

Autor: Thomas Reinwart

2010-04-12

office@reinwart.com

TFS Team Foundation Server 2010

Der Zweck eines Source Safe Systems	2
Warum verwendet man ein Source Safe System?	2
Der Klassiker: Microsoft Visual SourceSafe (VSS)	2
VSS – TFS Unterschiede	3
TFS 2010.....	5
TFS Prozessvorlagen: Projekttyp MS-Agile , Scrum,	6
TFS Begriffe	6
Workspace	6
WorkItem	6
Check In/Out beim TFS	6
Lock	7
CheckIn Policies	7
Shelving.....	7
Reports.....	8
TFS Power Tools.....	8
Änderungen TFS 2008 /2010	9
Installation des TFS Team Explorer am Client	9
Team Foundation Server 2008 and Expression Studio 3	9
Visual Studio Team System 2008 Team Foundation Server MSSCCI Provider	10
TFS Web Service List	10
Projektmigration von VSS zu TFS	11
Migration VSS -> TFS: Aufbau des Mapping Files	13

Der Zweck eines Source Safe Systems

Ein SourceSafe System ist der Datentresor der Quellcodes, mit dem eine Versionsverwaltung der Projekte durchgeführt wird. Ohne vollständigen Quellcode lässt sich eine Anwendung nicht kompilieren. Die zentrale Quellcode Verwaltung ist die Basis für eine professionelle Softwareentwicklung, für ein Team oder für einen einzelnen Entwickler.

Es gibt viele gute SourceSafe Systeme, ich werde hier 2 Systeme von Microsoft beschreiben, Visual SourceSafe (VSS) – der Klassiker und den Nachfolger Microsoft Team Foundation Server (TFS).

Warum verwendet man ein Source Safe System?

- Zentrale Ablage des (gemeinsamen) Sourcecodes, von Dokumentation, ...
- Versionierung, Historie, Label
- Paralleles Entwickeln im Team, Branch, Merge
- Sicherung zentral (Absicherung gegen Verlust des Entwicklungsrechners)
- Immer aktueller Stand: regelmäßiges ablegen in VSS während der Arbeit
- Arbeiten im Entwicklungsteam, unterschiedliche Standorte möglich
- Ausgangspunkt für einen Build Prozess

Der Klassiker: Microsoft Visual SourceSafe (VSS)

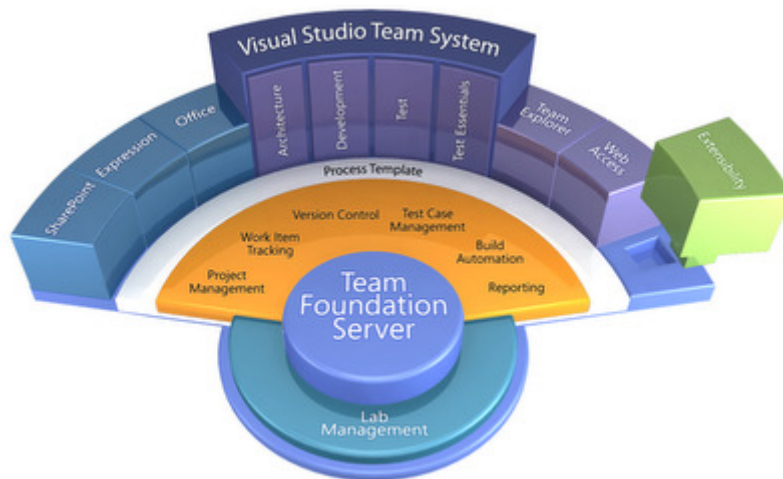
VSS gibt es seit 1992. Visual Source Safe ist eine Versions- und Dateiverwaltung für Dateien beliebigen Typs (Text, Binär, Grafik, etc.). Visual Source Safe (VSS) eignet sich insbesondere zur Verwaltung und Versionierung von Quellcode. VSS ist im Wesentlichen eine „Datenbank“ (File basierend) mit besonderen Fähigkeiten zur Versionsverwaltung von Dateien. Die Änderungen von einer Version einer Quelldatei zur nächsten werden mittels eines Delta-Algorithmus gespeichert, sodass einerseits ein „Zurücksteigen“ auf eine ältere Version bzw. das Nachvollziehen von Änderungen von einer Version zur nächsten ermöglicht sind. Weiters achtet VSS darauf, dass Änderungen eines Benutzers nicht durch gleichzeitige Änderungen in demselben File eines anderen Benutzers überschrieben werden.

