

Thema, Ziele: Versionsverwaltung mit Subversion (SVN)

Aufgabe 1: TortoiseSVN kennenlernen – in einem bestehendem Repository stöbern

1. Einführung

Subversion wird häufig zur Verwaltung von Open-Source-Projekten eingesetzt. "CxxTest" ist ein solches Open-Source-Projekt das mit Subversion verwaltetet wird. Es beinhaltet, wie CppUnit, ebenfalls ein C++ Unit-Testing-Framework.

Dieses Projekt soll nun dazu dienen "TortoiseSVN" kennenzulernen. "TortoiseSVN" ist kein eigenständiges Programm, sondern ein Programm, das in den Windows-File-Explorer integriert ist.

Allgemeines – **Wichtig:**

- Keine Leerzeichen oder Umlaute in Datei- und Verzeichnisnamen verwenden! – das erspart Ärger.
- Verwendete Abkürzungen: FE für File-Explorer, KM für Kontextmenü, KME für Kontextmenü-Eintrag.

2. Repository Browser, Log Messages

Öffnen Sie ein File-Explorer-Fenster und erzeugen Sie damit in Ihrem HOME-Verzeichnis ("U:\") ein neues Verzeichnis "tmp_cxxtest" (→ "U:\tmp_cxxtest"). Wählen Sie nun das soeben erstellte Verzeichnis an und öffnen Sie mit der rechten Maustaste das Kontextmenu. Gehen Sie darin mit der Maus zum Eintrag "TortoiseSVN" und wählen Sie dann "Repro-Browser" (linke Maustaste).

Es öffnet sich dann ein Fenster "URL", in das Sie den URL vom Repository des Projekts "CxxTest" eingeben können.

Dieser lautet "http://cxxtest.tigris.org/svn/cxxtest" (Username: guest, kein Passwort).

Beachten Sie:

SVN-Repositories werden *immer* über eine URL identifiziert. Dieser beinhaltet das verwendete Protokoll (hier "http://"), den Ort des File-Servers (hier "cxxtest.tigris.org") sowie den Ort des Repositories auf dem File-Server (hier "/svn/cxxtest").

Nach Eingabe dieses URL's (mit Okay bestätigen) öffnet sich der Repository-Browser, eine Art File-Explorer, mit dem Sie aber innerhalb des Repository auch in die Vergangenheit "reisen" können, das heisst frühere Revisionen betrachten können.

Beachten Sie, auf der obersten Ebene innerhalb dieses Repository finden Sie die üblichen Verzeichnisse "trunk", "tags", "branches" (und hier evt. zusätzlich "stables").

2.1 Fragen:

- a) bei welcher Revision-Number ist man bei diesem Projekt inzwischen angelangt?
- b) In welchem Verzeichnis der obersten Ebene wurde zuletzt eine Änderung durchgeführt?
- c) Wann (Datum, Zeitpunkt) wurde diese Änderung durchgeführt?

2.2 Oben rechts finden Sie den mit "HEAD" beschrifteten Button zur Eingabe der Revision welche angezeigt werden soll. ("HEAD" steht für die zurzeit höchste Revision-Number im Repository.)

Finden Sie damit heraus, wann (Datum, Zeitpunkt) die ersten Dateien eing检eicht wurden. Wie viele Dateien (gewöhnliche und Verzeichnisse) wurden dabei eingeeicht?

Gehen Sie wieder zurück zur letzten (neuesten) Revision. Wählen Sie nun das Verzeichnis "trunk" an und öffnen Sie im KM "Show Log".

Wählen Sie dort Revision 129 (abgekürzt "r129") an und analysieren Sie die Meldungen im untenstehenden Datenfenster. Beachten Sie auch die ausführliche Beschreibung im mittleren Fenster (Fenster vergrössern).

2.3 Schliessen Sie alle geöffneten "Log Messages"-Fenster. Wählen Sie im "Repository Browser" das Verzeichnis "tags" an.

Finden Sie nun heraus, wer (Autor) dieses Verzeichnis "tags" erstellt hat.

3. "checkout" – Daten aus dem Repository "auschecken"

"Auschecken" heisst, alle Daten (einer Version) aus dem Repository in ein neues, noch nicht durch SVN verwaltetes, lokales Verzeichnis zu kopieren.

Durch diesen Vorgang entsteht erst die sogenannte "working-copy", welche immer auch durch SVN verwaltet wird.

Normalerweise werden nicht alle im Repository vorhandenen Daten "ausgecheckt" sondern nur der "trunk" oder der interessierende Zweig aus den Verzeichnissen "tags" oder "branches".

