

Versionsverwaltung mit DotSVN

Subversion mit .NET

Im Bereich Versionsverwaltung von Quellcode war CVS lange Zeit der Standard. Subversion ist die Weiterentwicklung von CVS. Und bald wird DotSVN die Arbeit mit Subversion unter .NET ermöglichen. dotnetpro gibt einen Überblick über den Stand der Dinge.

Viele Entwickler kennen das Quellcodemanagement über Subversion, kurz SVN. Hierbei handelt es sich um ein von CollabNet initiiertes Open-Source-Projekt [1] mit dem Ziel, den Vorgänger CVS [2] weiterzuentwickeln. Charmant an SVN sind Features wie Tagging, Branching sowie eine alltagstaugliche Mergefunktion zur Auflösung von Konflikten. Diese Funktionen sucht man etwa bei Microsoft Source Safe bislang vergeblich. Mittlerweile gibt es eine große Anzahl von Tools und Add-ins für Subversion. Für den .NET-Entwickler sind im Tagesgeschäft folgende interessant:

- *TortoiseSVN*. Eine Namespace-Erweiterung für den Windows Explorer. Hierdurch kann der Explorer als SVN-Client benutzt werden.
- *Ankh*. Ein Visual-Studio-Add-in, das sich ähnlich wie MS Source Safe in die Entwicklungsumgebung integriert.
- *RapidSVN*. Ein vollständiger SVN-Client in Form einer eigenständigen Applikation.

Auf einen Blick

Autor



Marcus Peters entwickelt Software für die .NET-Plattform seit der Beta-Phase des Frameworks. Er ist Software Engineer bei der adesso AG in Dortmund und über mpeters@adesso.de zu erreichen.

dotnetpro.code
A0803DotSVN

Inhalt

- ▶ Überblick über das SVN-Projekt.
- ▶ Tools und Add-ins rund um SVN.
- ▶ Mit DotSVN lesend auf SVN-Repositories zugreifen.