Die letzte verfügbare Version von VSS war die Version 2005.

VSS – TFS Unterschiede

VSS kann nur Files (SourceCode) auf dem Filesystem verwalten, TFS verwaltet den Code performanter und sicherer in der SQL Server DB und bietet darüber hinaus wesentlich mehr Möglichkeiten als ein reines SourceSafe System.

Bisherige TFS Versionen: 2005, 2008, 2010



Unterschiede Technik:

- Umfang: TFS ist ein voll integriertes Application Life Cycle Management Tool, VSS ein einfaches Source Control System
- Architektur: TFS ist ein Client/Server Produkt, VSS ein standalone Client mit Ad-Hoc Server
- Team-Größe: TFS eignet sich für Teams bis 3500 Mitarbeiter, VSS für Einzelne oder kleinere Teams
- Speicherung: TFS basiert auf SQL Server, VSS auf dem File System mit 4GB Limit
- Zuverlässigkeit: TFS verwendet transaktionale Check-In's, VSS unterstützt dies nicht
- Remote-Access: TFS basiert auf optimierten Web Services und eignet sich hervorragend für Remote Access, VSS besitzt seit VSS 2005 einfache Web Services für "kleinere" Remote Zugriffe
- Security: TFS basiert auf Active Directory und/oder Windows Security, VSS hat seine eigene Applikations spezifische Security

Unterschiede Funktionalitäten:

- Pinning: Wurde in TFS durch professionelle Branching/Merging Funktionen ersetzt
- Sharing: Sharing by Reference und Merging ermöglichen in TFS robustere Changes
- Destroy: Gibt es ab TFS 2008
- Shadow Folders: Kein Equivalent in TFS
- Working Folders: TFS besitzt Workspaces, von denen jeder Benutzer mehrere besitzen kann.

Was bietet TFS:

- Work Item Integration erlaubt dem Benutzer ein umfassendes Dokumentieren seiner Arbeit
- Project Alerts (Mail) informieren den Benutzer unmittelbar über erfolgte Änderungen
- Check-In Policies reduzieren das Risiko beim Erstellen der Builds und erhöhen die Qualität der Check-In's

- Branching/Merging auf Changeset Ebene ermöglichen ein feines Management von Code Änderungen
- Integration von Work Items und Test Resultaten erlauben eine volle Traceability
- Verbesserte Zusammenarbeit und Kommunikation (Stand der Dinge,)
- Laufend Status und Prognosen (Reports) ob sauber compilierbar, unit getestet, Statusübergänge, Bugs, Aufgaben,) über den Projektfortschritt an sich
- Arbeiten mit vertrauten Werkzeugen: Projektmanager mit Excel und MS Projekt, Entwickler mit Visual Studio, weiters Sharepoint, SQL Server Reports, Web Access
- Laufende Qualitätsprüfung von Anfang an (Was kostet es einen Fehler auszubessern während Entwicklung, Testphase, Produktion –geht bis Faktor 1000)
- Regelmäßige Integration (Build von Projekten), Builds am Server nach verschiedenen definierbaren Regeln (beim Checkin, nach Zeit, ...)

