

.NET Web Services  
Version Control System  
Interim Report

Dafydd Vaughan  
327039  
dafydd@dafyddvaughan.co.uk

February 9, 2007

**Abstract**

Over the past three months, significant progress has been made in developing a version control system that uses Microsoft .NET Framework Web Services. This Interim Report looks at the progress made so far, the problems encountered, their solutions and provides an updated timeline for the project.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Outline . . . . .	3
<b>2</b>	<b>Project Progress</b>	<b>4</b>
2.1	SVN Research . . . . .	4
2.2	Web Service Examples . . . . .	4
<b>3</b>	<b>Problems Faced</b>	<b>6</b>
3.1	Sending Files . . . . .	6
3.2	Merging Files . . . . .	6
<b>4</b>	<b>Changes Since Initial Document</b>	<b>8</b>
4.1	.NET Framework 3.0 . . . . .	8
4.2	SQL Server . . . . .	8
4.3	Client Responsibility . . . . .	9
4.4	System Features . . . . .	9
<b>5</b>	<b>Project Timeline</b>	<b>11</b>
5.1	Milestones . . . . .	11
5.2	Timeline . . . . .	12
	<b>Glossary</b>	<b>13</b>
<b>A</b>	<b>Appendix: SVN Screenshots</b>	<b>14</b>
<b>B</b>	<b>Appendix: C# Application Screenshots</b>	<b>17</b>

# 1 Introduction

Since the Initial Document for this project was completed in October 2006, significant progress has been made in developing the .NET Web Services based version control system. Over the last few months, I have conducted research into other version control systems and experimented with Web Services, C# applications and a new version of the Microsoft .NET Framework. During these experiments I have encountered a number of problems that have had to be overcome and have had to re-evaluate the project milestones and timeline.

This Interim Report outlines the progress achieved so far, the problems faced and the changes that have been made to the project since the writing of the Initial Document.

## 1.1 Outline

- **Section 2 - Project Progress** looks at the progress made with the development of the version control system
- **Section 3 - Problems Encountered** outlines the problems encountered so far and the steps taken to overcome them
- **Section 4 - Changes** details the changes to the project specification outlined in the Initial Document
- **Section 5 - Project Timeline** shows the changes made to the project milestones and timeline
- **Section 6 - Glossary** contains a list of acronyms used throughout the document alongside their meanings

## 2 Project Progress

In the three months since the Initial Document was produced, significant progress has been made into the development of the .NET Web Services based Version Control System. In these three months, I have undertaken research into how an existing version control system operates - through using it in a real-world development process - and produced several Web Services examples showing how they work and how they link with C# applications.

This section details the progress made so far in the development of this project.

### 2.1 SVN Research

To better understand how existing version control systems work, I have made use of one example - Subversion (SVN) - during the production of another project. Over recent months, I have used SVN to aid in the development of a new website for Swansea University's Student Newspaper - The Waterfront. The new website ([waterfrontonline.co.uk](http://waterfrontonline.co.uk)) has been developed using the server side scripting language PHP and deployed on a Linux based web server.

The production of this website involved several developers making changes to code, graphics and the structure of the site simultaneously. This needed to be easily managed to avoid more than one user overwriting another user's changes. As a result, SVN was chosen to act as the version control system. The SVN server (repository) is located on a server running Linux (Ubuntu) with clients running on Ubuntu, Microsoft Windows XP and Microsoft Windows Vista. Communication between the clients and repository is performed using the Secure Shell (SSH) protocol. Screenshots showing SVN in use can be found in the Appendix A.

The use of SVN in a real-world development environment has enabled me to review the features that should be provided by my Version Control System. As a result, a number of changes have been made to the project plan outlined in the Initial Document. The full details of these changes will be outlined in section 4.

### 2.2 Web Service Examples

As part of the original project plan, a milestone to experiment with Web Services was included. The aim of this milestone was to enable me to obtain a greater understanding of Web Services and how they are implemented in the Microsoft .NET Framework.

Since the beginning of the project, I have spent a great deal of time developing example Web Services - each growing in complexity and usefulness. I have also

linked a number of these Web Services with front end C# applications. These applications invoke the Web Services on the remote server. In the following sections, I have detailed a selection of the Web Services I have created.

### 2.2.1 Hello World

This very simple Web Service that, when invoked, returns a string containing the words “Hello World”.

