

# 3 Anhang

## SOFTWAREVERSIONSVERWALTUNG MIT SUBVERSION

### 3.4 Einführung

### 3.5 Voraussetzungen zur Installation

### 3.6 Installation

Client-/Server-Komponente

Grafische Steuerungskomponente des Clients

### 3.7 Verwendung

Konfiguration

Auschecken eines Repositories

Nutzen der Objekte

Zurückschreiben der Objekte

Aktualisierung lokaler Objekte

Erstbefüllen eines Repositories

Empfohlene Ordnungsstruktur eines Repositories

### 3.8 Konfliktauflösung

Ergebnis der Konfliktauflösung

### 3.9 Aufsetzen eines Subversion-Servers

Repository erstellen

Zugriff realisieren

Zugriffsberechtigung

### 3.10 Konfiguration neuerer Versionen von TortoiseSVN

Konfiguration des TFH-Netzwerkzugangs

Dieses Manuskript steht allen Teilnehmern der Lehrveranstaltung *Qualitätsmanagement (QM)* an der *Technischen Fachhochschule Berlin* als unterrichtsbegleitendes Lehrmaterial frei zur Verfügung. Die Nutzung durch andere Personen oder zu anderen Zwecken bedarf zur Vermeidung möglicher Verletzungen des deutschen Urheberrechts der vorherigen Inkennzeichnung und Erlaubnis des Autors.

## 3.4 Einführung

---

Das folgende Dokument beschreibt die Installation und den Betrieb des frei verfügbaren Werkzeugs zur Dokumentenversionsverwaltung **Subversion** und dessen grafisch orientierte Steuerungskomponente des Clients **TortoiseSVN** auf einer Windows-Plattform.

## 3.5 Voraussetzungen zur Installation

---

Subversion besteht aus einer **textorientierten Client-/Server-Komponente** und bei Bedarf aus einer grafischen Steuerungskomponente des Clients. Die erstgenannte Komponente ist im Internet unter folgender Adresse zu finden:

<http://subversion.tigris.org>.

Im Downloadbereich kann aus verschiedenen Installationsdateien ausgewählt werden. Empfohlen wird das jüngste als stabil (*stable*) gekennzeichnete Installationspaket

[svn-X.X.X-setup.exe](#),

wobei X.X.X der aktuellen Versionsnummer dieses Pakets entspricht.

Die **zweitgenannte Komponente** ist im Internet unter folgender Adresse zu finden:

<http://tortoisesvn.tigris.org>.

Es handelt sich hierbei um eine **grafisch orientierte Steuerungskomponente** des Subversion-Clients, welche als Plug-In für den Windows Explorer konzipiert wurde.

Im Downloadbereich ist jeweils die jüngste Version für das zugrunde liegende Windows zu verwenden. Zur Auswahl stehen Versionen für **Windows 98/Me** sowie **Windows NT4/W2K/XP**.

Bei Bedarf darf auch das **deutsche LanguagePack** heruntergeladen und installiert werden. Jedoch beziehen sich die in dieser Dokumentation verwendeten Abbildungen und Bezeichnungen nur auf die englische Version.

## 3.6 Installation

---

### Client-/Server-Komponente

Installieren Sie zunächst die textorientierte Client-/Server-Komponente von Subversion. Der vorgeschlagene **Installationspfad** kann unverändert übernommen werden. Die während der Installation angebotenen **Verknüpfungen** können nach Belieben übernommen oder abgelehnt werden.

## Grafische Steuerungskomponente des Clients

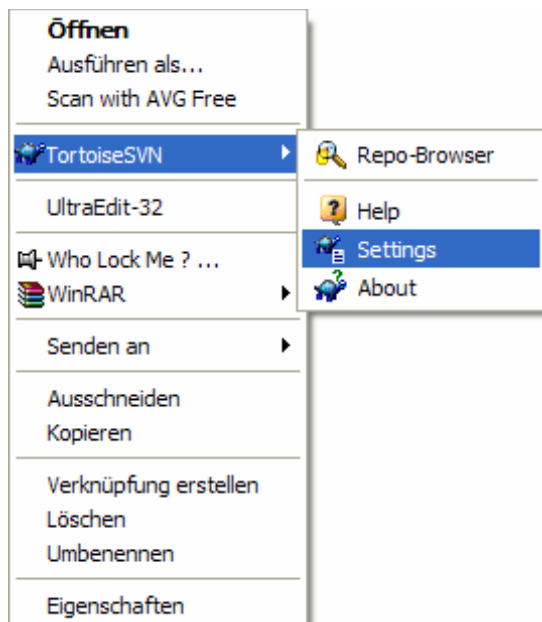
Sie steht als MSI-Paket zur Verfügung, das durch einen Doppelklick gestartet wird. Auch hier kann der vorgeschlagene Installationspfad unverändert übernommen werden. In Abhängigkeit davon, ob nur der augenblicklich windowsseitig angemeldete Benutzer mit der Steuerungskomponente arbeiten soll, oder alle Benutzer, wird in der gleichen Maske entweder Just Me oder Everyone ausgewählt. Der Rest der Installation ist selbsterklärend.