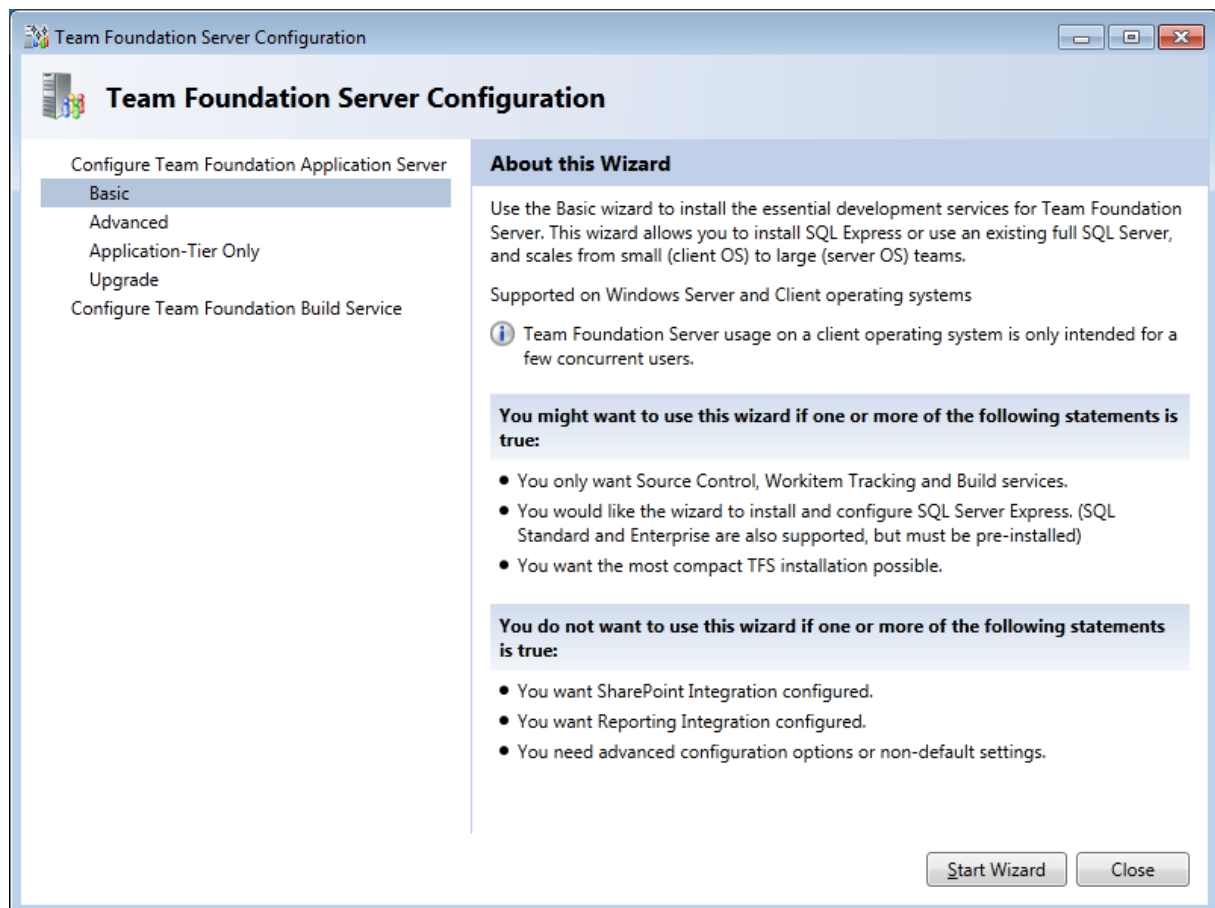
TFS 2010

Neuerungen zur TFS Version 2008 sind:

- TFS kann auch am Domain Controller installiert werden (kleine Firmen haben ev. nur eine geringe Anzahl von Servern)
- Die Restriktion der Installation auf Windows Server gibt es nicht mehr, eine Installation auf Clients - wie Windows Vista und Windows 7 - ist auch möglich. (etwa Notebook)
- Die „Basic“ Installations Variante kommt auch ohne Sharepoint Server Installation aus
- Bei „Basic“ wird auch SQL Server Express als Datenbank zugelassen
- 32 und 64 Bit OS Support
- Preis? (noch nicht bekannt, derzeit Beta RC getestet)

Die Installation besteht aus drei Wizzard unterstützten Paketen: Basic, Standard und Advanced.