Listing 1: HelloWorld() Web Service XML Response

```
<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://localhost/vcs/Test/HelloWorld">
Hello World
</string>
```

### 2.2.2 String Checker

This Web Service takes a string of characters and checks to see if it matches a string that is hard-coded into the service. If they match, the Web Service returns a boolean value of “true”, otherwise it returns “false”.

### 2.2.3 Number Evaluator

This Web Service takes two numbers and compares them. The service then returns the bigger of the two numbers to the client. If both numbers are identical, the service returns the number.

This Web Service example was implemented using a C# application as the front end. Screen shots showing this example can be found in Appendix B.

### 2.2.4 .NET Framework 3.0

Since the Initial Document was produced, Microsoft has released version 3.0 of its .NET Framework. This update includes several new tools for communicating with Web Services as well as many other additions. To explore these new tools, I produced a number of examples that make use of these updates. The changes in the .NET Framework 3.0 will be discussed later.

## 3 Problems Faced

During the development process so far, I have faced a number of problems that have had to be overcome. In the following sections, I have outlined some of these problems and how I intend to resolve them.

### 3.1 Sending Files

One of the many problems faced so far has been how to send multiple files between the server and client applications. Traditionally, the way to approach this problem would be to send each file individually, however, there are a number of disadvantages to the use of this method.

- **Bandwidth** - all the files being sent across a network (possibly the Internet) while uncompressed could lead to a large bandwidth bill.
- **Time** - the time taken to send each uncompressed file could potentially be quite long. Additionally, the time taken for the Web Service to invoke its methods for each file could be quite large.

In Microsoft's latest version of the .NET Framework - Version 3.0 - they have released a new library called "Packaging" (`System.io.packaging`). This new library enables files to be "packaged" into a compressed ZIP archive while preserving their directory structure.

This new library could help reduce both the bandwidth used and the time taken to send the files between the client and server. Not only can the files be compressed, but multiple files can be sent at the same time by including multiple files in each ZIP archive.

### 3.2 Merging Files

Another problem faced is that of how to merge files should a clash occur. A clash might occur if, for example, a file has been edited by two different people at the same time. Other systems such as SVN warn the second user that a potential problem exists and allows them to download an updated copy of the from the repository.

An alternative method is to attempt to merge the files automatically. However this method could cause further issues if, for example, a change by person B affects the change made by person A.

In the case of this Version Control System, I will be making use of the first method that is employed by SVN. This means that the job of resolving clashes has been passed to the system user.

## 4 Changes Since Initial Document

Since the Initial Document was produced in October 2006, a number of changes have been made to the way the Version Control System will work. In the following sections, I will outline some of these changes. Additionally, in section 4.4, I will summarise the system's features.

### 4.1 .NET Framework 3.0

As mentioned in section 3.1, Microsoft has released a new version of its .NET Framework. This new version - 3.0 - adds many new tools to this programming framework. These additions include the Windows Communication Foundation, Windows Presentation Foundation, Windows Workflow Foundation and Windows CardSpace. The Windows Communication Foundation provides extra features for enabling communication between applications through the use of Web Services.

Of most importance is the new "packaging" library (`System.io.packaging`) included in the Windows Communication Framework. This new library will be used to send files between the server and the client as described in section 3.1.

The server application will package the latest version of the file repository into a compressed ZIP archive when a user "checks out" a copy and the client application will package files to be "committed" back to the server. The use of this library will reduce both the bandwidth used to transmit the files between the server and the client as well as minimise the time taken for the transmission to take place.

### 4.2 SQL Server

A further change in the way the server application will operate is the use of SQL Server. This database server will be used to hold all previous versions of the project repository - making use of 'blob' fields to store the files.

SQL Server has been chosen as it is the recommended database system for use with .NET applications; however other systems such as `mysql` could also be used to make the system available on multiple platforms.

The use of a database system such as SQL Server will also enable some of the optional features mentioned in the Initial Document - such as logging - to be implemented at a later date.

### 4.3 Client Responsibility

An essential feature described in the Initial Document is for the server application to “merge” the changes to files if a clash has occurred. As mentioned in section 3.2, there are a number of methods that can be used to do this. At the time of writing the Initial Document, it was thought that the server application would be made responsible for this, however after conducting research into how other version control systems (like SVN) operate, it has been decided to transfer this responsibility to the user of the system. Details of this issue were covered in section 3.2.

