



# **Development Environment and Software Configuration Management (SCM) Interoperability Lab: CollabNet Subversion, AnkhSVN, and Microsoft Visual Studio**

Published: March 2009

## **Abstract**

---

For enterprises and developers, there are many possible choices for their IDE and SCM systems, which results in often having to deal with multiple heterogeneous IDE and SCM systems both within and across project teams. In this lab, we are testing an interoperability scenario between Microsoft Visual Studio 2008 IDE and CollabNet Subversion SCM through the use of AnkhSVN, a Subversion source control provider for Visual Studio.

## Table of Contents

Development Environment and Software Configuration Management (SCM) Interoperability Lab: CollabNet Subversion, AnkhSVN, and Microsoft Visual Studio .....	1
Abstract.....	1
Introduction.....	3
Lab Scenario.....	4
Lab Products and Solutions.....	5
AnkhSVN Plug-in for Visual Studio 2005 and 2008 .....	5
CollabNet Subversion .....	5
Microsoft Visual Studio Team System 2008/ Microsoft Visual Studio 2008 Professional Edition .....	5
Microsoft Team Foundation Server.....	5
Lab Execution .....	6
Some terminology.....	6
Initial configuration.....	8
Connecting to Apache Foundation's SVN repository using the Repository Explorer.....	9
Conclusions.....	13
Links .....	14
Appendix: The Interop Vendor Alliance .....	17

# Introduction

---

Whether in teams or working individually, developers usually rely on a version control system or software configuration management (SCM) solution to store and manage their source code. Operations like check-in/check-out, versioning, branching/merging, code repository sharing, and remote & distributed access are part of the overall application lifecycle management process. Modern integrated development environments (IDE), such as Microsoft Visual Studio, allow the use of various SCM systems to cope with heterogeneous environments.

For enterprises and developers, there are many possible choices for their IDE and SCM systems, which often results in having to deal with multiple IDE and SCM systems both within and across project teams.

In this lab, we are testing an interoperability scenario between Microsoft Visual Studio 2008 IDE and CollabNet Subversion SCM through the use of AnkhSVN, a Subversion source control provider for Visual Studio.

# Lab Scenario

The lab is based on a real use case scenario, which is Microsoft’s code contribution to the Stonehenge project. Stonehenge is an Apache Incubator project to build a set of example applications for SOA that spans multiple languages and platforms while demonstrating best practices and interoperability. Microsoft, along with other companies like WSO2, have contributed code for a web-services based sample application (called StockTrader) to the Stonehenge open source efforts. Microsoft’s code can be found here: <https://svn.apache.org/repos/asf/incubator/stonehenge>.

In this lab we are going to document our experience in using two source control systems to manage the StockTrader/Stonehenge source code. Internally Microsoft uses Visual Studio Team Foundation Server to manage the StockTrader/Stonehenge projects, solutions and contribution from the developers that are working on the project, but the Apache Foundation uses CollabNet Subversion as its SCM system. This gave us the opportunity to test the interoperability between Microsoft Visual Studio and Subversion using AnkhSVN on a real-world scenario.