## 3.7 Verwendung

Nun wird nur noch mit TortoiseSVN gearbeitet, der grafischen Steuerungskomponente des Subversion-Clients, die ihrerseits die textorientierten Subversion-Kommandos aufruft. **Im [Abschnitt 3.10: Konfiguration neuerer Versionen von TortoiseSVN] finden Sie die notwendigen Einstellungen der jeweils aktuellen Softwareversion. Schauen Sie bitte dort nach, falls eine im nächsten Abschnitt dokumentierte Einstellung nicht mehr „passen“ sollte.**

### Konfiguration

TortoiseSVN bindet sich nach der Installation in das Kontextmenü des Windows Explorers ein, das erscheint, wenn in ihm mit der rechten Maustaste auf eine Datei oder einen Ordner geklickt wird. Das erscheinende Kontextmenü sieht so oder ähnlich der Abbildung aus:



Die Konfiguration wird unter dem Menüeintrag *Settings* vorgenommen. Folgende Einstellungsänderungen sind empfehlenerweise vorzunehmen:

#### Reiter *Main*

- *Force shell icon refresh* aktivieren.

#### Reiter *Look and Feel*

- *Recurse into folders to fetch the status* aktivieren.
- ~~In der Liste *Context Menu* sämtliche Haken entfernen.~~

