

## Kapitel 54

# Quellcodeverwaltung

### In diesem Kapitel:

Visual SourceSafe (VSS)

1124

Quellcodeverwaltung mit Subversion

1128

Dieses Kapitel thematisiert die Quellcodeverwaltung mit dem Microsoft-Produkt *Visual SourceSafe 2005* (VSS) und dem Open Source-Werkzeug *Subversion*.

## Visual SourceSafe (VSS)

Visual SourceSafe (VSS) ist das Versionsverwaltungssystem, das Microsoft bereits vor dem .NET-Zeitalter mit Visual Studio ausgeliefert bzw. einzeln verkauft hat. Im Vergleich zu Konkurrenzprodukten (siehe [HS06]), auch im Vergleich zu Open Source-Werkzeugen wie CVS und Subversion, hat VSS in der Vergangenheit keinen begeisternden Eindruck hinterlassen.

---

**HINWEIS** Die aktuelle Version ist Visual SourceSafe 2005 (VSS 2005). Es gibt keine neue Version im Zuge von Visual Studio 2008.

---

Anders als Konkurrenzprodukte besitzt SourceSafe keine echte Client-Server-Architektur; vielmehr ist das Projektarchiv nur eine Freigabe im Netzwerk, die eine SourceSafe-Datenbank in einem proprietären Format enthält. Kernelement dieser Datenbank ist die Datei *srcsafe.ini*. Unterhalb dieser Datei existieren zwei Verzeichnisse */Data* und */Temp*.

Der Filesharingzugriff der Clients auf die Datenbank war in der Vergangenheit eine der großen Schwächen von SourceSafe: Das Redmonder Produkt baut seine Kommunikation nicht direkt auf TCP/IP auf, sondern setzt das Server Message Block (SMB)-Protokoll voraus. Soll SourceSafe auch in Umgebungen zum Einsatz kommen, in denen die TCP-Ports 139 (SMB over NetBIOS over TCP/IP; vor Windows 2000) bzw. 445 (SMB over TCP/IP; Option ab Windows 2000) nicht zur Verfügung stehen, benötigt man also ein Virtuelles Privates Netzwerk (VPN). Dieses Manko konnte man nur mit dem 200 US-Dollar teuren Zusatzprodukt SourceOffSite der Firma SourceGear überwinden.

In VSS 2005 ist der HTTP-Zugang bereits im Kernprodukt unter dem Namen *SourceSafe Internet* enthalten. SourceSafe Internet benötigt einen IIS-Webserver auf dem System, auf dem die Datenbank liegt. Als Client kann beim HTTP-Zugang nur Visual Studio ab Version 2005 zum Einsatz kommen.

Die Datenbank bietet dem Anwender VSS in der Form eines Dateisystems an, in welchem er Projekte und Dateien ablegen kann. Die Clientanwendung erhält dabei ein Arbeitsverzeichnis des gewünschten Projekts, und der Anwender kann in diesem arbeiten.

---

**HINWEIS** Große Schwierigkeiten hatte Microsoft in der Vergangenheit mit dem VSS-Datenbankformat: In Foren berichteten zahlreiche Anwender über Datenbankinkonsistenzen und Datenverluste. Microsoft verspricht bei der VSS 2005-Installation: »Robustness improvements reduce the need to analyze, administer, and repair databases.«. Ob dies gerechtfertigt ist, wird die Praxis erst noch zeigen müssen. Microsoft empfiehlt in der Produktdokumentation weiterhin den regelmäßigen Einsatz des mitgelieferten Werkzeugs *Analyze.exe* einmal pro Woche.

---

## VSS-Clients

Microsoft liefert drei verschiedene Clients für VSS:

- diverse Kommandozeilenwerkzeuge (*ss.exe*, *ssarc.exe* etc.),
- eine Windows-Anwendung mit dem Namen Visual SourceSafe Explorer (Abbildung 54.1),
- ein Add-In für Visual Studio.

Zur Verwaltung von Datenbanken und Benutzern gibt es noch eine Windows-Anwendung mit Namen *SourceSafe Administrator*.