Die Basic Variante ist jene, die ohne Sharepoint auskommt und sich auch mit SQL Express begnügt, diese sogar selber installiert, wenn diese noch nicht am Rechner ist, ebenso erfolgt Konfiguration im IIS automatisch. In 20 Minuten ist TFS Basic lauffähig.



TFS Prozessvorlagen: Projekttyp MS-Agile , Scrum, ...

Bei der Anlage eines Projektes muss eine Prozessvorlage gewählt werden. Je nach gewählten Prozess ist das erzeugte Projekt anders aufgebaut (Bsp. GUI Felder bei Workitems) und auch das Verhalten entspricht dem gewählten Prozess.

- MSF for agile software development (Microsoft Standardvorlage)
- MSF for CMMI (Capability Maturity Model Integrated) process improvement (Microsoft Standardvorlage)
- Scrum for Team System (Conchango)
- Microsoft process template (auf Codeplex)

Weitere wie Rational Unified Process, V-Modell XT, Essential Unified Process usw. werden angeboten.

TFS Begriffe

Workspace

Ist eine Zuordnung zwischen dem clientseitigen Projektordner (c:\projekte\MeinProjekt) und dem Ordner im TFS. Ein Workspace gehört jeweils einem User und einem Computer. Man kann mehrere Workspaces auf einem Client anlegen. Im „pending changes“ (Änderungen zum CheckOut) Fenster werden die Elemente des aktuellen verwendeten Workspace angezeigt.

WorkItem

Ein WorkItem entspricht einer Aufgabe im Projekt, die der Kategorie Bug, Quality, Risk, Scenario oder dem Typ Task entspricht.

Einem CheckIn Vorgang wird ein WorkItem zugeordnet, somit kann man im Nachhinein feststellen, in welchen Files sich ein Bug ausgewirkt hat.

Check In/Out beim TFS

Begriffe:

CheckIN: Upload auf TFS

CheckOut: „ich möchte es bearbeiten“

Get: Download letzte bzw. bestimmte Version vom TFS

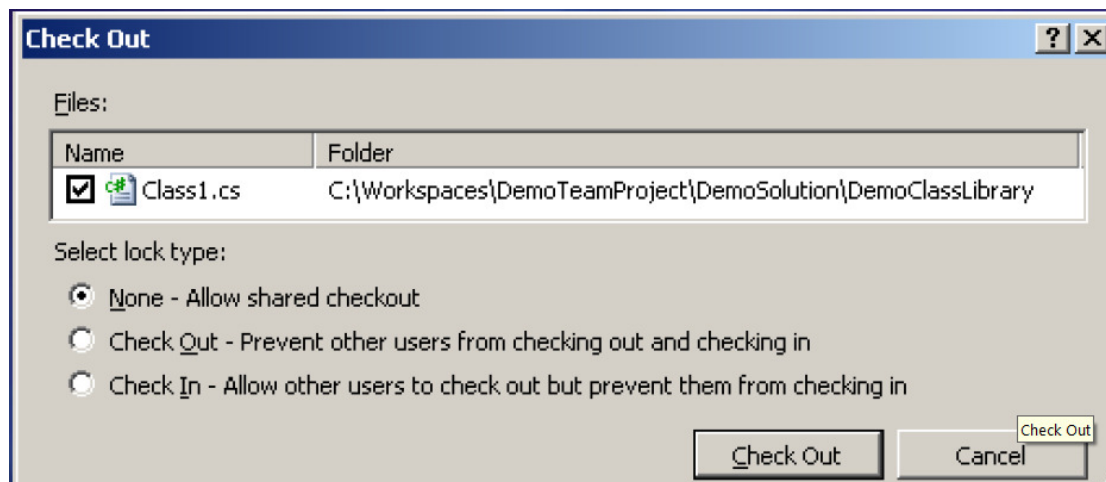
Beim Einchecken werden die Files als „Changeset“ als DB Transaktion in den TFS übertragen. Ein Changeset enthält die Metadaten, sowie (je nach Config optional oder verpflichtend) einen Kommentar, beliebige Anzahl an Eincheck Notizen und eine Verknüpfung zur Aufgabe.

