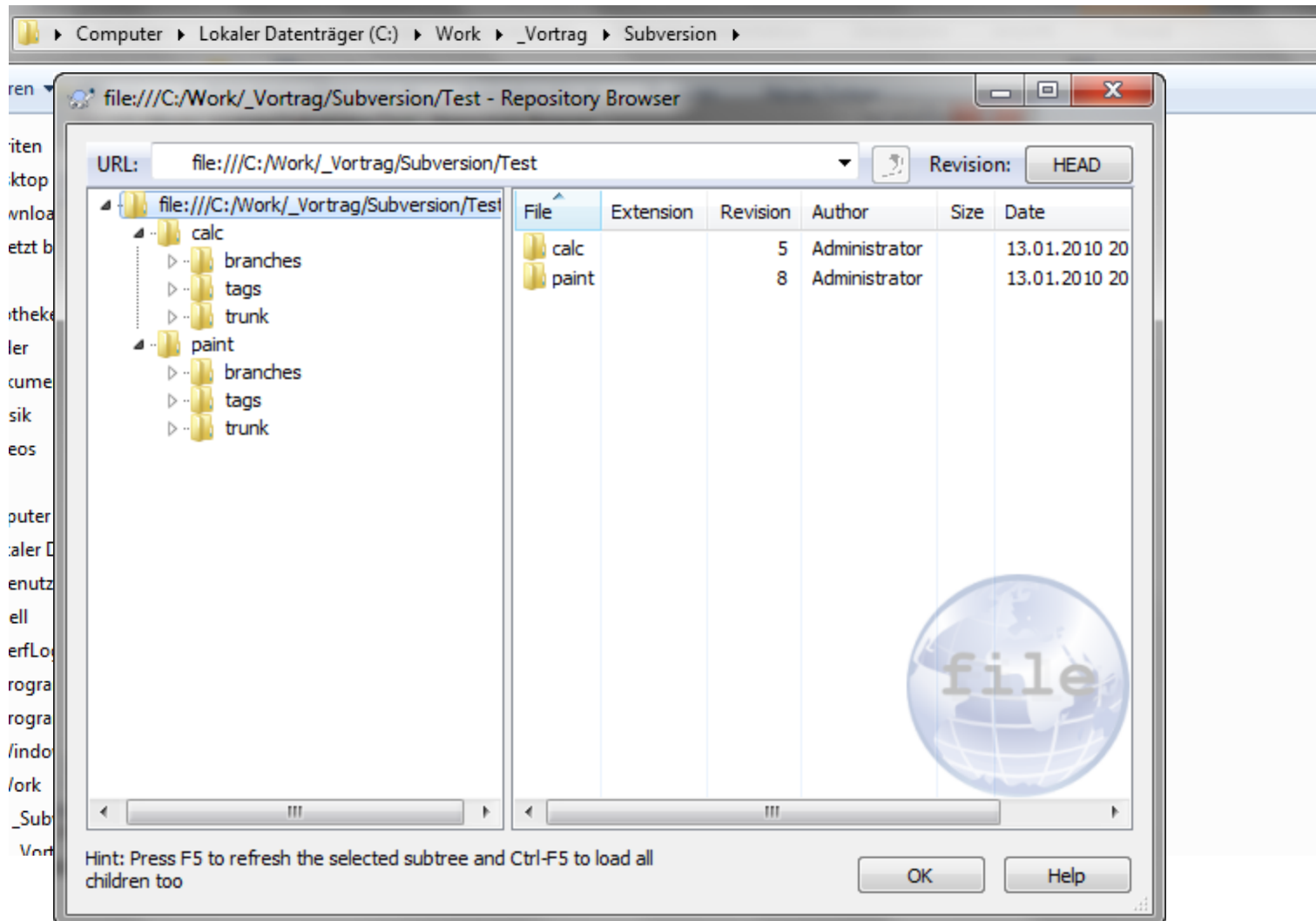
- Anlegen eines Repositories
 - Mit Kommandozeile
 - Mit TortoiseSVN
- Konfiguration der Benutzer