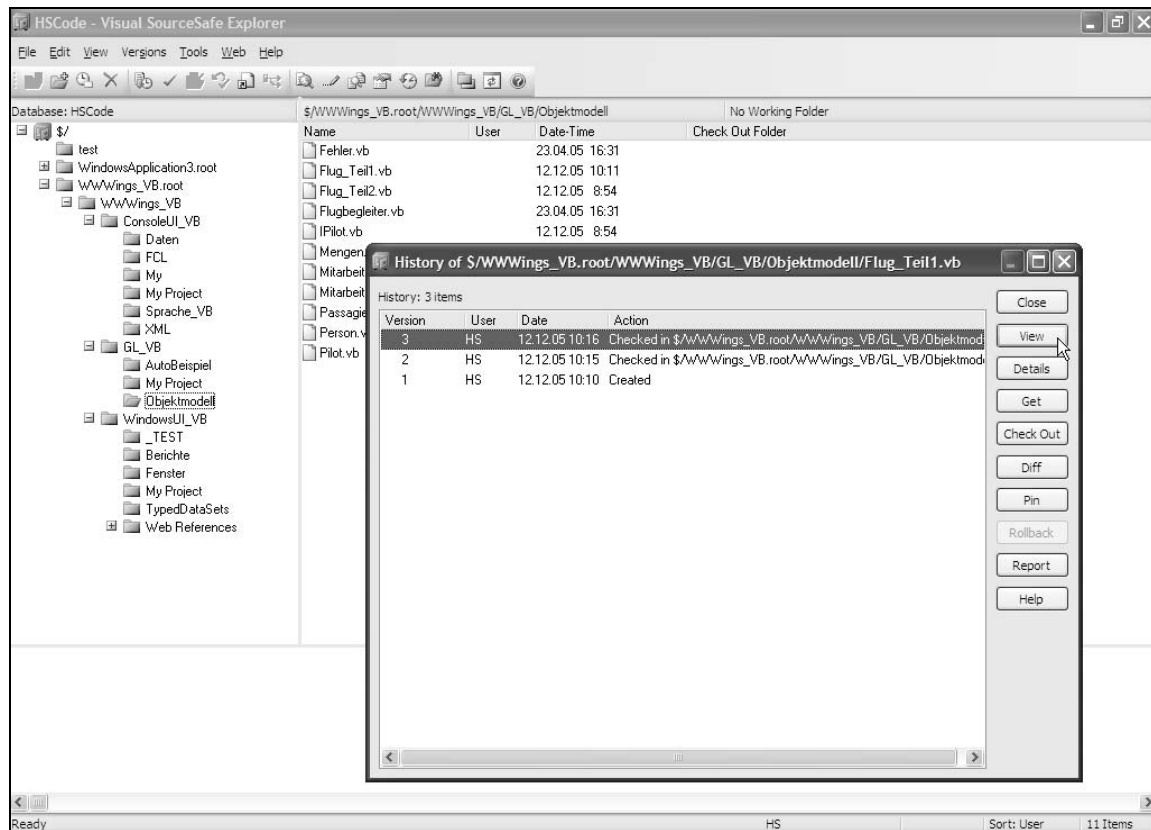


Abbildung 54.1 Visual Studio-Projekte im Visual SourceSafe Explorer

## Erstellen einer SourceSafe-Datenbank

Das Erstellen einer SourceSafe-Datenbank ist über den Menüpunkt *File/New Database* im SourceSafe Administrator möglich. Bei der Erstellung sind ein Dateisystempfad und das gewünschte Sperrmodell auszuwählen.

Visual Studio bietet zwei Modelle für das Sperren von Dateien:

- **Sperren-Ändern-Entsperren-Modell (Lock-Modify-Unlock-Modell):** Eine Datei wird beim Auschecken gesperrt, sodass nur ein Benutzer gleichzeitig an der Datei arbeiten kann. So entstehen keine Änderungskonflikte in den Codedateien.
- **Kopieren-Ändern-Zusammenführen-Modell (Copy-Modify-Merge-Modell):** Alle Entwickler können jederzeit alle Dateien auschecken und bearbeiten. Bei Änderungskonflikten versucht SourceSafe, die Änderungen automatisch zusammenzuführen. Wenn dies nicht automatisch möglich ist, sind die Konflikte manuell aufzulösen.