An der gewohnten Bedienung am GUI ändert sich in der Visual Studio Umgebung bei Verwendung des TFS nicht viel. Der einzige Unterschied ist, dass ein File in VSS exklusiv für einen User ausgecheckt und damit für die anderen gesperrt wurde, in TFS kann ein File mehrfach ausgecheckt werden. Der TFS prüft bei CheckIn automatisch, ob am Server bereits eine unterschiedliche Version zu der einzucheckenden besteht. Ist dies der Fall, erscheint beim letzten User des gemeinsamen Projekts,

der den CheckIn Vorgang durchführt, ein weiterer Dialog, der darauf aufmerksam macht und die Differenz in zwei Fenstern nebeneinander farblich darstellt. Man kann die Differenz nun manuell oder automatisch durchführen lassen.

Lock

TFS verwendet „Multiple Checkout“, d.h. jeder kann die gleichen Files bearbeiten. Es ist für jedes Projekt ein/aus schaltbar. Es ist pro Datentyp für den gesamten TFS gültig. (z.B. macht es für jpg Files keinen Sinn etc.) Außerdem ist ein Lock auch bei einem CheckOut explizit wählbar.



CheckIn Policies

Eine CheckIn Policy kann pro Projekt definiert werden. Diese Richtlinie prüft den Code lokal ab, je nach Ergebnis wird entschieden, ob eingchecked werden darf. Die Policy kann mit eigenen Regeln erweitert werden. Der Sinn dahinter ist, einen sauberen Code im Sinne einer Richtlinie zu erhalten und Folgefehler, Wartungsaufwand etc. dadurch zu minimieren. Der User kann diese Richtlinie auch übergehen (Override), muss dafür aber einen Grund angeben, ev. hat es sich um einen super wichtigen Hotfix gehandelt der keine Zeit für Richtlinien zulässt. Die Begründung ist im TFS lesbar.

Fertig einsetzbare Richtlinien:

- Work item policy
- Code analysis
- Test policy
- CI Build not failed policy
- Custom Patch policy
- Forbidden patterns policy
- Changset comment policy
- Work item query policy

Shelving

Der Sinn eines SourceSafe Systems ist, den Code mindestens einmal täglich (im Normalfall wesentlich öfter) einzuchecken. Gründe sind: Datensicherung, Rückstiegsmöglichkeiten auf vorherige Version bei der Entwicklung, Erkrankung, ...

Der eingchecked Code soll von hoher Qualität sein und immer kompilierbar sein. (CheckIn Policies)

Wie kann ich meinen derzeit in Arbeit befindlichen Code trotzdem am TFS ablegen? - Shelving
Fertiger Code (Funktionsfähig, CheckIn Policy OK, Build bar) wird eingchecked, anderer Code wird
geshelved („ins Regal gelegt“).

Mittels shelving kann also unfertige Arbeit gesichert werden (Unterbrechung, Wochenende, Urlaub,
...). Auch die Weitergabe von Code an einen Kollegen (der ist z.B. Experte für diese Funktion) ist
damit direkt möglich, Code Versand per Mail ist dadurch überflüssig.

Die Zwischenspeicherung (Checkpoints) vor dem nach Hause gehen oder das wechseln zwischen
parallelen Aufgaben lassen sich damit lösen. Shelves setzen multiple CheckOut voraus.

Reports

Die gesammelten Daten der Changes, Workitems usw. im Projekt speichert der TFS in einer eigenen
Datenbank und dient als Basis für eine umfangreiche Auswertungsmöglichkeit.

TFS Power Tools

Bieten zusätzliche Funktion, wie der Name schon verrät.