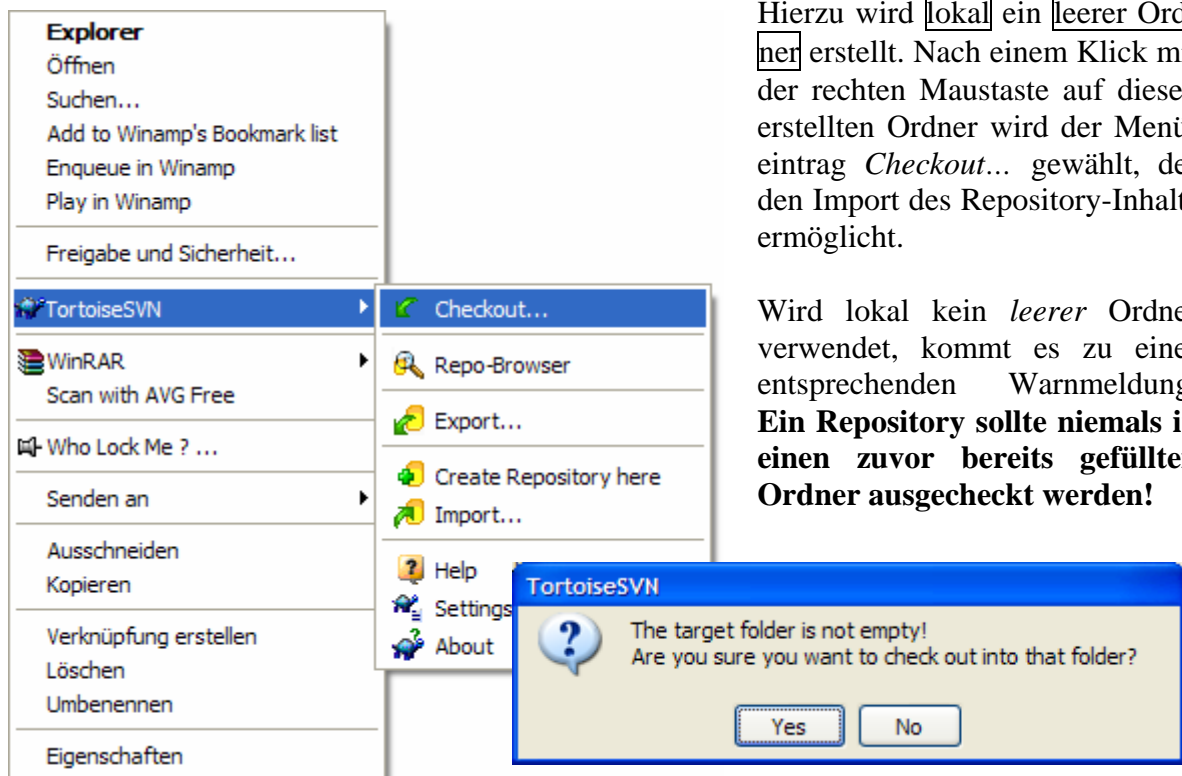
#### Reiter *Network*

Für das *TFH-OpenNet* ist hier der *Proxy Server* (*proxy.tfh-berlin.de*, Port 2784) ohne Benutzerdaten zu definieren und zu aktivieren.

Nach der Installation des deutschen LanguagePacks sind entsprechende Spracheinstellungen vorzunehmen, um die deutsche Sprache zu aktivieren. Sie befinden sich im Bereich *Misc* des Reiters *Main*.

## Auschecken eines Repositories

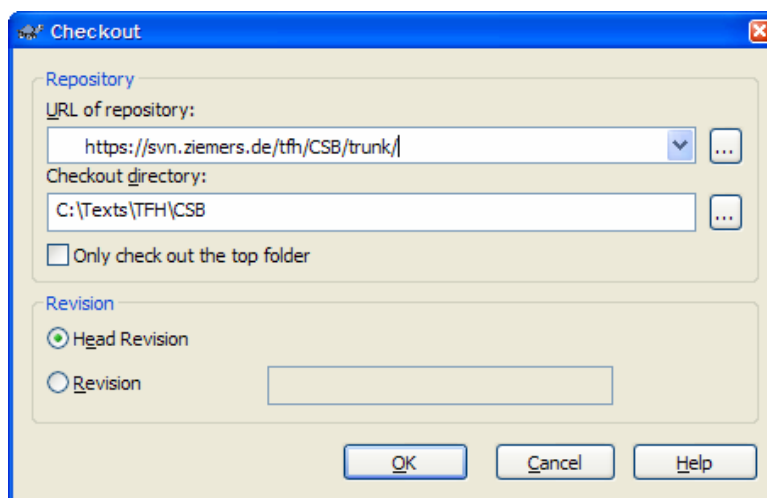
**Repository** nennt man die **zentrale Datenhaltung einer Dokumentenversionsverwaltung**. Um die Objekte eines Repositories nutzen zu können, muss sein Inhalt zunächst **ausgecheckt (checkout)** werden:



Hierzu wird **lokal** ein **leerer Ordner** erstellt. Nach einem Klick mit der rechten Maustaste auf diesen erstellten Ordner wird der Menüeintrag **Checkout...** gewählt, der den Import des Repository-Inhalts ermöglicht.

Wird lokal kein *leerer* Ordner verwendet, kommt es zu einer entsprechenden Warnmeldung. **Ein Repository sollte niemals in einen zuvor bereits gefüllten Ordner ausgecheckt werden!**

Nun muss die **Adresse des gewünschten Repositories** angegeben werden:



Dazu ist der **URL des gewünschten Repositories** einzugeben. TortoiseSVN fragt selbst nach den **Zugangsdaten** des Repositories, sofern eine Authentifizierung erforderlich ist.

Der eigentliche Auscheck-Vorgang des Repositories läuft dann vollautomatisch ab.