**HINWEIS**

Es gibt viele Überlegungen und Meinungen dazu, welches das bessere Modell für das Versionsmanagement ist. Sie sollten die Entscheidung von der Änderungshäufigkeit und der Teamgröße abhängig machen.

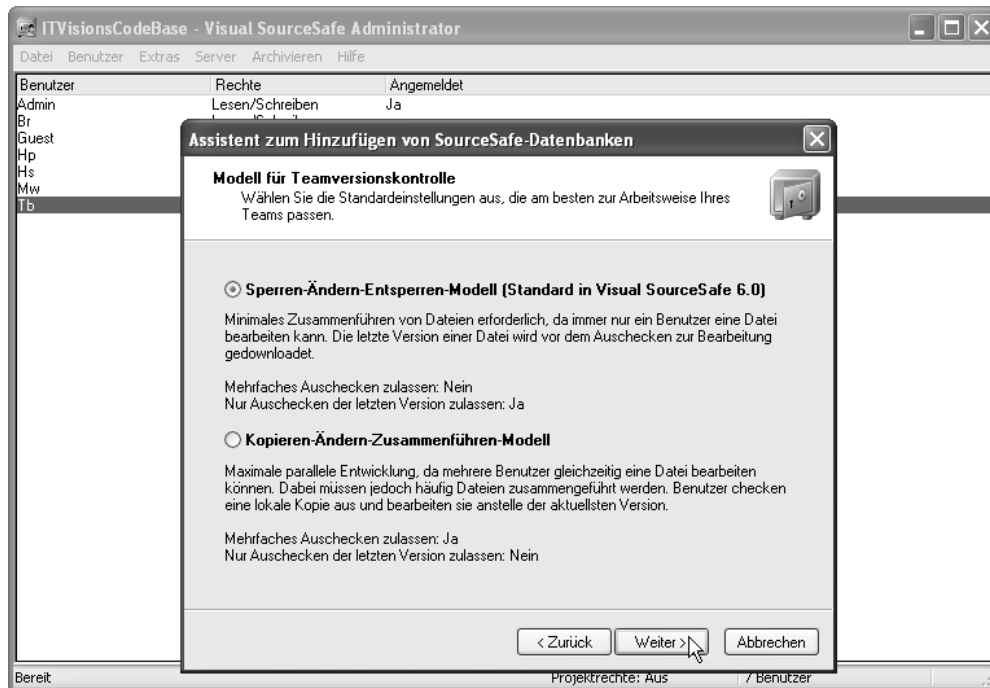


Abbildung 54.2 Auswahl des Sperrmodells beim Anlegen einer Datenbank

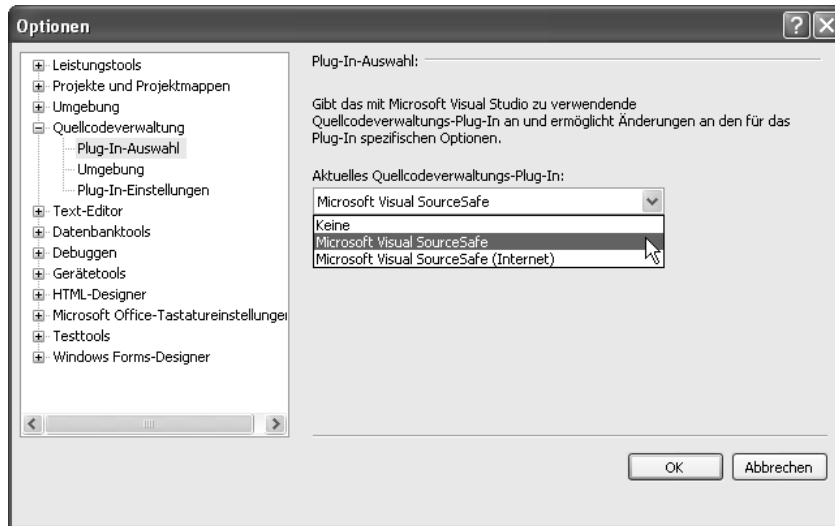
Standardmäßig unterscheidet VSS lediglich zwischen Lese- und Lese-/Schreibrechten für das gesamte Projekt. Zusätzlich können noch Benutzer zu Administratoren erhoben werden; diese können dann andere Benutzer anlegen, Zugriffsrechte erteilen und weitere administrative Tätigkeiten ausführen. Sollte diese Form der Rechtevergabe nicht genügen, so kann man eine erweiterte Rechtevergabe aktivieren (*Tools/Options/Project Rights* im SourceSafe Administrator). Dabei unterteilt VSS die Benutzerrechte in die vier Rechte Lesen, Lesen/Schreiben (Auschecken/Einchecken), Hinzufügen und Löschen.