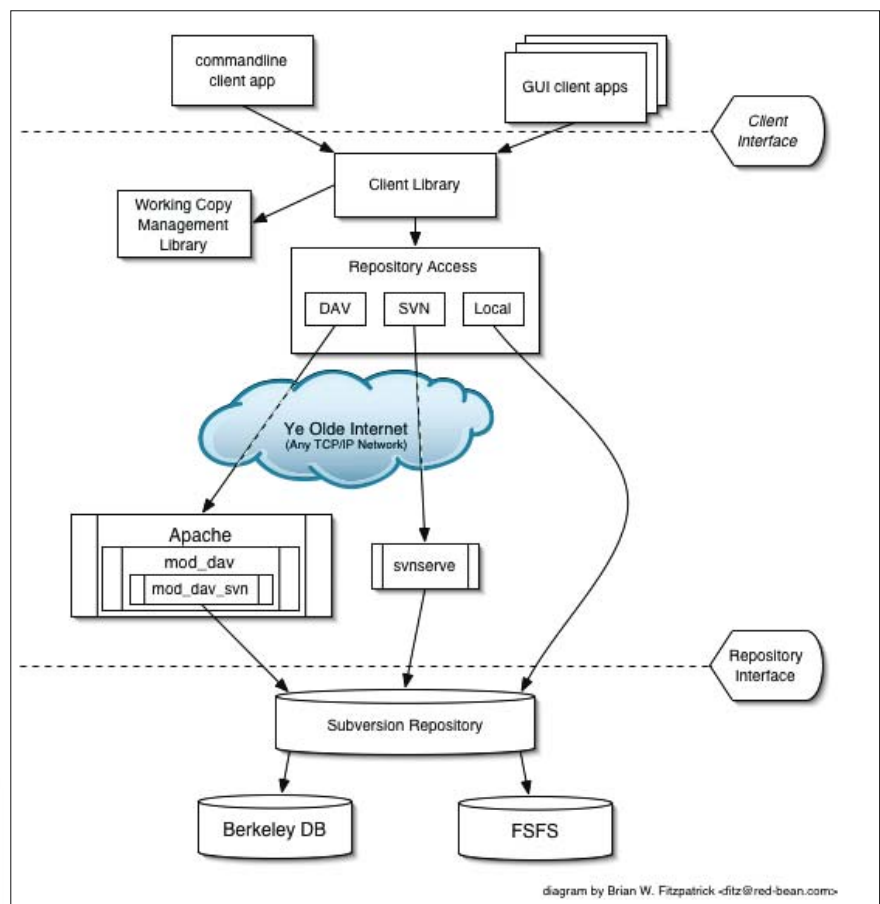


Abbildung 1 Subversion-Architektur.

Prinzipiell kommt SVN ohne UI-Komponenten daher und wurde in einer Schichtenarchitektur umgesetzt. Hier können Anwendungsentwickler auf verschiedenen Ebenen ansetzen, um spezifische Lösungen auf dieser Architektur zu entwickeln. Abbildung 1 zeigt eine Architektur-skizze aus dem freien Buch zu SVN [3].

Hier zunächst ein kurzer Überblick über die SVN-Komponenten:

- *Client Library*. Übernimmt die typischen Clientaufgaben für das Management der Arbeitskopie des Repositories wie Update vom Repository in ein lokales Arbeitsverzeichnis, Com-

mit der veränderten Dateien in das Repository et cetera.

- *Repository Access*. Abstrahiert den Zugriff auf das Repository über verschiedene Protokolle mit der Intention, das Repository lokal beziehungsweise via Netzwerk zu erreichen. Folgende Zugriffspfade sind hier berücksichtigt:

- Via WebDav, bisher vornehmlich über eine Apache2-Installation.
- Via SVN-Protokoll für den Zugriff über das SVN-Servermodul *svnservice*.
- Lokaler Zugriff auf das Repository über ein Dateisystem.

- *Apache*. Ist der vom SVN-Projekt präferierte Host für den Zugriff auf ein Repository via WebDav.
- *Svnserve*. SVN-Servermodul für den Betrieb via `svn://`-Protokoll. Eine Version für Windows ist verfügbar und kann als Windows-Service betrieben werden [4].
- *Subversion Repository*. Das eigentliche Repository ist über ein Schnittstellenkonzept abstrahiert und kann in einer Berkeley DB oder einem Dateisystem betrieben werden.

DotSVN

Die oben genannten Tools TortoiseSVN sowie Ankh sind Open-Source-Projekte für die Integration von SVN in Windows beziehungsweise Visual Studio. Darüber hinaus existieren verschiedenste Projekte für native Implementierungen einer SVN-Client-Library [5]. Für das .NET Framework wurde das Projekt DotSVN von George Chiramattel ins Leben gerufen [6].

Die eigentliche Intention von DotSVN ist die Unterstützung des Projektes nTrac, das einen Zugriff auf SVN-Repositories benötigt. Bei nTrac handelt es sich um die Portierung eines Bugtracking-Systems auf das .NET Framework. Erklärtes Ziel von DotSVN ist es, absolute Kompatibilität zu wahren, wobei über DotSVN angelegte Repositories ebenfalls mit den gängigen Tools zugreifbar sein sollen. Gegenwärtig gibt es Implementierungen für die Komponenten „Repository Access“ und „SVN Repository“, wobei zurzeit nur der lokale Zugriff auf ein Dateisystem-Repository unterstützt wird.