## Nutzen der Objekte

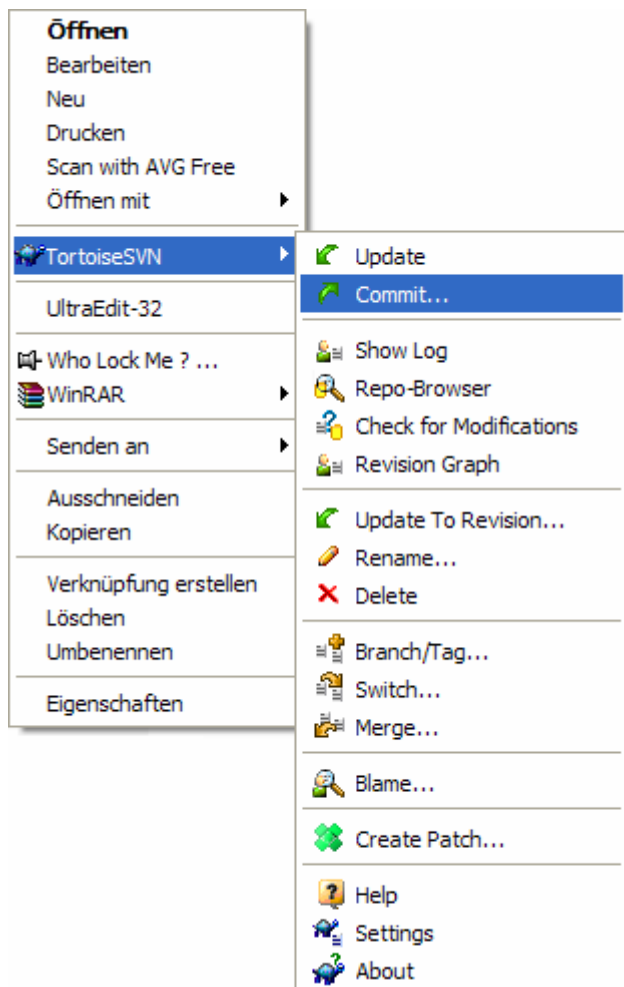
Nach dem erfolgreichen Auschecken eines Repositories steht der Inhalt im Ordner **zur lokalen Bearbeitung** zur Verfügung.

Dateien und Ordner können mittels des Explorer-Kontextmenüs **hinzugefügt** (*Add*), **gelöscht** (*Delete*) oder **umbenannt** (*Rename*) werden. In der Detailansicht des Explorers vermittelt ein zusätzliches Symbol zu jeder Datei und jedem Ordner ihren/seinen jeweiligen Zustand. Beispiele dieser Symbole können in den TortoiseSVN *Settings...* auf der Registerkarte *Look and Feel* nach Klick auf die Schaltfläche *Select Overlay Set* betrachtet werden.

Dateien und Ordner können ferner **lokal beliebig bearbeitet** werden. Ein Ausrufezeichen kennzeichnet diejenigen Dateien und Ordner, die lokal bearbeitet, aber noch nicht zurück in das Repository geschrieben wurden. Sie befinden sich **nicht in Synchronisation mit dem Repository**.

## Zurückschreiben der Objekte

Lokal bearbeitete Dateien und Ordner sollten „gelegentlich“ zurück in das Repository übertragen werden.



Dazu dient der Menüeintrag *Commit...* Alle lokal getätigten Modifikationen werden zurück **in das Repository geschrieben**.

Kommt es dabei zu einem **Versionskonflikt** zwischen einem Objekt im Repository und einem lokal bearbeiteten Objekt, macht eine Fehlermeldung darauf aufmerksam.

Konflikte sind **primär kommunikativ zwischen den Verursachern zu lösen**. Ein **Werkzeug zur Visualisierung des Konflikts namens TortoiseMerge** ist jedoch in TortoiseSVN enthalten, das sekundär bei der Lösung unterstützen kann.

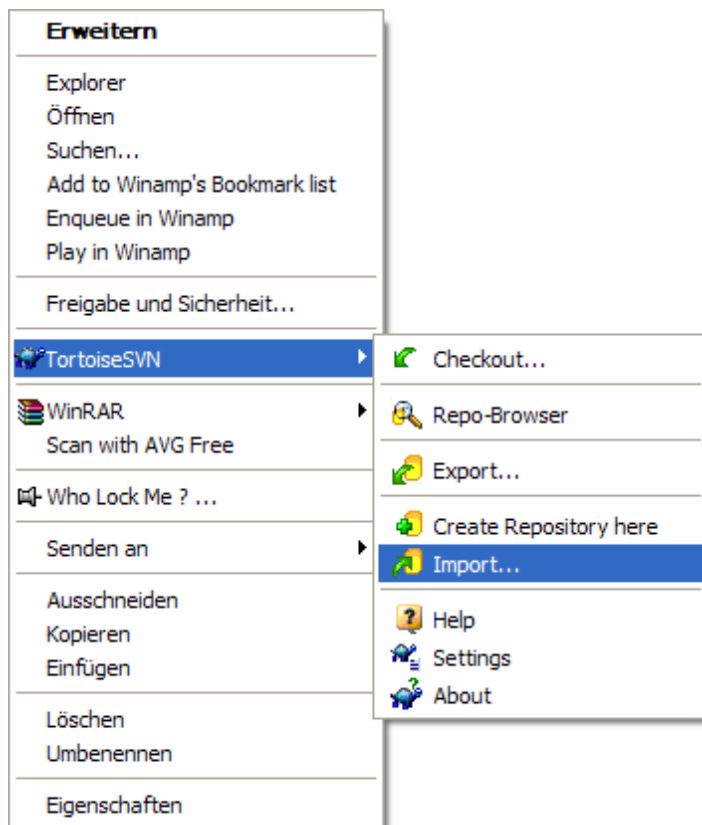
Dies wird im Abschnitt **[Konfliktauflösung]** dieser Dokumentation beschrieben.