### 4.4 System Features

This section outlines the updated features I will be aiming to implement into the version control system. The features list is split into 2 categories:

- **Essential Features** - functionality that must be included for the project to be considered a success
- **Extra Features** - functionality I would like to have completed but are not essential

#### 4.4.1 Essential Features

The features included within this section are categorised “essential” and need to be completed for the project to be considered a success.

1. A Web Service “server” that:
  - allows a client to see files contained within the project
  - allows a client to “check out” (download) files to the local machine
  - updates revision number on commit (upload)
  - checks version number of clients copy against server copy on commit
  - informs the client application if a clash occurs
2. A Web Service “client” that:
  - downloads a copy of a file to the local machine
  - allows the user to “commit” their copy to the server
  - allows the user to download an updated version of a file should a clash occur

#### 4.4.2 Extra Features

The features included within this section are features that I would like to be completed by the end of the project, however they are not considered essential.

1. A server that maintains previous versions of a project and enables the project to be rolled back
2. A server that maintains a log of all changes for auditing purposes
3. A server that allows files to be “locked” from edit when necessary
4. A server with a web interface for easy management
5. A server and client with in built authentication
6. A Java based client for communicating with the server to test out the new Web Service features of the latest version of the Java platform

## 5 Project Timeline

Since writing the Initial Document, a number of changes to the project timeline have occurred. These have been caused by undertaking further research into other version control systems such as SVN as described in section 2.1 and the release of the new version of the .NET Framework. The following sections outline the changed milestones and project timeline.

### 5.1 Milestones

The milestones below mark the important progress points in the development of the version control system:

- **Initial Document Completion (20th October 2006)** - at this point a full feature list will have been created and the designing of the final software will be able to commence
- **SVN Research Completion (15th December 2006)** - at this point, I will have achieved a greater understanding of how another version control system - SVN - works and the features available
- **Web Service Experiment Completion (30th January 2007)** - at this point, I will have achieved a greater understanding of Web Services and how they will be used to create the final client and server applications
- **Prototype Build Completion (28th February 2007)** - at this point, a prototype of the server application will have been created
- **Essential Functionality Completion (30th March 2007)** - at this point, a beta version of the server and client will have been created, enabling testing to commence and the development of additional features
- **Release Candidate 1 (15th Apr 2006)** - at this point, the development and testing will have been completed
- **Project Completion (4th May 2006)** - at this point, the project will be complete

## 5.2 Timeline

Date	Details
29th August 2006	Background research of Web Services, version control systems and the .NET Framework begins
1st September 2006	SVN research begins
20th October 2006	Initial document complete
23rd October 2006	Web Service experiments begin
20th November 2006	Project presentation at the Undergraduate Computer Science Colloquium at Gregynog
15th December 2006	SVN research complete
30th January 2007	Web Service experiments complete
31st January 2006	Timeline review and development of key features begins
8th February 2007	Interim report complete
28th February 2007	Prototype of server application complete
30th March 2007	Essential features completed
30th March 2007	Testing and extra feature development begins
15th April 2007	Feature complete release candidate available
4th May 2007	Project completion

## Glossary

- .NET** Microsoft .NET Framework - Microsoft's new technology to enable greater communication between programs, 3-6, 8, 11
- PHP** PHP - A server side web page dynamic scripting language commonly used on Linux servers, 4
- SSH** Secure Shell - the communications protocol used by SVN to send messages between the client and server applications, 4
- SVN** Subversion - a new open source version management system built to replace CVS, 4, 6, 11

## A Appendix: SVN Screenshots

This appendix contains screenshots of the Subversion (SVN) version control system in operation on a Microsoft Windows XP client.