Struktur



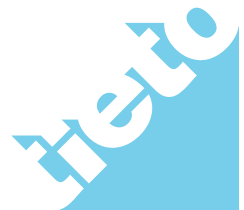
Anlegen der Basis Struktur



Demo

- Struktur aufbauen
 - TortoiseSVN (Repository-Browser)
 - Kommandozeile

```
set svnroot = "cd c:\work\_Subversion"  
cd %svnroot%  
svnadmin create ugberlin  
cd c:\work  
svn co file:///%svnroot%/ugberlin  
cd ugberlin  
md paint calc  
cd paint  
md trunk tags branches  
cd ..\calc  
md trunk tags branches  
cd ..  
svn add *  
svn ci -m "init"
```



Verwendung + Administration

- Direkter Zugriff auf das Repository
 - <file:///path/to/repository>
- svnserve – TCP basierter Server
 - `svnserve -d -r <path/to/repos> --listen-host <host oder IP>`

Automatischer Start als Windows-Dienst

```
sc create svnserve ↵  
    binpath="\"c:\Program Files\SilkSvn\bin\svnserve.exe\" ↵  
    --service --root c:\work\_Subversion" ↵  
    displayname="Subversion" ↵  
    depend=tcPIP ↵  
    start=auto
```



Sichern & Wiederherstellen

#Backups

```
svnadmin dump ugberlin > ugberlin.dumpfile
```

```
svnadmin dump ugberlin -r 23 > rev-23.dumpfile
```

```
svnadmin dump ugberlin -r 100:200 > revs-100-200.dumpfile
```

#Restore

```
svnadmin load newrepos < ugberlin.dumpfile
```

#Merge

```
svnadmin create newrepos
```

```
svn mkdir -m "Initial project roots" \ file:///./calc \ file:///./paint
```

```
svnadmin load newrepos --parent-dir calc < calc.dumpfile
```

```
svnadmin load newrepos --parent-dir paint < paint.dumpfile
```

#Splitt

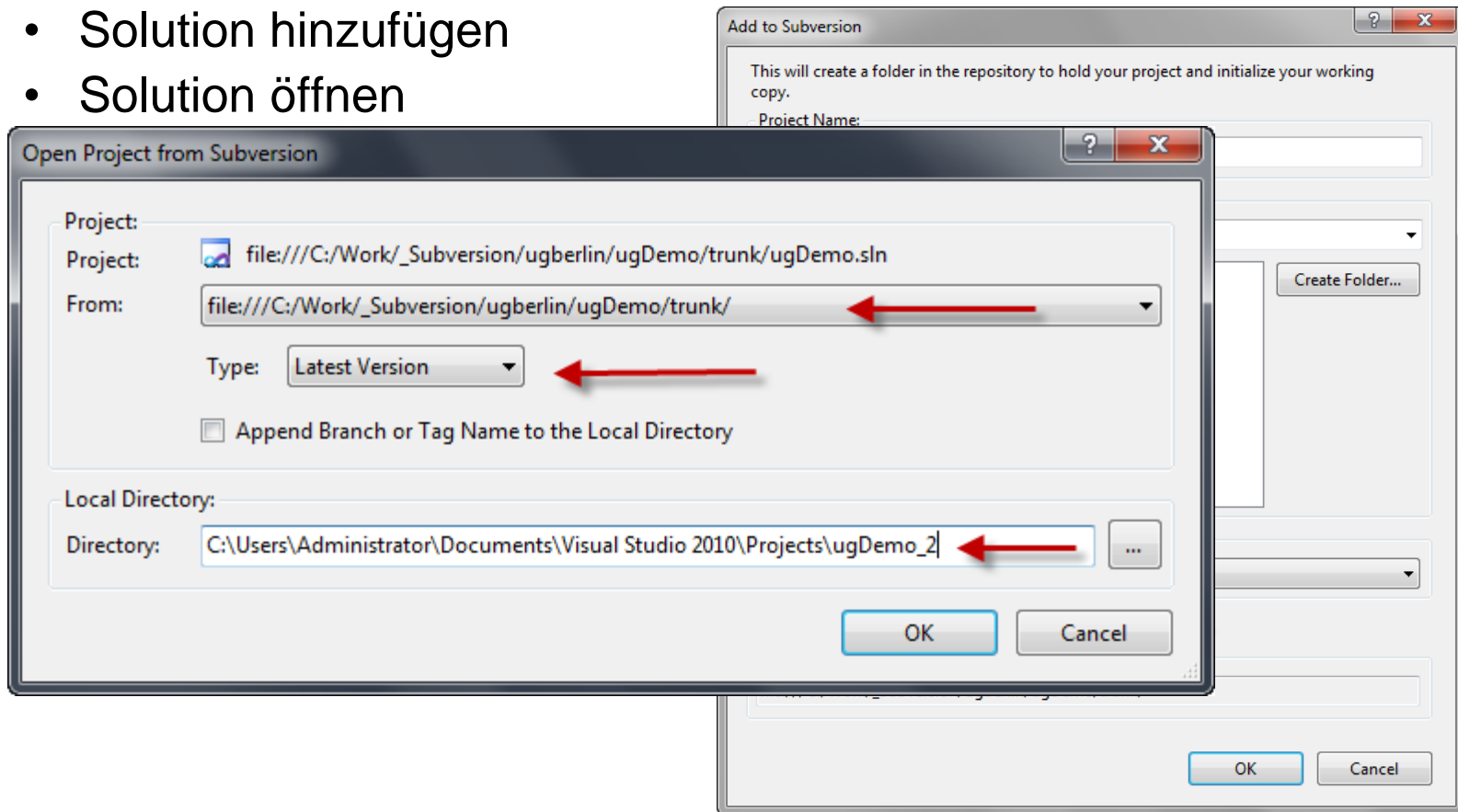
```
svndumpfilter include calc < repos.dumpfile > calc.dumpfile
```

```
svndumpfilter include paint < repos.dumpfile > paint.dumpfile
```



