

Neues in der TFS 2008-Versionsverwaltung

Compare, Annotate und Destroy

Der neue Team Foundation Server bedeutet für die Versionsverwaltung eher eine Evolution, denn eine Revolution. Und das ist auch gut so! TFS-Experte Neno Loje stellt die Neuerungen im Detail vor.

Auf einen Blick



Neno Loje ist als freier Berater Teil von thinkecture und Microsoft MVP für Visual Studio Team System & Team Foundation Server. Bei Fragen erreichen Sie ihn über seine Webseite www.teamssystempro.de.

Inhalt

- › Was ist neu in der Versionsverwaltung des TFS 2008?
- › Compare, Annotate, Destroy, Get latest on check-out.
- › Workspace-Mapping, Auswerten von Override-Begründungen.

dotnetpro.code

A0807TFS2008

Das Upgrade auf die neue Version 2008 des Team Foundation Servers ist technisch gesehen kein sehr großer Schritt. Vielmehr sind die meisten Änderungen im Bereich der Fehlerbereinigung und anhand der tatsächlichen, wie auch der gefühlten Geschwindigkeit zu finden.

Zugegeben, wenn Sie nach dem wichtigsten Grund fragen, warum Sie von einem bestehenden TFS 2005 auf den neuen TFS 2008 wechseln sollten, empfiehlt sich als Lektüre die Liste der behobenen Fehler. Danach kann man sich kaum mehr vorstellen, jemals erfolgreich mit der alten Version gearbeitet zu haben!

Den größten Teil der neuen Funktionen und Fehlerbeseitigungen sind übrigens dem internem Einsatz des Produkts bei Microsoft zu verdanken. Der Produktiv-TFS von Microsofts Entwicklungsabteilung enthielt Ende April 2008 mehr als zwei Terabyte komprimierter Daten und insgesamt 236 Millionen Dateien [1].

Aber nun zu den eigentlichen Neuerungen:

- **Destroy** erlaubt das endgültige Löschen von Dateien.
- **Annotate** bringt ans Licht, wer wann was an einer Datei verändert hat.
- **Compare** erlaubt das Vergleichen von lokalen Ordnern oder TFS-Ordnern mit anderen TFS-Ordnern.
- Die Funktion **Get latest on check-out** ruft beim Auschecken die aktuellste Version ab.
- Verbesserungen beim Workspace Mapping erlauben dem Entwickler, nur noch diejenigen Dateien vom Server zu holen, die er auch wirklich benötigt.
- Check-in Policies: Das Umgehen der Richtlinien (**Override**) kann nun durch einen eigenen Bericht ausgewertet werden.
- Offline arbeiten, direkt aus Visual Studio heraus.

Destroy: Endgültiges Löschen

Der TFS 2005 bietet nur eine logische Löschfunktion, das heißt, Gelöschtes wurde im Normalfall lediglich ausgeblendet und konnte über den Visual-Studio-Optionsdialog wieder eingeblen-

det werden. Man konnte gelöschte Dateien und Ordner also problemlos wiederherstellen. Auch beim Löschen von Teamprojekten (über *tfsdeleteproject.exe*) wurde der Teil in der Versionsverwaltung nur logisch gelöscht und konnte nachträglich rekonstruiert werden.

Der TFS 2008 bietet neben dem beschriebenen logischen Löschen (**Delete**) nun auch das physikalische Löschen (**Destroy**) [2] an, welches die Daten unwiderruflich zerstört. Das Kommando **Destroy** steht nur auf Kommandozeile (*tf.exe destroy*), nicht in der Visual-Studio-IDE, zur Verfügung und kann ausschließlich von TFS-Administratoren genutzt werden.

Das ist auch so gewollt, denn in der Praxis wird man einen Zerstörbefehl eher selten benutzen wollen – zumal man damit die Historie verfälschen kann. Beim Aufruf von **Destroy** unterscheidet man zwei Einsatzfälle:

- Das Löschen von Dateien, die man definitiv nicht mehr benötigt, weil beispielsweise das Projekt eingestellt wurde oder versehentlich etwas eingechekkt wurde und man von vorne beginnen möchte. Nach der Zerstörung ist nicht mehr erkennbar, dass diese Dateien und Ordner jemals existierten.
- Der zweite Einsatzfall ist die Verkleinerung der Versionsverwaltung. Etwa wenn durch häufiges Einchecken großer Dateien das Volumen stetig wächst und eigentlich nur die letzte Version oder die Versionen der vergangenen Woche benötigt werden.

Dabei kann während der Zerstörung bestimmt werden, ob alle alten Versionen bis auf die aktuellste verschwinden sollen, oder alles bis zu einem bestimmten Datum gelöscht werden soll. Die Versionshistorie kann dabei erhalten bleiben (Parameter */keephistory*). Versucht man eine zerstörte Version wiederherzustellen, erhält man die in Abbildung 1 gezeigte Fehlermeldung.

Ein Beispiel für das endgültige Löschen älterer Versionen: Auf der Kommandozeile (*tf destroy*) wird der Serverpfad angegeben (hier: *\$/AdventureWorks/adventureworks/Default.aspx.cs*), der Parameter */keephistory* angehängt, um die Historie

Den vollständigen Artikel lesen Sie in:



dotnetpro 07/2008 auf Seite 16

dotnetpro-Abonnenten können diesen über das Online-Archiv herunterladen:

<http://www.dotnetpro.de/articles/onlinearticle2680.aspx>