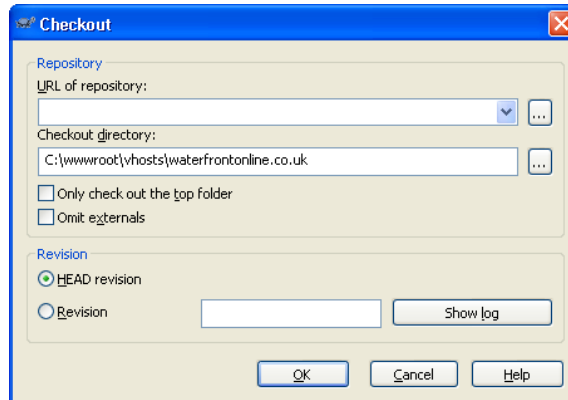


Figure 1: SVN “checkout” form

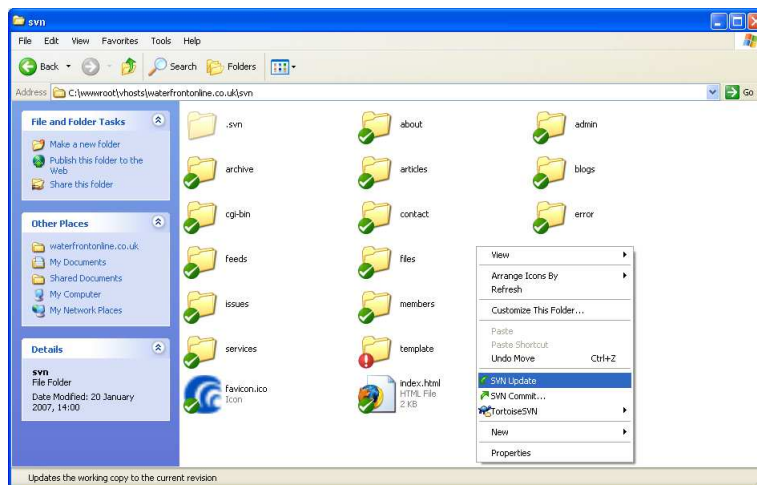


Figure 2: SVN integrated into the Windows shell

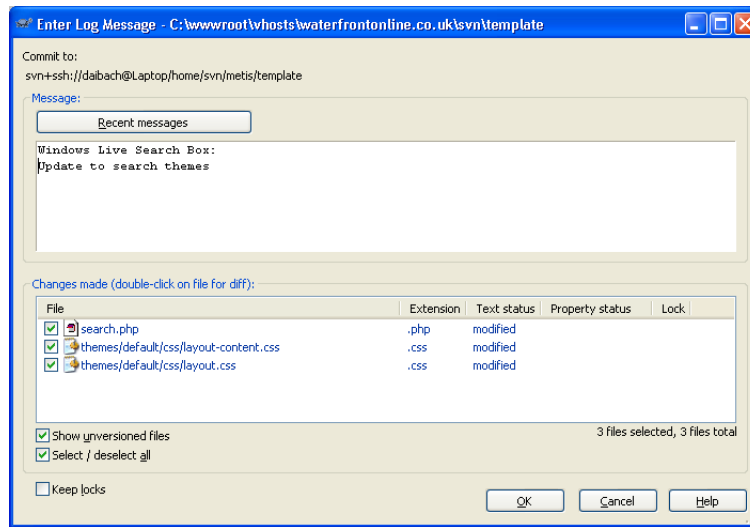


Figure 3: SVN commit form to enable user to specify a reason for update

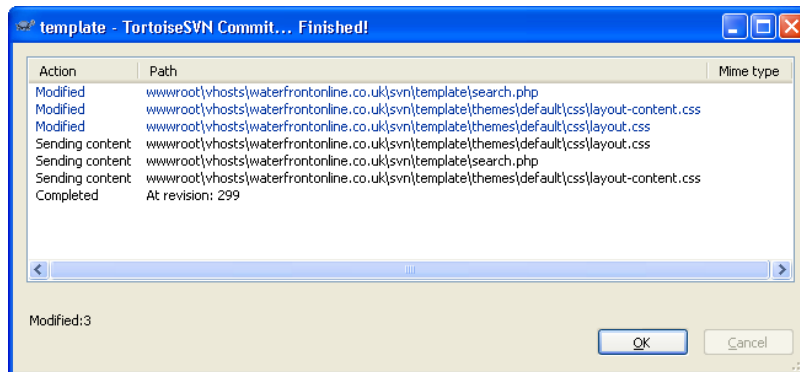


Figure 4: SVN commit progress box

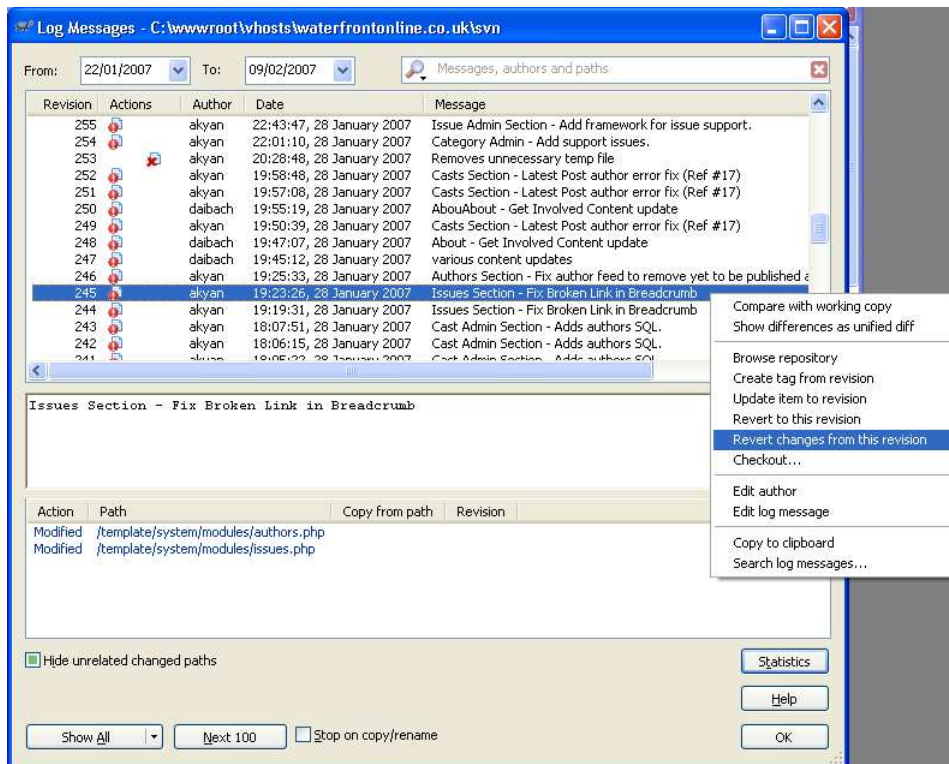


Figure 5: SVN version manager