## Aktualisierung lokale Objekte

Vor jedem Beginn einer neuen „thematischen Arbeitsetappe“ der Bearbeitung lokaler Dateien oder Ordner sollte der lokale Bestand mit dem des zentralen Repositories abgeglichen werden. Dazu dient der Menüeintrag Update im Kontextmenü (siehe auch letzte Abbildung). Hierbei werden alle aktuelleren Objekte des Repositories in das lokale Dateisystem übertragen. Lokal evtl. bereits bearbeitete Objekte werden dabei jedoch niemals überschrieben.

## Erstbefüllen eines Repositories

Zur Erstbefüllung eines leeren Repositories dient ein lokaler, mit Dateien und Ordnern gefüllter Ordner, der im allgemeinen nicht bereits unter Versionsverwaltung eines anderen Repositories stehen sollte (obwohl dies theoretisch möglich wäre).



Dieser Ordner wird mittels des Menüeintrags *Import...* in ein Repository übertragen. Dazu muss der URL des gewünschten Repositories angegeben werden. Der eigentliche Übertragungsvorgang geschieht dann vollautomatisch.

Es ist zu beachten, dass der übertragene Dateisystembaum durch diesen Vorgang nicht automatisch lokal unter Versionsverwaltung kommt. Hierzu dient ausschließlich der obige Skriptabschnitt **[Auschecken eines Repositories]**.

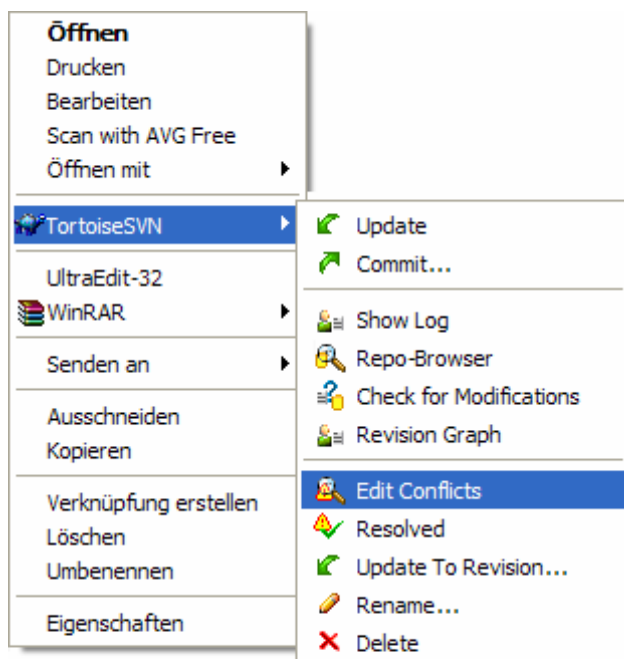
## Empfohlene Ordnungsstruktur eines Repositories

Als generelle Ordnungsstruktur eines jeden Repositories empfiehlt es sich, darin zunächst einen Ordner \trunk zu erstellen, in dem alle aktuellen Objektversionen abgelegt werden, einen Ordner \tags, in dem freigegebene Veröffentlichungen (*releases*) abgelegt werden, und einen Ordner \branches, in dem parallele Versionsstände abgelegt werden.

**Aktuelle Objektversionen** befinden sich aktuell in der Entwicklung (*Windows 2035*). **Veröffentlichungen** sind zur Benutzung bereits freigegebene Versionsstände (*Windows 95*). **Parallele Versionsstände** (*Office für Windows, Office für Macintosh*) sind parallele Entwicklungsstände, bei denen die Hoffnung besteht, sie in der Zukunft irgendwann wieder zusammenbringen zu können.