3rd Party Tools

- [AnkhSVN](#) freies Plugin für Visual Studio 2005/2008/2010
 - Solution hinzufügen
 - Solution öffnen



Demo

- Neues Projekt einfügen
- Projekt aus Subversion öffnen

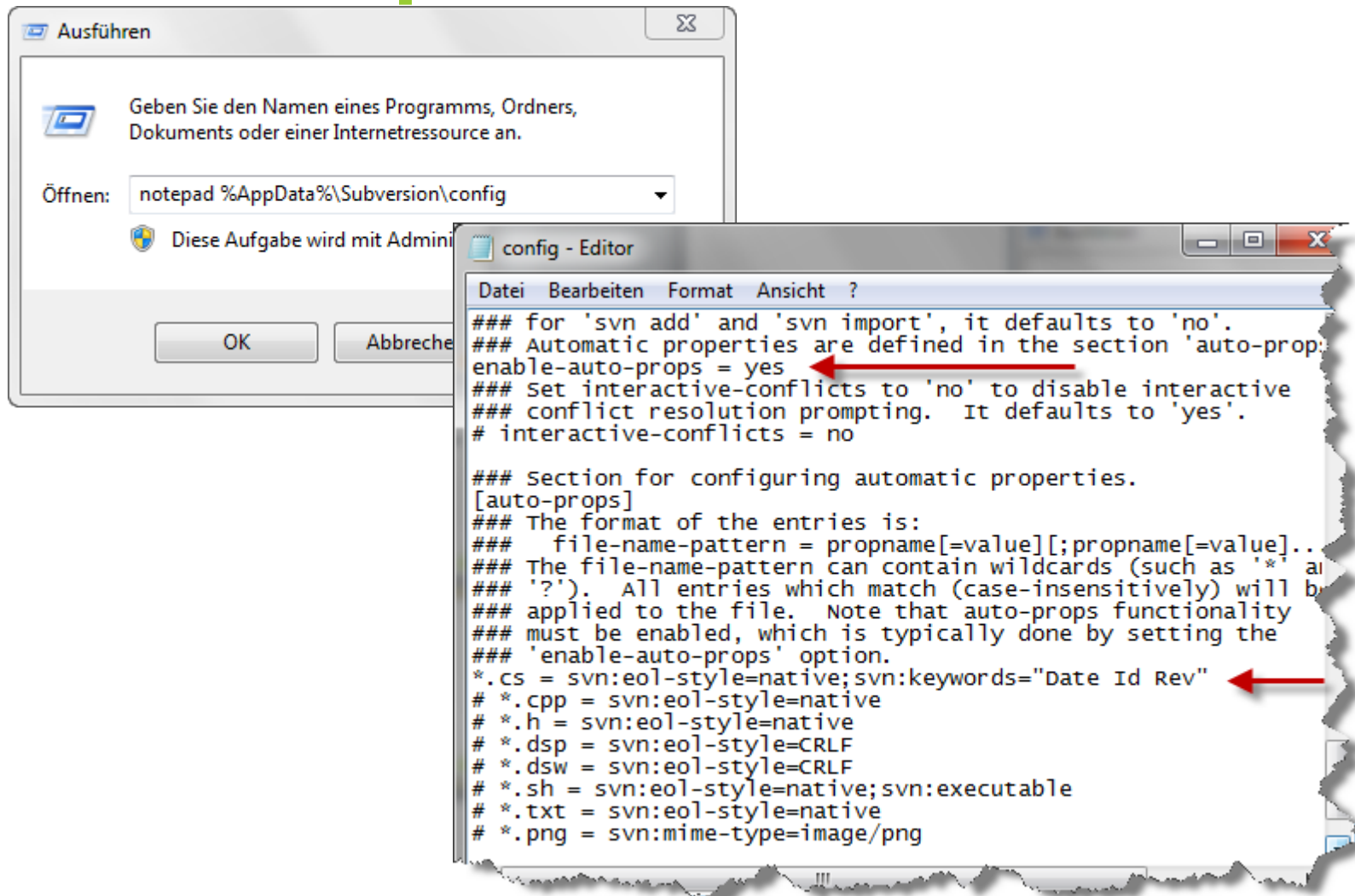


Schlüsselwörter

- Durch setzen der Eigenschaft **svn:keywords** können diverse Schlüsselwörter ersetzt werden
 - `$Date$`
 - `$Revision$` (`Rev`)
 - `$Author$`
 - `$HeadURL$`
 - `Id`
 - `$LastChangesDate$`
- **Beispiel:**
 - `svn propset svn:keywords "Date Author" calc.cs`



Auto-Properties



The image shows two overlapping windows. The background window is a Windows 'Ausführen' (Run) dialog box with the text 'Geben Sie den Namen eines Programms, Ordners, Dokuments oder einer Internetressource an.' and the input field containing 'notepad %AppData%\Subversion\config'. The foreground window is a Notepad editor titled 'config - Editor' showing the contents of the SVN configuration file. The text in the editor is as follows:

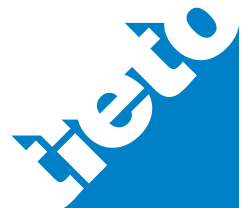
```
### for 'svn add' and 'svn import', it defaults to 'no'.
### Automatic properties are defined in the section 'auto-prop:
enable-auto-props = yes
### Set interactive-conflicts to 'no' to disable interactive
### conflict resolution prompting. It defaults to 'yes'.
# interactive-conflicts = no

### Section for configuring automatic properties.
[auto-props]
### The format of the entries is:
###   file-name-pattern = proptype[=value][;proptype[=value]..
### The file-name-pattern can contain wildcards (such as '*' and
### '?'). All entries which match (case-insensitively) will be
### applied to the file. Note that auto-props functionality
### must be enabled, which is typically done by setting the
### 'enable-auto-props' option.
*.cs = svn:eol-style=native;svn:keywords="Date Id Rev"
# *.cpp = svn:eol-style=native
# *.h = svn:eol-style=native
# *.dsp = svn:eol-style=CRLF
# *.dsw = svn:eol-style=CRLF
# *.sh = svn:eol-style=native;svn:executable
# *.txt = svn:eol-style=native
# *.png = svn:mime-type=image/png
```

Two red arrows point to the lines 'enable-auto-props = yes' and '*.cs = svn:eol-style=native;svn:keywords="Date Id Rev"'. The Notepad window has a jagged, torn-paper edge effect on its right side.