Da das Anlegen und Verwalten von Projekten über Verzeichnisse abläuft und die Verwaltung der Benutzerrechte lediglich Verzeichnisse betrifft, kann der VSS-Administrator die Benutzerrechte auch nur auf Projekt- und Verzeichnisebene vergeben. Eine feinere Rechtevergabe – um beispielsweise eine einzelne Datei nur für den schreibenden Zugriff durch einen Benutzer freizugeben – existiert nicht.

## Integration in Visual Studio

Visual SourceSafe bietet ein Add-In für Visual Studio, das aber nicht im Add-In-Manager erscheint. VSS muss nach der Installation von Visual Studio nachträglich auf jedem Client installiert werden. Die Serverkomponenten von VSS werden nur auf den Computern benötigt, auf denen die zentralen Quelldatenbanken verwaltet werden sollen.

Nach der Installation des VSS-Add-Ins scheint sich nichts in Visual Studio verändert zu haben. Tückischerweise werden die Quellcodeverwaltungsoptionen erst sichtbar, nachdem der Entwickler unter *Extras/Optionen/Quellcodeverwaltung* entweder *Visual SourceSafe* oder *Visual SourceSafe (Internet)* als Provider aktiviert hat.



**Abbildung 54.3** Aktivierung von VSS in Visual Studio

Nach der Aktivierung des VSS-Add-Ins stehen in Visual Studio folgende Funktionen zur Verfügung:

- Beim Erstellen eines Projekts gibt es ein Kontrollkästchen *Add to Source Control*. Wenn dieses Kontrollkästchen aktiviert wird, fragt Visual Studio beim Anlegen des Projekts in einem Dialogfenster nach der VSS-Datenbank, in die das Projekt eingestellt werden soll.
- Beim Öffnen eines Projekts existiert die Option, ein Projekt aus einer bestehenden VSS-Datenbank zu öffnen.
- Das Menü *Datei/Quellcodeverwaltung* (*File/Source Control*) bietet die Möglichkeit, ein bereits geöffnetes Projekt unter die Versionsverwaltung zu stellen, das heißt in eine SourceSafe-Datenbank aufzunehmen. Wenn das Projekt unter Versionsverwaltung steht, bietet das Menü zahlreiche Versionsverwaltungsfunktionen wie Check Out, Check In oder View History.
- Viele Funktionen des Menüs *Datei/Quellcodeverwaltung* sind auch über das Kontextmenü eines Projekts erreichbar (Abbildung 54.4).