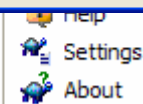
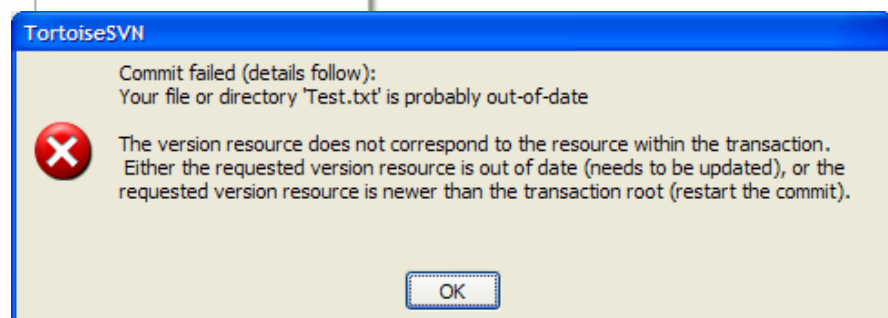
### 3.8 Konfliktauflösung

TortoiseSVN unterstützt bei der **Konfliktauflösung**. Im Falle eines Konflikts **beim Zurückschreiben von Dateien** mittels *Commit* kommt es zu der unten abgebildeten Fehlermeldung.



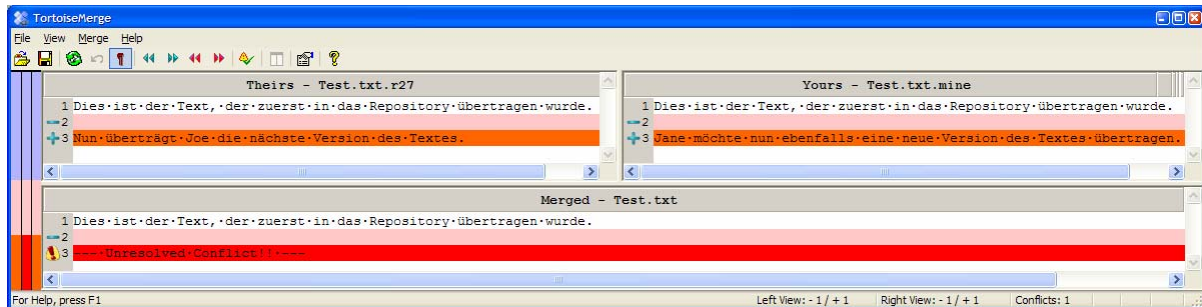
Dann muss zunächst eine **Aktualisierung des lokalen Datei- und Ordnerbestands** mittels *Update* durchgeführt werden. Das konfliktverursachende Dateiojekt wird dabei durch ein entsprechendes Symbol gekennzeichnet. Ferner sind drei neue Dateiobjekte entstanden: *Dateiname.mine*, *Dateiname.rrevision1* und *Dateiname.rrevision2*. Sie dienen zur werkzeugunterstützten Konfliktauflösung.

Über den Menüeintrag *Edit Conflicts* wird die Konfliktauflösung mit dem in TortoiseSVN eingebauten Werkzeug TortoiseMerge gestartet:

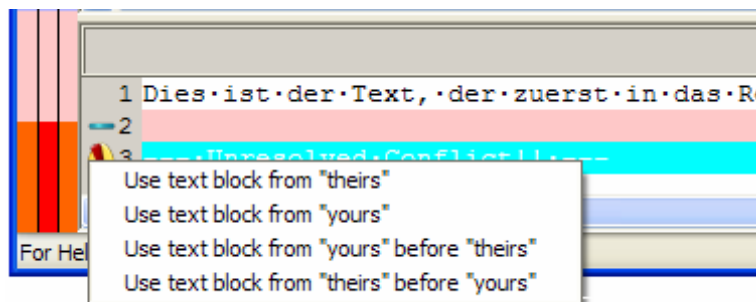


Das Hauptfenster von TortoiseMerge ist dreigeteilt. **Oben links** wird der Inhalt derjenigen Dateiversion dargestellt, die sich zur Zeit im Repository befindet, **rechts** davon die lokal bearbeitete Dateiversion.

**Unten** wird nun im folgenden Arbeitsvorgang das Ergebnis der Konfliktauflösung entstehen:



Die beiden Symbole + und – in den oberen Fensterhälften symbolisieren diejenigen Zeilen, die in den beiden Dateiversionen hinzugefügt (+) oder entfernt (–) wurden.