Hooks

- [start-commit](#)
- [pre-commit](#)
- [post-commit](#)
- [pre-revprop-change](#)
- [post-revprop-change](#)
- [pre-lock](#)
- [post-lock](#)
- [pre-unlock](#)
- [post-unlock](#)



Beispiel

Beispiel Programm in C# für **Start_Commit**

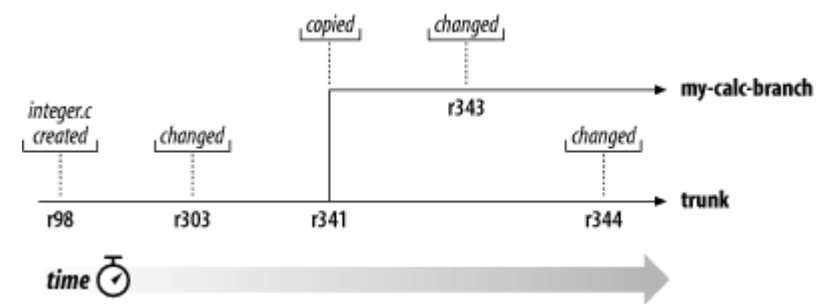
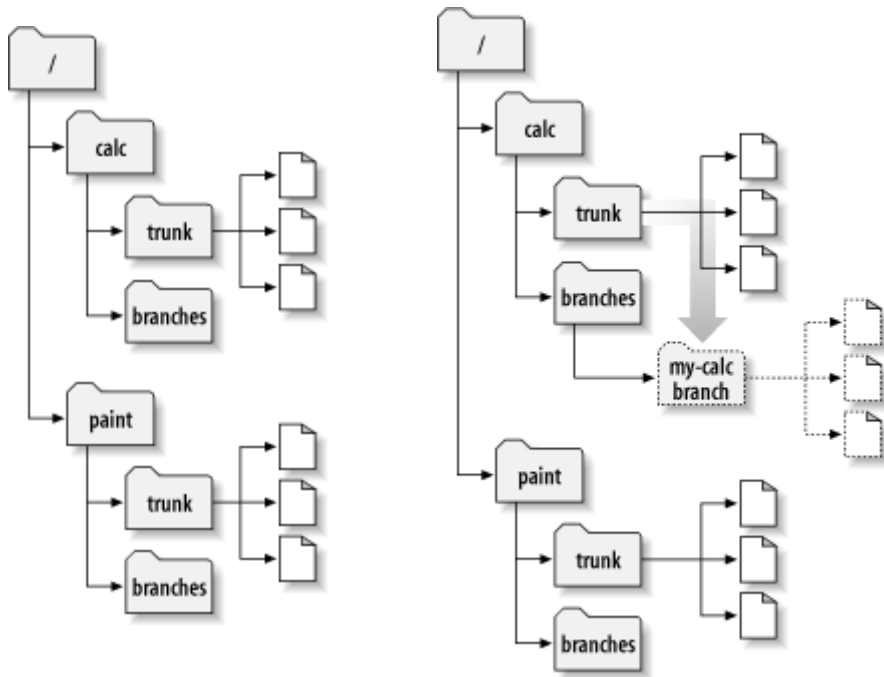
- Hinweise: unter Windows müssen die Skripte eine ausführbare Endung haben (exe, bat, cmd)
- Unter Linux ist keine Endung erforderlich
- Hook-Skripte werden sowohl vom **svnserve** als auch von den **Clients**, welche direkt auf das Repository zugreifen beachtet

```
class Program {
    static int Main(string[] args) {
        var user = (args.Length > 1) ? args[1] : "";
        if (user == "mfi") {
            return 0;
        } else {
            Console.Error.WriteLine("Nur der Benutzer MFI darf synchronisieren !");
            return 1;
        }
    }
}
```



Branching / Tagging

- `svn copy trunk branches/my-calc-branch`
- `svn ci -m "Creating a private branch of /calc/trunk."`



Merging

- `svn diff -c 344 svn://repos/calc/trunk my-calc-branch`
- `svn merge -c 344 svn://repos/calc/trunk my-calc-branch`
- `svn merge svn://repos/branch1@150 svn://repos/branch2@212 my-working-copy`



Demo

- Erzeugen eines Tags
- Erzeugen eines Branches
- Merge des Branches zurück auf die Hauptline



Thank you

Subversion, Quellcodeverwaltung a la Open Source

Matthias Fischer

Technical Expert .NET

Tieto, JOSI / Microsoft

Matthias.Fischer@tieto.com