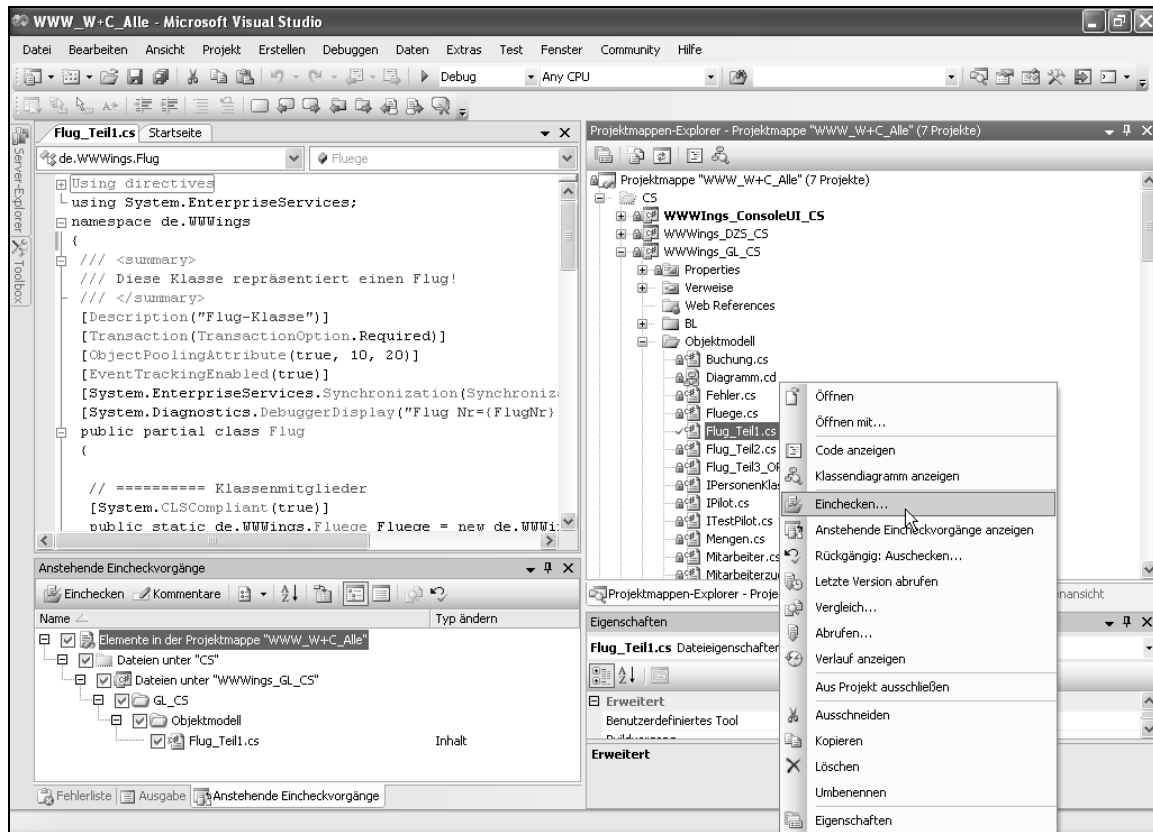


Abbildung 54.4 Einchecken einer geänderten Datei in Visual Studio

## Quellcodeverwaltung mit Subversion

Subversion ist ein Open Source-Quellcodeverwaltungssystem, welches mit dem Ziel ins Leben gerufen wurde, die konzeptuellen Schwächen des Concurrent Versions System (CVS) [CVS01] zu beseitigen. Subversion erreichte im Jahr 2004 die Versionsnummer 1.0.

Anders als CVS unterstützt Subversion auch die Versionierung auf Verzeichnisebene und somit das Umbenennen und Verschieben von Dateien und Verzeichnissen unter Beibehaltung ihrer Versionsgeschichte. VSS-überlegen ist Subversion auch im Hinblick auf die Transaktionsunterstützung. Subversion achtet nicht nur beim Einchecken darauf, dass auf dem Server ein konsistenter Zustand verbleibt, sondern auch beim Auschecken darauf, dass der Entwickler eine komplette Arbeitskopie erhält und somit nicht an fehlenden Dateien verzweifeln muss.

### HINWEIS

Die Firma des Autors verwendete bisher *Subversion* zur Versionsverwaltung und plant nun den Umstieg auf Visual Studio Team System Version Control.

## Server und verschiedene Clients

Subversion gibt es für eine Vielzahl von Linux- und Unix-Varianten sowie auf Windows und Mac OS. Bei den Clients hat man eine große Auswahl. Ein Kommandozeilencient wird mit dem Server [SVN01] mitgeliefert. Die Subversion-Produktreferenz [SVN03] enthält zahlreiche GUI-Clients (z.B. *RapidSVN* [SVN02]), eine Windows-Explorer-Erweiterung (*TortoiseSVN* [SVN06]). Für Visual Studio .NET 2002/2003 sowie Visual Studio Versionen 2005/2008 existiert das Plugin *AnkhSvn* [SVN07]. Weitere Plugins gibt es für Eclipse [SVN05] und Emacs [SVN04]. Durch *sourcecross* [SCS01] erhält Subversion auch das SCC-API, das eine Verbindung zu allen Microsoft Source Control Interface (MSSCCI)-fähigen Werkzeugen ermöglicht. Da Microsoft ein Non-Disclosure-Agreement (NDA) für sein SCCI-API verlangt, ist sourcecross zwar Freeware, aber keine Open Source.