3.1 Im Folgenden soll (nur) der "trunk" aus dem Repository in das lokale Verzeichnis "tmp_cxxtest" kopiert, also "ausgecheckt" werden:

Schliessen Sie alle TortoiseSVN-Fenster. Wählen Sie nun im KM des FE im Verzeichnis "tmp_cxxtest" den Eintrag "SVNCheckout..." aus. Geben Sie dann im "Checkout" Fenster als Repository-URL folgendes ein:

`"http://cxxtest.tigris.org/svn/cxxtest/trunk"`

Beachten Sie die Endung "**trunk**". Damit wird nur – wie beabsichtigt - der Inhalt des Repository-internen Verzeichnisses "trunk" kopiert.

Beachten Sie auch, dass Sie in diesem Fenster die gewünschte Revision angeben können, welche "ausgecheckt" werden soll. – Das ist unsere Zeitmaschine! – (Wir bleiben bei "HEAD", der neuesten Revision.)

Daneben kann auch noch das "Log"-Fenster geöffnet werden.

Bestätigen Sie nun durch "OK", dass der "checkout" ausgeführt werden soll.

3.2 Betrachten Sie, nachdem der "checkout" ausgeführt worden ist, wie sich im File-Explorer die Symbole verändert haben. Betrachten Sie auch die Dateien in den entstandenen Unterverzeichnissen.

Das Verzeichnis "tmp_cxxtest" steht nun unter Subversion-Verwaltung, erkennbar mit "TortoiseSVN" am grünen "Okay-Haken", im File-Explorer am darin vorhandenen Unterverzeichnis ".svn". Dieses Unterverzeichnis ".svn" enthält Verwaltungsinformation. Es enthält z.B. den URL des zugehörigen Repository.

3.3 Ändern Sie nun im Unterverzeichnis "sample" die Datei "yes_no_runner.cpp" so, dass zusätzlich noch der Spruch "Hello World" ausgegeben wird.

Beachten Sie nun, wie die Symbole im File-Explorer sich ändern.

4. "Diff" – Änderungen betrachten

4.1 Wählen Sie im FE die Datei "yes_no_runner.cpp" an und öffnen Sie im KM "TortoiseSVN"→"Diff". Sie sehen nun die alte und neue Version übersichtlich neben einander dargestellt.

5. "Commit" – Daten ins Repository zurückschreiben, ins Repository "einchecken"

5.1 Die soeben vorgenommene Änderung soll nun wieder ins Repository zurückgeschrieben werden.

Wählen Sie dazu im KM des FE im Verzeichnis "tmp_cxxtest" den Eintrag "SVN Commit..." an. Geben Sie anschliessend einen sinnvollen Kommentar ein und führen Sie das Commit aus.

Es öffnet sich nun ein Fenster zur Eingabe von Benutzername und Passwort. Brechen Sie nun hier das Commit ab.

6. Daten in der "working copy" rückgängig machen.

6.1 Da wir mit unserem Änderungsversuch gescheitert sind, soll auch in unserer Arbeitskopie "working copy" alle Änderungen im Verzeichnis "sample" – aber nur darin – rückgängig gemacht werden.

Wählen Sie dazu im KM des FE im Verzeichnis "sample" den Eintrag "TortoiseSVN"→"Revert" an. Wählen Sie die Dateien aus und führen Sie das "Revert..." aus. Beachten Sie, wie sich die Symbole im FE dadurch ändern.

Aufgabe 2: Mit eigenem (lokalen) Repository arbeiten – Projekt "AliBaba"

1. Eigenes, lokales Repository erzeugen

Für ein Projekt mit Namen "AliBaba", bei dem Sie der einzige Projektbeteiligte sind, soll ein lokales SVN-Repository erstellt werden.

Erstellen Sie dazu mit Hilfe des FE in Ihrem Home-Verzeichnis das Verzeichnis "svn-root" und darin das Verzeichnis "svn-repos_AliBaba".

Wählen Sie nun im KM des FE im Verzeichnis "svn-repos_AliBaba" den Eintrag "TortoiseSVN" → "Create Repository here" an.

Betrachten Sie anschliessend die dadurch erzeugte File-Struktur im Verzeichnis "svn-repos_AliBaba".

2. Grundstruktur ins Repository schreiben

Erstellen Sie mit Hilfe des FE in Ihrem Home-Verzeichnis ein Verzeichnis "tmp" und darin die Verzeichnisse "trunk", "tags" und "branches".

Wählen Sie nun im KM des FE im Verzeichnis "tmp" den Eintrag "TortoiseSVN" → "Import..." an und geben Sie anschliessend als Repository den URL "file:///U:/svn-root/svn-repos_AliBaba" an, sowie den Kommentar "Grundstruktur erzeugt" (o.ä.) an und schliessend Sie den "Import" ab.

Löschen Sie nun das Verzeichnis "tmp".

Betrachten Sie anschliessend mit Hilfe des Repository-Browsers den neuen Inhalt Ihres Repositories.