In der unteren Fensterhälfte wird ausgewählt, welche obige **Zeile zur Auflösung des Konflikts** herangezogen werden soll. Dazu klickt man mit der rechten Maustaste auf das Konfliktsymbol (Ausrufezeichen) in der unteren Fensterhälfte. Zur **Übernahmeaus-**

**wahl** steht die Zeile aus dem Repository (*Use text block from „theirs“*), die lokal bearbeitete Version (*Use text block from „yours“*), sowie zwei Varianten, die sowohl die Zeile aus dem Repository als auch die lokal bearbeitete in unterschiedlichen Reihenfolgen berücksichtigen (*Use text block from „yours“ before „theirs“* und *Use text block from „theirs“ before „yours“*).

## Ergebnis der Konfliktauflösung

Im unteren Fensterteil entsteht nach und nach eine neue, **konfliktbereinigte Variante** der beiden konfliktverursachenden Dateiversionen. Diese muss zum Abschluss innerhalb von TortoiseMerge gespeichert werden (*Save the modified file*). Danach kann auch der Konflikt selbst als aufgelöst gekennzeichnet (*Marks a file as resolved in Subversion*) und TortoiseMerge beendet werden.

Nun ist im Kontextmenü des Windows Browsers ein *Commit...* möglich, und die konfliktbereinigte Dateiversion kann in das Repository übertragen werden.



## 3.9 Aufsetzen eines Subversion-Servers

---

Einen eigenen Subversion-Server aufzusetzen, ist recht aufwendig, und würde den Rahmen dieses Dokuments bei weitem übersteigen. Hier wird auf zahlreiche Dokumentation im Internet verwiesen. Ein generelles Vorgehen sei an dieser Stelle aber dennoch kurz umrissen:

### Repository erstellen

Zur Erstellung eines neuen Repositories dient der Subversion-Befehl

```
svnadmin create --fs-type fsfs repository
```

oder die Anwendung TortoiseSVN. Die Verwendung der in Subversion integrierten Datenbank *BerkeleyDB* bietet sich heutzutage nicht mehr an. Es wird vielmehr empfohlen, dass mittlerweile ausreichend stabile, ebenfalls integrierte Dateisystem *fsfs* zu verwenden.

### Zugriff realisieren

Der Zugriff auf die Repositories wird im einfachsten Fall über den in Subversion bereits eingebauten Serverdienst realisiert. Er ist allerdings nicht sehr stabil und weist auch keine hohe Leistungsfähigkeit auf. Ein vollwertiger Webserver (beispielsweise *Apache*) und der HTTP-Protokollaufsatz *WebDAV* eignen sich hier besser. Ein vollwertiger Webserver bietet ferner den Vorteil, Subversion auch über den normalen Webbrowser und durch Proxies und Routers hindurch verwenden zu können, ohne dass es dabei zu größeren Problemen mit Firewall- oder NAT-Einstellungen kommen würde.

### Zugriffsberechtigung

Ein Webserver kann auch die Überprüfung und Beachtung der Zugriffsberechtigungen zu den Subversion-Repositories verwenden, jedoch bringt Subversion bereits ein komfortables rollenbasiertes Berechtigungskonzept mit, dessen Granularität bis auf den einzelnen Ordner in den Repositories reicht.

## **3.10 Konfiguration neuerer Versionen von TortoiseSVN**

---

### Reiter *General*

- Hier sind keine Einstellungen erforderlich.

### Reiter *Look And Feel*

- Aktivieren Sie den Haken der Zeilen *Checkout...*, *Update* und *Commit...* im Listview *Cascaded Context Menu*.

### Reiter *Look and Feel/Icon Overlays*

- Aktivieren Sie *Show overlay status icons recursively*.

### **Konfiguration des TFH-Netzwerkzugangs**

Das Rechenzentrum der TFH legt ein recht „dynamisches Verhalten“ an den Tag, wenn es um die augenblicklich gültige Konfiguration ihres Netzwerks geht. Die jeweils gültigen Einstellungen stehen auf jeden Fall in folgendem Skript:

<http://www.tfh-berlin.de/RZ/RZ.pac>

Interpretieren Sie den Inhalt dieses Skripts, falls die Verbindung zu TortoiseSVN nicht erfolgreich verlaufen sollte.