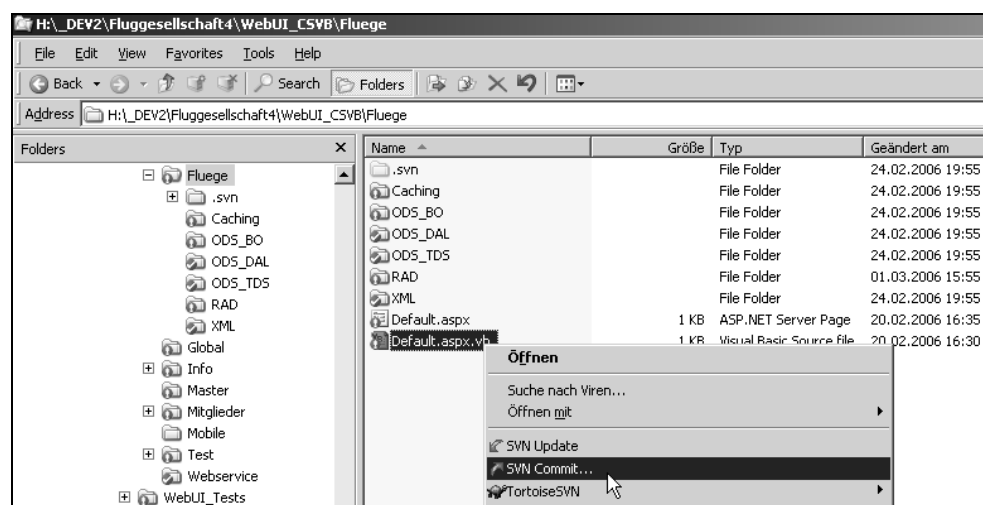


Abbildung 54.5 TortoiseSVN integriert einen Subversion-Client direkt in den Windows-Explorer

## Betriebsarten

Subversion kennt drei unterschiedliche Betriebsarten: Zum einen kann es lokal mit einer reinen Clientinstallation benutzt werden. Dies kann für einzelne Anwender nützlich sein, die keine gemeinschaftliche Entwicklung betreiben wollen. Im Mehrbenutzermodus startet der Server durch eine Exe-Datei und ist über einen frei wählbaren TCP-Port erreichbar. Dabei kann der Nutzer die TCP/IP-Verbindung auch mittels SSH tunneln, um eine verschlüsselte Übertragung zu erzielen. Der Betrieb als Windows-Dienst ist noch nicht vorgesehen bzw. nur über entsprechende Zusatzwerkzeuge möglich. Als dritte Variante erlaubt Subversion zusätzlich den Betrieb mittels des WebDAV-Moduls (Web-based Distributed Authoring and Versioning) für den Apache Webserver. Diese Betriebsart eignet sich vorrangig, wenn ein Zugang zum Server nur über HTTP möglich ist. Wird der Apache mittels SSL-Verbindung betrieben, so kann auch die Kommunikation verschlüsselt erfolgen.

Die Möglichkeiten zur Rechtevergabe unterscheiden sich je nach Betriebsart. Im Einzelplatzmodus wirken die Dateisystemrechte von Windows, die allerdings nur den Zugriff auf die Datenbank beschränken können und somit für ein komplettes Projektarchiv bindend sind. Im TCP-Modus ist die Rechtevergabe (Lese- und Schreibrecht) ebenfalls nur auf Projektarchieebene möglich, im WebDAV-Modus immerhin auf Verzeichnisebene.

Subversion verarbeitet Binärdateien intern genauso wie Textdateien; dies bedeutet, dass der Server für eine neue Revision einer Binärdatei nicht die gesamte Datei ablegt, sondern nur die Unterschiede zwischen den Versionen speichert, um damit Speicherplatz zu sparen. Im Performanzvorteil gegenüber Perforce ist Subversion auch beim Check-In: Da Subversion im Gegensatz zu Perforce Verwaltungsdaten in einem Verzeichnis *.svn* auch bei der lokalen Arbeitskopie ablegt, ist der Datenverkehr zum Server geringer.