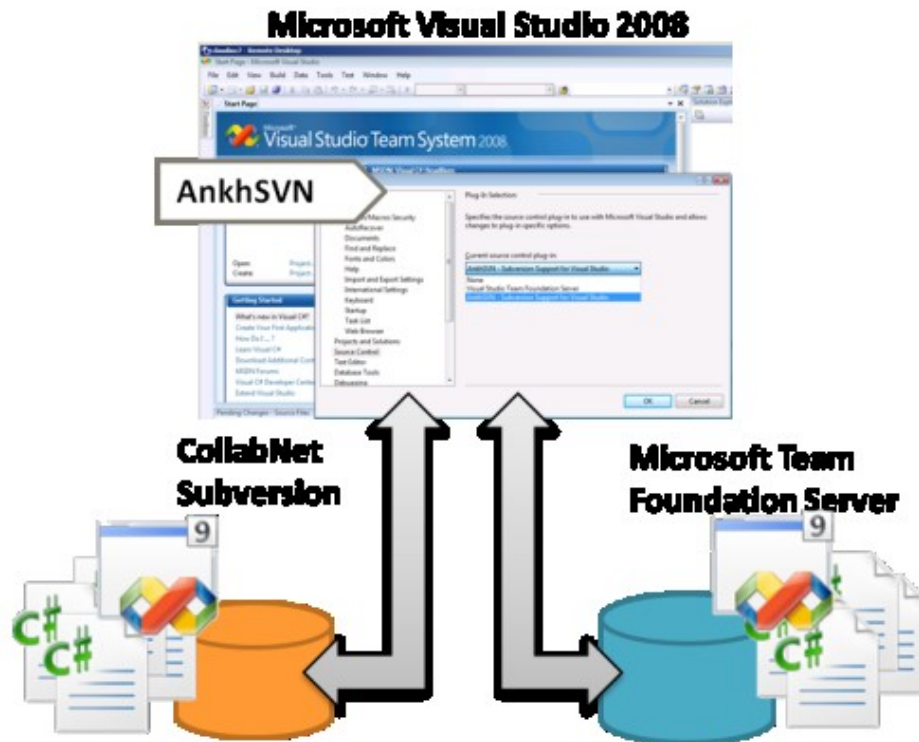


Figure 1: AnkhSVN source control provider for Visual Studio adds support for CollabNet Subversion

## Lab Products and Solutions

---

### **AnkhSVN Plug-in for Visual Studio 2005 and 2008**

AnkhSVN is a Subversion Source Control Provider for Visual Studio. This software allows you to perform most common version control operations directly from inside the Microsoft Visual Studio IDE, like viewing the status of your source code, updating your Subversion working copy and committing changes or browsing your repository. AnkhSVN utilizes the interfaces available via Visual Studio Extensibility (<http://msdn.microsoft.com/en-us/vsx/>).

More: <http://ankhsvn.open.collab.net/>

### **CollabNet Subversion**

Subversion is an open source version control and Software Configuration Management (SCM), initiated by CollabNet

More: <http://www.open.collab.net/products/subversion/>

### **Microsoft Visual Studio Team System 2008/ Microsoft Visual Studio 2008 Professional Edition**

Visual Studio is Microsoft's development environment for Microsoft Office, Windows Mobile, Windows Server and Web-based applications.

In this lab we used Visual Studio Team System 2008 Team Suite.

<http://www.microsoft.com/visualstudio/>

### **Microsoft Team Foundation Server**

Microsoft Visual Studio Team System 2008 Team Foundation Server is a team collaboration platform that combines team portal, version control, work-item tracking, build management, process guidance, and business intelligence into a unified server.

<http://msdn.microsoft.com/en-us/teamsystem/dd408382.aspx>

# Lab Execution

---

## Some terminology

First we need to become familiar with some basic Subversion terminology, and distinguish between a Visual Studio (VS) project and a Subversion (SVN) project.

- A SVN project is a folder in a Subversion repository.
- A VS project is a group of files that produce a single binary target (dll, exe, lib).
- A VS studio project is analogous, but not identical to a makefile.

A VS solution is a grouping of related VS projects, it has no concept of a build target. VS creates a solution for every project, but you can also create solutions ad-hoc, and add projects to them. Frequently, a VS project belongs to two or more VS solutions. Further, developers frequently create VS solutions locally on their machines as part of the development and debugging process, but these solutions are "throw aways" which don't get checked in.