3. "checkout" – Aus dem (eigenen) Repository auschecken

Erstellen Sie mit Hilfe des FE in Ihrem Home-Verzeichnis ein Verzeichnis "work_AliBaba".

Wählen Sie nun im KM des FE in diesem Verzeichnis "work_AliBaba" den Eintrag "SVN Checkout..." an und geben Sie dort als Repository den URL "file:///U:/svn-root/svn-repos_AliBaba/trunk" an.

Das Verzeichnis "work_AliBaba" steht nun unter Subversion-Verwaltung, erkennbar mit "TortoiseSVN" am grünen "Okay-Haken", im File-Explorer am darin vorhandenen Unterverzeichnis ".svn".

Merke: Ein Verzeichnis, das unter SVN-Verwaltung steht, wie hier "work_AliBaba", wird allgemein als "Arbeitsverzeichnis", "working-directory" oder "working-copy" bezeichnet.

4. Im eigenen Arbeitsverzeichnis arbeiten

Erstellen Sie in ihren SVN-Arbeits-Verzeichnis "work_AliBaba" das C++ Programm "helloworld.cpp", welches den Spruch "Hello World!" ausgibt.

Diese Datei "helloworld.cpp" steht jetzt noch nicht unter SVN-Verwaltung. Um sie in diese aufzunehmen, wählen Sie im KM des FE bei der Datei "helloworld.cpp" den Eintrag "TortoiseSVN" → "add..." an. Die Datei "helloworld.cpp" ist nun im FE mit einem blauen Plus-Zeichen dekoriert.

5. "commit" – Die Daten aus dem Arbeitsverzeichnis ins (eigene) Repository einchecken

Die soeben vorgenommene Änderung bei unserem Projekt "AliBaba" (das Hinzufügen von "helloworld.cpp") soll nun wieder ins Repository zurückgeschrieben werden.

Wählen Sie dazu im KM des FE im Verzeichnis "work_AliBaba" den Eintrag "SVN Commit..." an. Geben Sie anschliessend einen sinnvollen Kommentar ein und führen Sie das Commit aus.

6. "update" – Das eigene Arbeitsverzeichnis aktualisieren

Wenn unklar ist, ob das eigene Arbeits-Verzeichnis denselben Stand aufweist, wie das Repository, ist ein "update" sinnvoll. Diese Operation ist grundsätzlich "ungefährlich", da man ja bei Konflikten darauf aufmerksam gemacht wird.

Führen Sie das "update" aus.

Aufgabe 3: Verständnis- und Repetitionsfragen zu Subversion

- a) Wie viele Zähler für Revisions (Anzahl Revision-Counters) gibt es bei Subversion pro Repository?
- b) Weshalb wird bei SVN empfohlen für jedes eigenständige Projekt ein eigenes Repository zu verwenden?
- c) Wie lauten die zwei am häufigsten gebrachten SVN-Befehle (Subcommands) um einen Verzeichnisbaum von einem Repository in ein Arbeitsverzeichnis zu kopieren.
- d) Erklären Sie kurz (1..2 Sätze) den Unterschied zwischen den SVN-Befehlen "checkout" und "update".
- e) Finden Sie heraus, wozu der SVN-Befehl "export" dient. (Von wo nach wo werden Daten "exportiert"?)

Tipps zu Subversion und TortoiseSVN – häufige Probleme und ihre Lösung

1. **Problem:** Es werden nicht alle Icons von Subversion angezeigt.
Grund: Subversion Icon's von "Network drives" und "Removable drives" werden standardmässig nicht angezeigt.
Abhilfe: Im KM des FE "TortoiseSVN" → "Settings..." anwählen, danach "Icon Overlays" auswählen und bei "Drive Types" "Removable drives" und "Network drives" aktivieren.
2. **Problem:** Die Verzeichnisse ".svn" werden im File-Explorer nicht angezeigt.
Grund: Diese Verzeichnisse sind "hidden" (versteckt) und werden vom FE je nach Konfiguration deshalb nicht angezeigt.
Abhilfe: (Windows XP) Im File-Explorer via "Extras" → "Ordneroptionen" → "Ansicht" den Eintrag "Versteckte Dateien und Ordner" entsprechend ändern.
(Windows 7): ??
3. **Problem:** Nach dem Auschecken weisen alle Dateien als Änderungsdatum das aktuelle Datum (also alle das gleiche) statt ihr ursprüngliches auf.
Abhilfe: Im KM des FE "TortoiseSVN" → "Settings..." anwählen, danach "General" auswählen und bei "Subversion" das Kästchen "Set file dates to the "last commit time"" aktivieren. Dadurch erhalten die Dateien zwar nicht exakt ihr ursprüngliches Datum, aber wenigstens das von ihrem letzten "commit". (Bei SVN ohne TortoiseSVN im Config-File "use-commit-times = yes" hinzufügen.)