Subversion verwendet einen ausgezeichneten Mechanismus für die Entwicklungsverzweigung (Branching). Die Datei bzw. das Verzeichnis, für welche(s) der Entwickler einen neuen Entwicklungszweig einrichten will, wird nicht, wie bei anderen Systemen üblich, an eine neue Stelle im Projektarchiv kopiert, sondern es wird lediglich ein neuer Verweis auf die Datei bzw. das Verzeichnis angelegt. Aus der daraus resultierenden Vermeidung redundanter Daten ergibt sich eine große Platzersparnis. Außerdem nimmt die Durchführung einer solchen Aktion sehr viel weniger Zeit in Anspruch als ein Kopiervorgang. Insbesondere bei der Verzweigung von großen Projekten entsteht hier ein immenser Vorteil in Bezug auf Speicherplatz und Rechenzeit. Dies gilt auch im Hinblick auf anzufertigende Sicherungskopien des Projektarchivs. Neben dem einfachen Tagging-Mechanismus der anderen Produkte bietet Subversion auch die Möglichkeit, beliebige Metadaten in Form von Attribut-Wert-Paaren zu Verzeichnissen und Dateien abzulegen.

## Verwaltungsdaten

SVN speichert auf den Clients in jedem ausgecheckten Verzeichnis Verwaltungsdaten in einem Unterordner *.svn*. Da es in ASP.NET-Webprojekten unter Visual Studio .NET 2002/2003 Probleme mit Verzeichnissen gab, die mit einem Punkt beginnen, existiert die Möglichkeit, optional *\_svn* als Verzeichnisnamen zu verwenden. Das Verwaltungsverzeichnis ist in jedem Fall ein verstecktes Dateisystemverzeichnis.

## Integration in Visual Studio (AnkhSVN)

*AnkhSVN* ist ein Open Source-Add-In für Visual Studio, das die Entwicklungsumgebung zu einem Client für Subversion macht. Ab Version 2.0 ist AnkhSVN ein SCC-Client und nutzt daher auch die eingebauten Funktionen von Visual Studio.