A SVN project typically has sub-folders for 'trunk', 'branches', and 'tags'... For instance, if we have the following working folder structure:

```
C:\MyProjects\LearnSvn\MyLibrary
  MyLibrary.sln
  MyLibrary.csproj
  MyApplicationClass.cs
```

In SVN, the folder structure would look like this:

```
/LearnSvn/MyLibrary/
  trunk/
    MyLibrary.sln
    MyLibrary.csproj
    MyApplicationClass.cs
```

If we branch the solution from within VS, call the branch 'development', and commit it, this is what SVN looks like:

```
/LearnSvn/MyLibrary/
  trunk/
    MyLibrary.sln
    MyLibrary.csproj
    MyApplicationClass.cs
  branches/development/
    MyLibrary.sln
    MyLibrary.csproj
    MyApplicationClass.cs
```

Switching between 'trunk' and 'development' causes SVN to update the contents of the working folder (ie. MyLibrary) with the appropriate branch. The working folder does not mirror the SVN repository's structure. Instead, the working folder's contents get updated with the contents of either trunk/ or branches/development/, depending on which branch you 'switch to'.

Now suppose we have a test harness (console executable) that exercises MyLibrary.dll. We may want a new solution to host both VS projects, and a working folder structure that looks like this:

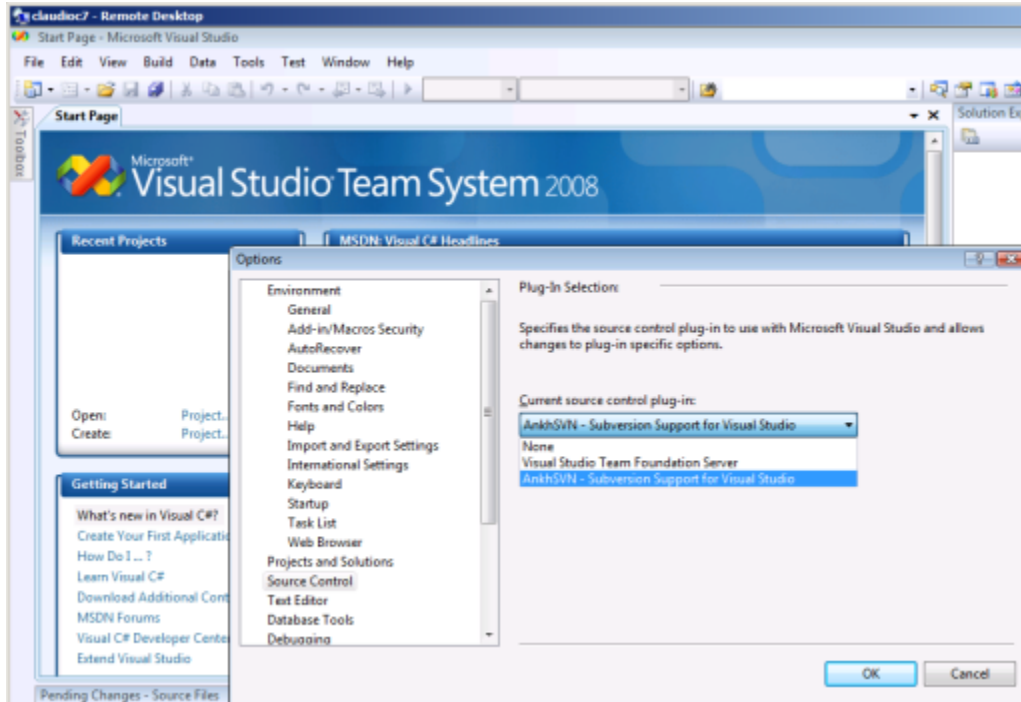
```
C:\MyProjects\LearnSvn\  
  LearnSvn.sln  
  MyLibraryTestHarness\  
    MyLibraryTestHarness.sln  
    MyLibraryTestHarness.csproj  
    HarnessProgram.cs  
  MyLibrary\  
    MyLibrary.sln  
    MyLibrary.csproj  
    MyApplicationClass.cs
```

In the above example, MyLibrary.csproj is included in two different solutions; MyLibrary.sln and LearnSvn.sln. Likewise MyLibraryTestHarness is included in two different solutions.

After we became familiar with Subversion terminology we were able to structure the StockTrader/Stonehenge projects and solutions in a way that works well with SVN.