Der DotSVN-Quellcode ist als Visual-Studio-Projekt verfügbar. Die Projektmappe beinhaltet neben den Quellen

Listing 1

Ein Repository öffnen.

```
// Open a connection to the repository
ISVNRepository repository = SVNRepositoryFactory.Create(new SVNURL(repositoryPath));
repository.OpenRepository();
```

Listing 2

Auf ein SVN-Repository mit DotSVN zugreifen.

```
/// <summary>
/// Browse the contents of a repository and adds the contents to the tree view
/// </summary>
/// <param name="revision">Revision of the repository to browse</param>
private void BrowseRepository(long revision) {
    try {
        // Get the contents of the repository and sort them
        ICollection<SVNDirEntry> dirEntries = repository.GetDir(string.Empty, revision, null);
        SortEntries(dirEntries);
        // Add the root entry to the repository browser's tree view
        txtURL.Text = repository.GetRepositoryRoot(true).ToString();
        TreeListViewItem root = repositoryTreeView.Items.Add(txtURL.Text, 0);
        // Add the other entries one by one to the tree view
        foreach (SVNDirEntry dirEntry in dirEntries) {
            AddItem(dirEntry, root);
        }
    }
    catch (Exception ex) {
        MessageBox.Show("Error occurred while connecting to the repository\n" + ex, "Exception",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

und NUnit-Testprojekten einige Beispielprojekte, die den gegenwärtigen Entwicklungsstand widerspiegeln, siehe Abbildung 2. Der Zugriff auf ein Repository erfolgt über die in Listing 1 beschriebene Methode. Sie liefert ein *Repository*-Objekt zurück, auf dem anschließend diverse Aktionen ausgeführt werden können. Lis-

ting 2 zeigt beispielhaft das Auslesen der Repository-Struktur in eine Baumansicht.

Fazit

Der gegenwärtige Stand erlaubt einen Read-only-Zugriff auf lokale SVN-Repositories mit einer Implementierung einer entsprechenden, SVN-kompatiblen .NET-Zugriffsschicht. Allerdings fehlen noch wichtige Dinge für die Arbeit mit den Repository-Inhalten (Update, Commit et cetera) sowie für das Management von Arbeitsverzeichnissen. |||||

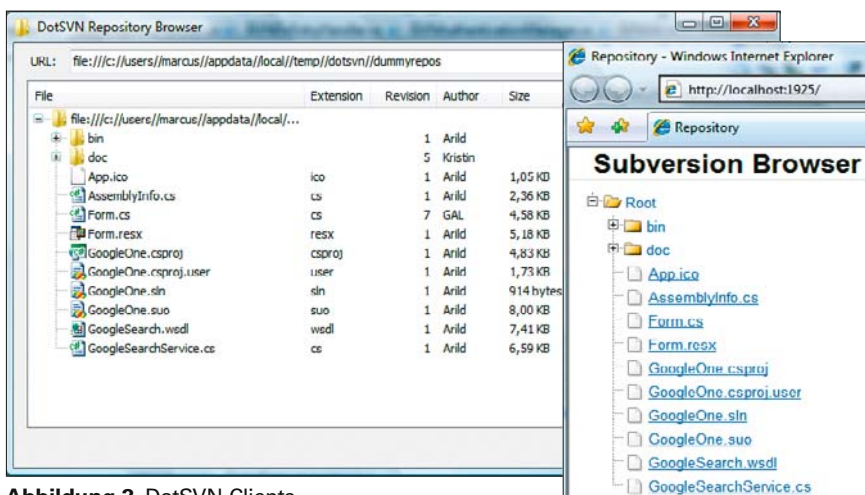


Abbildung 2 DotSVN-Clients.

- [1] Subversion Projekt-Homepage, <http://subversion.tigris.org>
- [2] Überblick über CVS in der Wikipedia, <http://de.wikipedia.org/wiki/CVS>
- [3] SVN-Buch, svnbook.red-bean.com
- [4] SVNServe als Windows-Dienst, [dnpLink SL0803DotSVN1](http://www.dnpLink.com/SL0803DotSVN1)
- [5] SVN Client-Libraries, SvnCpp (C++) <http://rapidsvn.tigris.org/svncpp.html> SVNKit (Java) <http://svnkit.com/>
- [6] DotSVN Homepage, www.dotsvn.net