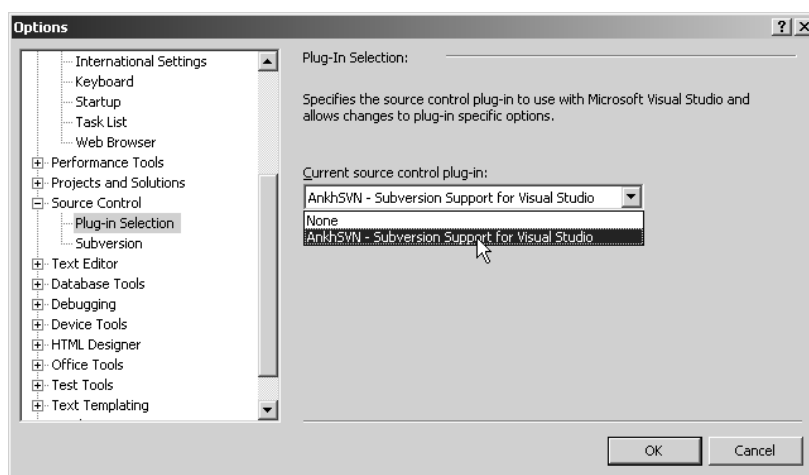


Abbildung 54.6 Aktivieren des AnkhSCN-SCC-Plug-Ins in Visual Studio





Abbildung 54.7 Integration von AnkhSVN in den Projektmappen-Explorer

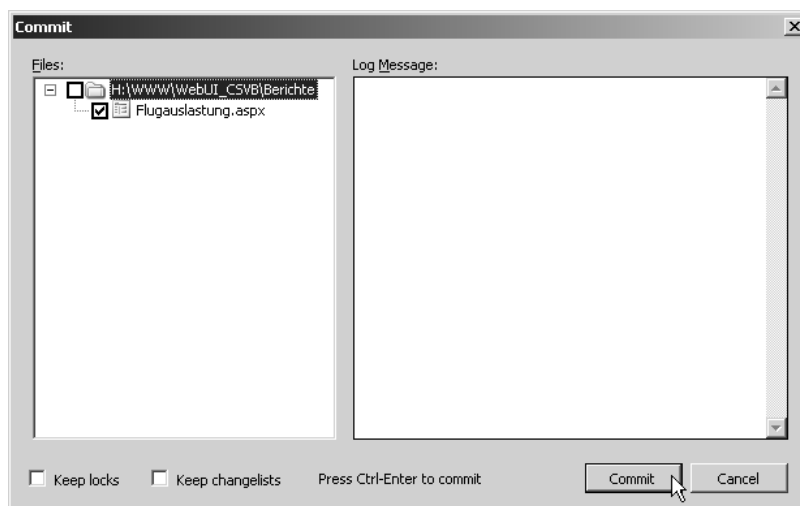


Abbildung 54.8 Dialog beim Einchecken von Dateien

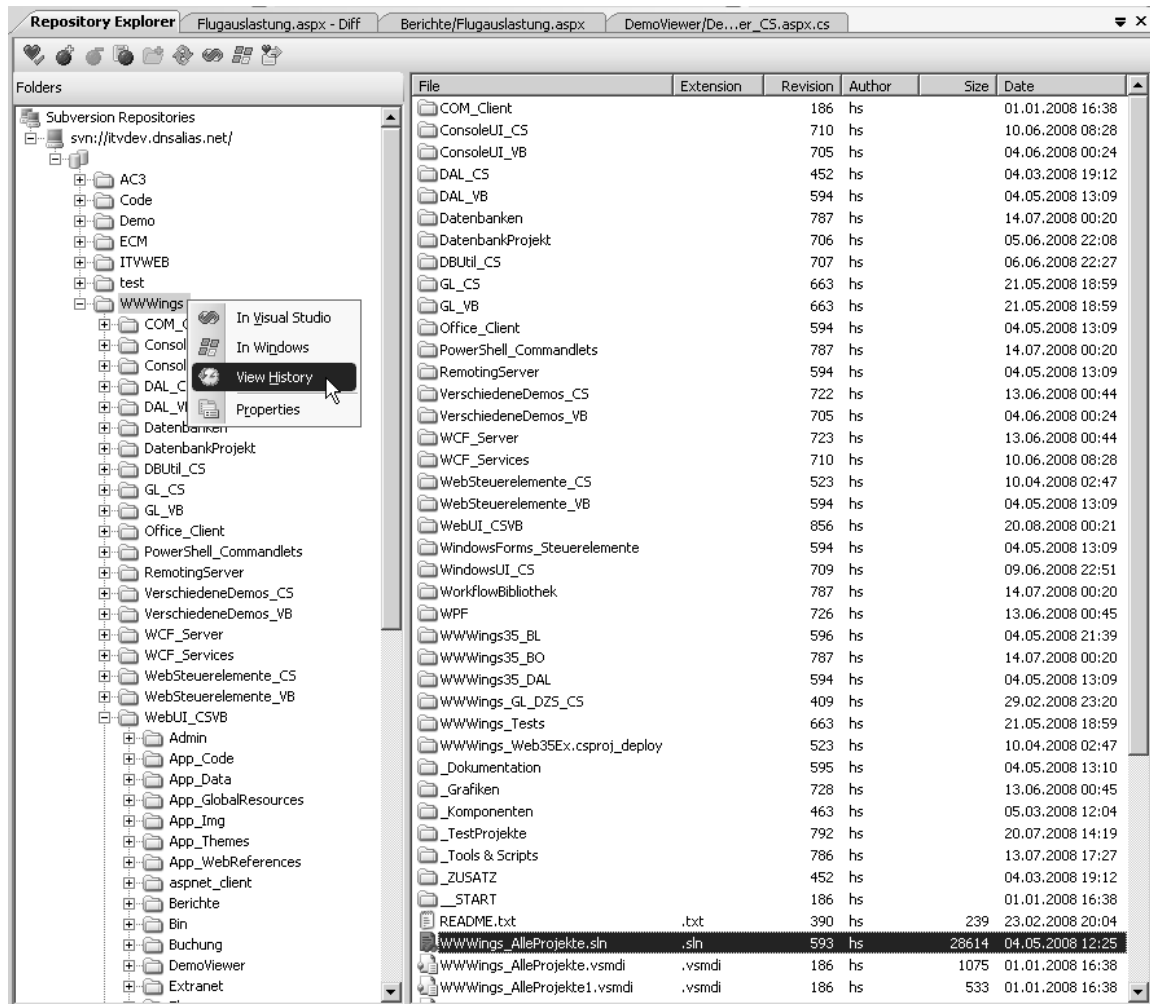


Abbildung 54.9 Repository Explorer