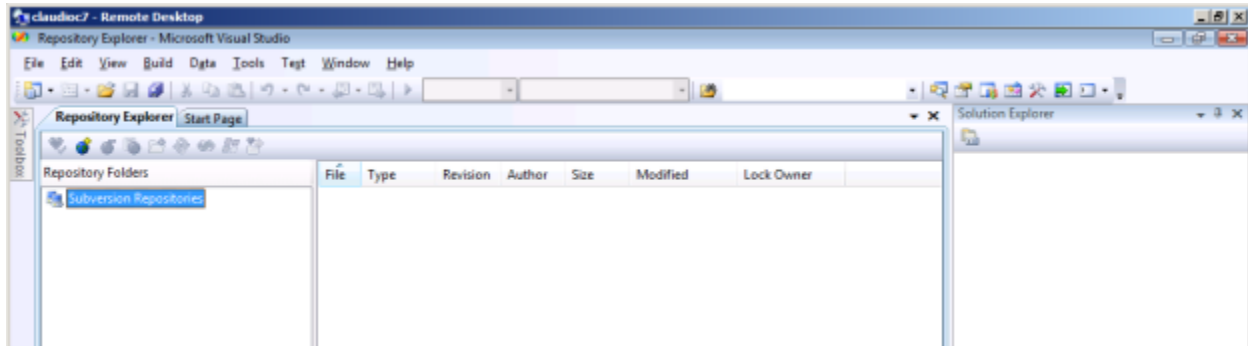
## Initial configuration

After AnkhSVN is installed on a machine that runs Visual Studio 2005 or 2008, it can be selected as the source control plug-in for projects and solutions within Visual Studio.



## Connecting to Apache Foundation's SVN repository using the Repository Explorer

The next step was to connect to Apache Foundation's SVN repository using the Repository Explorer provided by the AnkhSVN plug-in



We added the URL of the root repository:

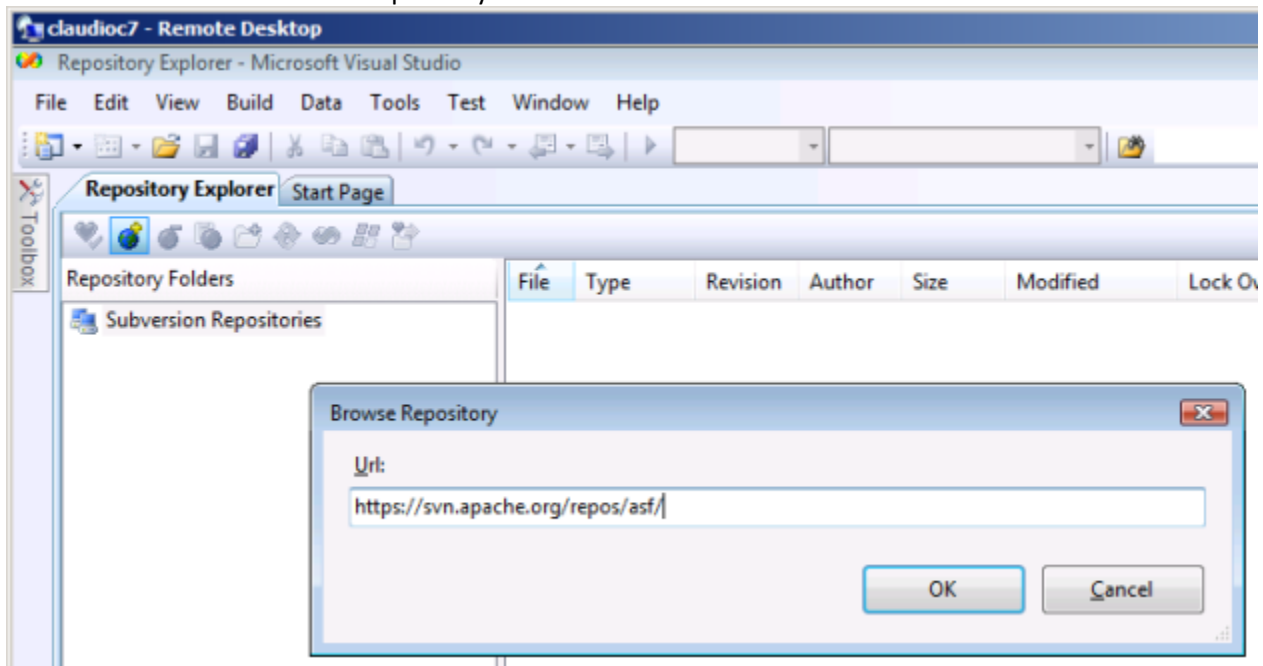


Figure 2. Adding the URL of the root repository

After the root URL was set, we were able to navigate through the Apache SVN repository to find the location that was created for the StockTrader/Stonehenge project:

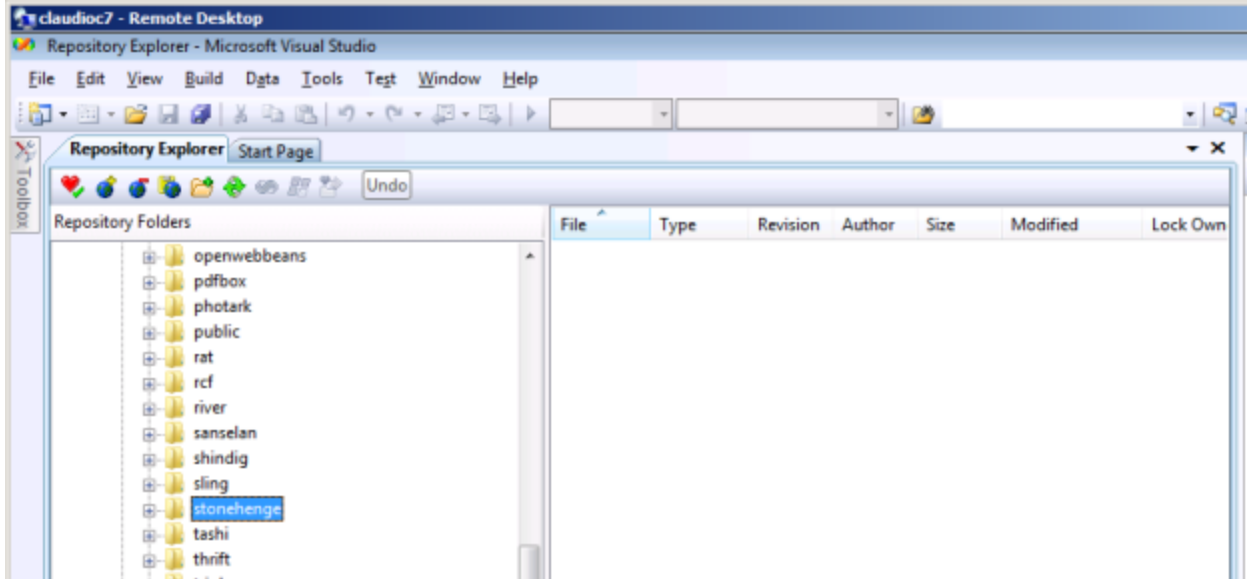


Figure 3. Navigating to the directory set for the StockTrader/Stonehenge project

We then opened one of the StockTrader/Stonehenge solutions:

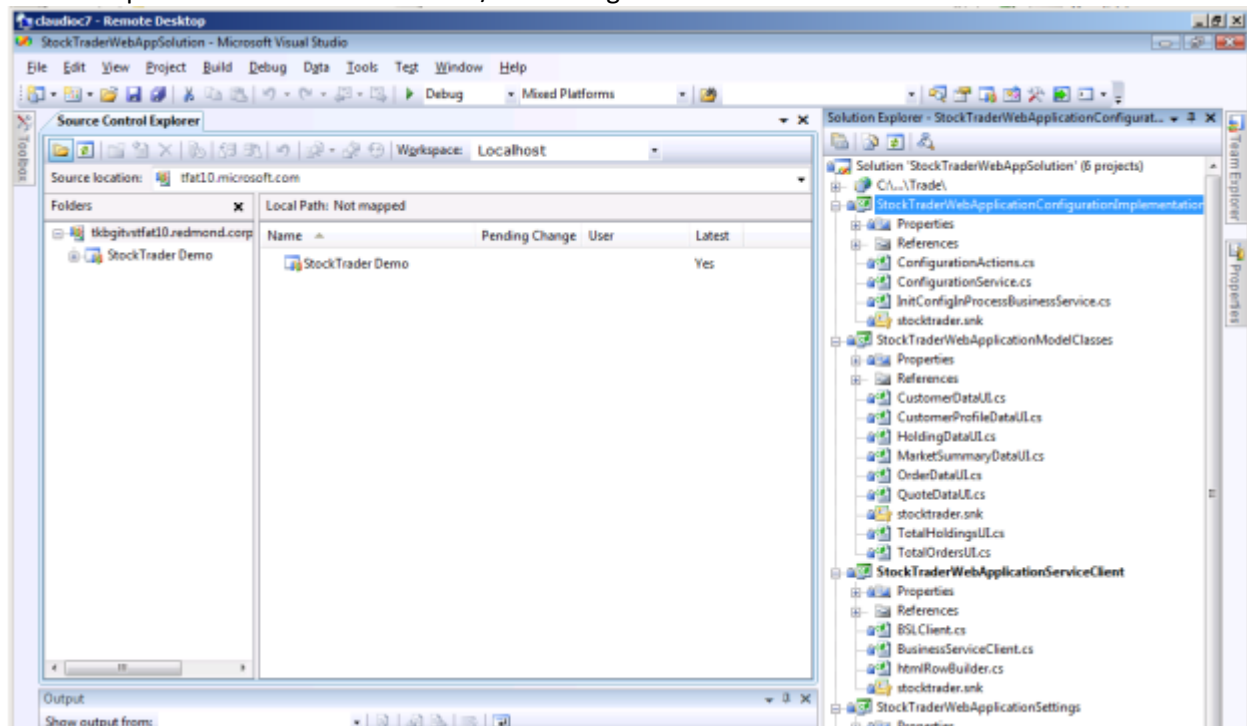


Figure 4. Opening one of the StockTrader/Stonehenge solutions

In Microsoft StockTrader/Stonehenge, source code is managed using TFS. The first step was to unbind the project from TFS so that Subversion can be selected for source control and used to submit the code to the Apache repository.