## B Appendix: C# Application Screenshots

This appendix contains screenshots of the Number Evaluator Web Service C# application front end.

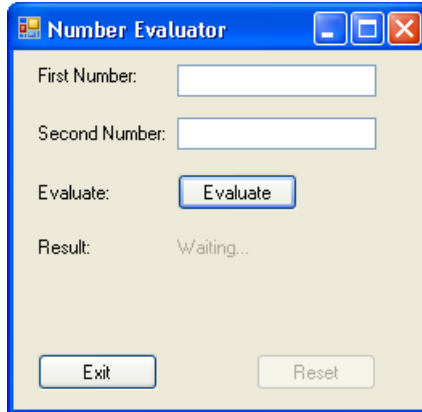


Figure 6: C# Application - Clear Form

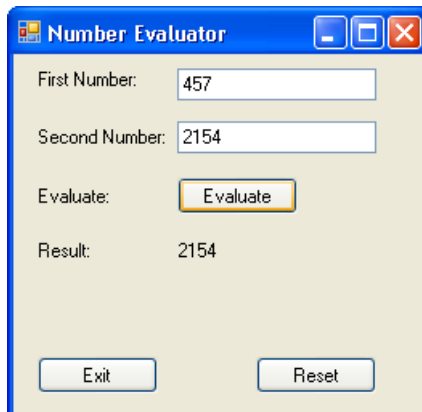


Figure 7: C# Application - First Number Smaller - Result

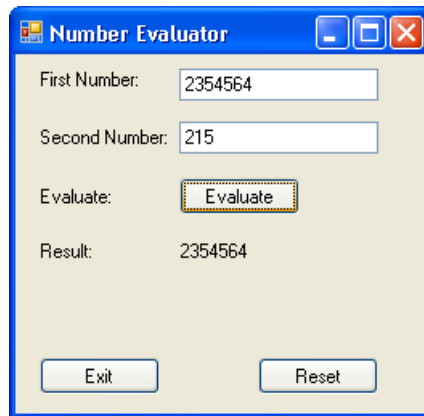


Figure 8: C# Application - First Number Bigger - Result

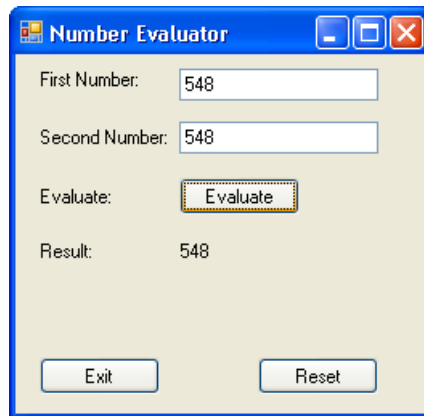


Figure 9: C# Application - Both Numbers Equal - Result