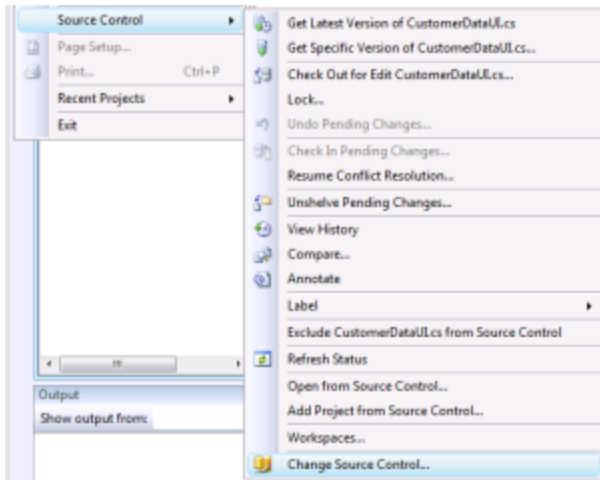


Figure 5: Unbinding and change to a new source control

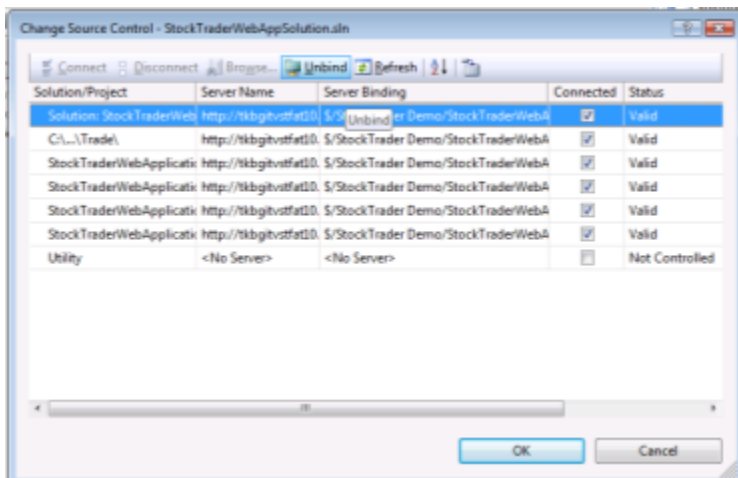


Figure 6: Unbinding and change to a new source control, cont.

After the solution was unbound we could select AnkhSVN as the plug-in for source control

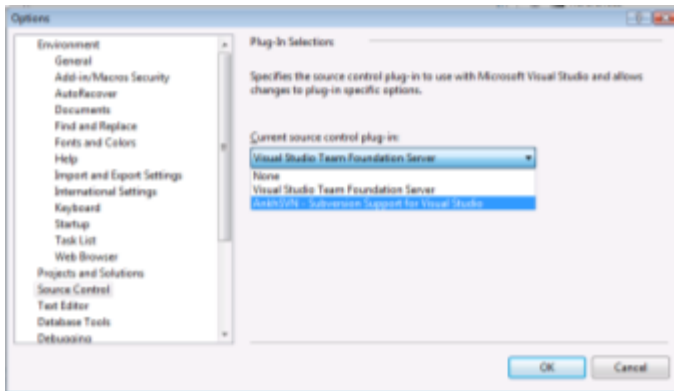


Figure 7: Unbinding and change to a new source control, end.

The project is then committed to the subversion repository:

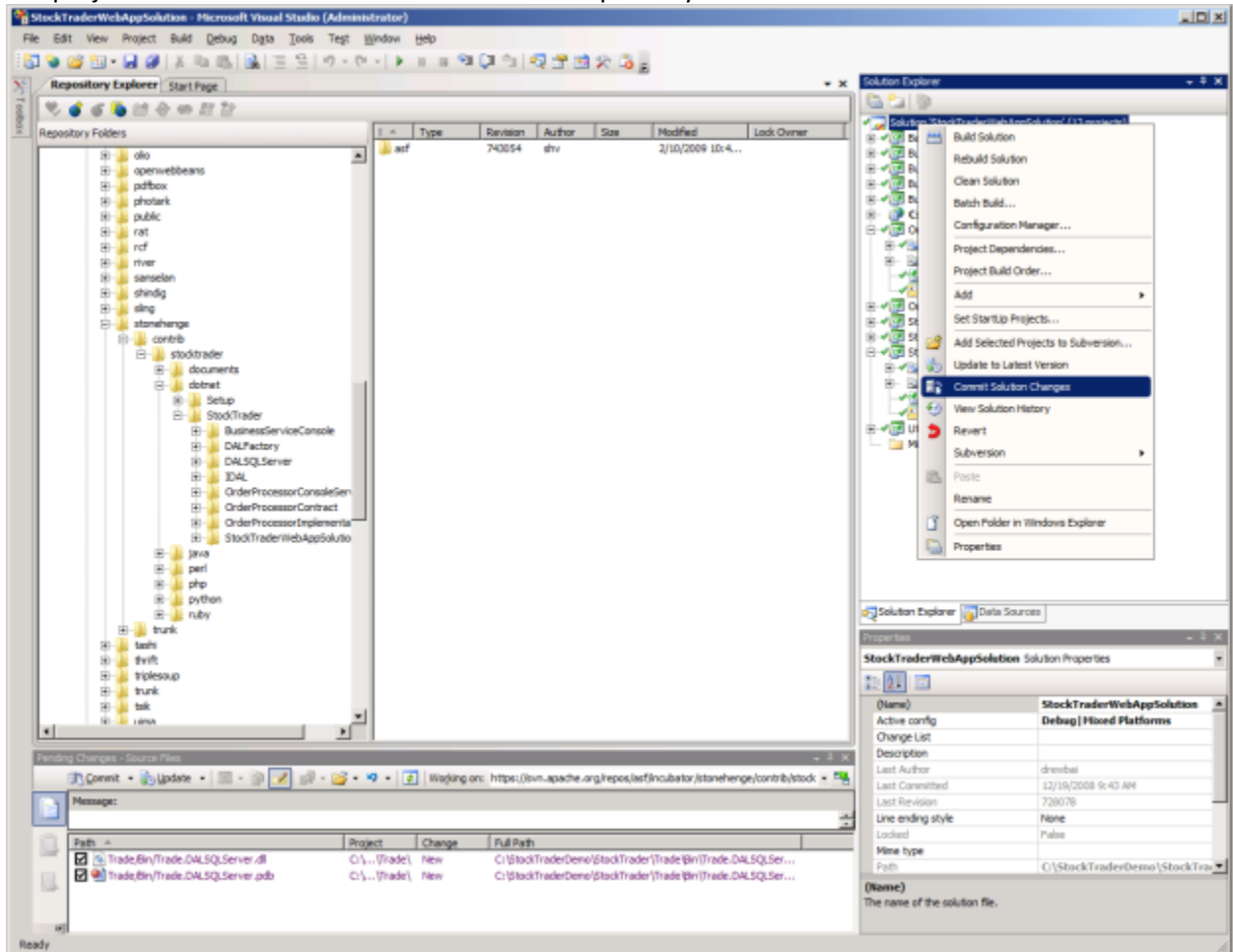


Figure 8: committing the project to the subversion repository

## Conclusions

---

Based on a real use case scenario, this lab shows how developers can work with multiple source control systems from within Visual Studio.

The lab highlighted how AnkhSVN enables interoperability between Visual Studio and a Subversion source control system. The AnkhSVN Subversion Source Control Provider for Visual Studio also demonstrates how extensions can be added to Visual Studio using the extensibility model provided by Microsoft.

# Links

---

For more information visit:

<http://interopvendoralliance.org/>

Stonehenge:

- <http://incubator.apache.org/stonehenge/index.html>
- <http://wiki.apache.org/incubator/StonehengeProposal>

AnkhSVN:

- <http://ankhsvn.open.collab.net/>

CollabNet Subversion:

- <http://www.open.collab.net/products/subversion/>

## Appendix: The Interop Vendor Alliance

---

The Interop Vendor Alliance (IVA) is a cross-industry group of global software and hardware vendors that share the common goal of making mixed technologies work better together for their customers. When people work together, the solutions they design are more likely to work together as well. The IVA provides a venue for its members to share customer feedback and to tackle common interoperability challenges. Through member meetings, hands-on labs, and online forums, IVA members work together to solve common interoperability challenges while improving the effectiveness of their own solutions. Scenarios addressed by the IVA include system management, virtualization, identity management, data integration, storage management, interoperability of developer tools, in addition to portal integration.

The IVA supports:

- Vendor collaboration to foster interoperability
- Scenario-based testing for interoperability
- Communicating with customers about interoperability solutions

## Table of Figures

Figure 1: AnkhSVN source control provider for Visual Studio adds support for Collabnet Subversion .....	4
Figure 2. Adding the URL of the root repository .....	9
Figure 3. Navigating to the directory set for the StockTrader/Stonehenge project .....	10
Figure 4. Opening one of the StockTrader/Stonehenge solutions .....	10
Figure 5: Unbinding and change to a new source control .....	11
Figure 6: Unbinding and change to a new source control, cont.....	11
Figure 7: Unbinding and change to a new source control, end. ....	11
Figure 8: committing the project to the subversion repository.....	12

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

This documentation is licensed to you under the Creative Commons Attribution-Share Alike 3.0 United States License.



To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA. When using this documentation, provide the following attribution: Development environment and source control interoperability lab: CollabNet Subversion, AnkhSVN, and Microsoft Visual Studio.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2008 Microsoft Corporation.

Microsoft, SharePoint, and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.