- Command line tool (TFPT.EXE)
- Team Explorer IDE menu additions
- Build Notification tool
- TFS Best Practices Analyzer
- Process Template Editor
- Work Item Templates
- Custom check-in policies
- TFS Server Manager
- TFS Users tool
- Alert Editor
- TFS Power Shell Extensions
- TFS Windows Shell Extensions
- Team Members Tool

<http://www.microsoft.com/downloads/details.aspx?FamilyId=FBD14EEA-781F-45A1-8C46-9F6BA2F68BF0&displaylang=en#filelist>

Änderungen TFS 2008 /2010

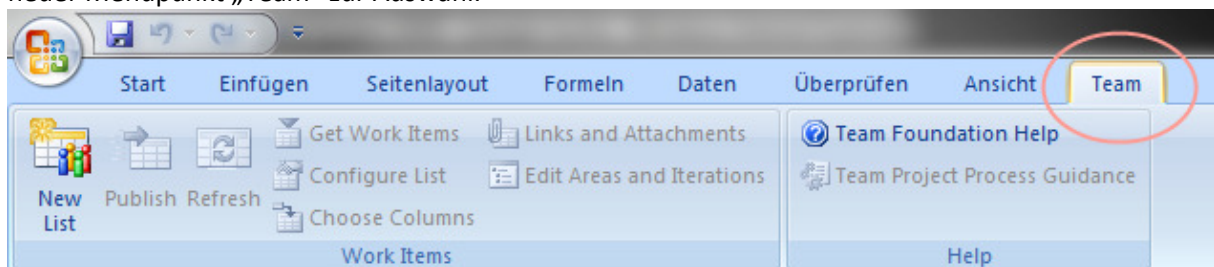
Der **Build Prozess (MS Build)** basiert nun auf der **Workflow Foundation (WF) 4.0** und kann dadurch wesentlich komplexere Prozesse abbilden. Der Build Rechner, der ein anderer als der TFS sein kann, unterstützt Build queuing. Es ist auch möglich, mehrere Build Rechner zu verwenden, die über **build agent pooling** gemanaged werden.

Die **Branche / Merge Funktion** wird auch **graphisch dargestellt**, kann auch per drag and drop bedient werden.

Je größer das Entwicklungsteam wird, desto häufiger kommt es zu Build Fehlern, die gemeldet werden, die Entwickler beheben die Fehler und der Prozess beginnt von neuem. Mittels **Gated checkin** und **Buddy builds** wird der Build und die Unit Tests vor dem Einchecken am Server durchgeführt, was die Build Fehlerrate am Server reduziert.

Installation des TFS Team Explorer am Client

Der TFS Client kann unabhängig von der installierten Visual Studio Edition installiert werden. Nach der Installation des TFS Clients stehen in den Anwendungen Visual Studio, MS Excel, MS Projekt ein neuer Menüpunkt „Team“ zur Auswahl.



Bsp. Excel 2007 Team Integration im Menu

<http://www.microsoft.com/downloads/details.aspx?FamilyID=0ed12659-3d41-4420-bbb0-a46e51bfca86&displaylang=en>

Team Foundation Server 2008 and Expression Studio 3

Um den TFS in Expression Projekten nutzen zu können ist das Update *VS90SP1-KB967483-x86.exe* nach dem SP1 von Visual Studio notwendig.

Visual Studio Team System 2008 Team Foundation Server MSSCCI Provider

Dient für ältere oder Fremdprodukte, die keinen TFS Explorer unterstützen. Vor der Installation muss der „Visual Studio Team System 2008 Team Explorer“ installiert werden, MSSCCI Setup anschließend.

<http://www.microsoft.com/downloads/details.aspx?familyid=FAEB7636-644E-451A-90D4-7947217DA0E7&displaylang=en>

- Visual Studio .NET 2003
- Visual C++ 6 SP6
- Visual Visual Basic 6 SP6
- Visual FoxPro 9 SP1
- Microsoft Access 2003 SP2
- SQL Server Management Studio
- Sparx Systems Enterprise Architect 6.1
- Sybase PowerBuilder 10.5
- Toad for SQL Server 2.0

Die TFS Schnittstelle für die Quellcodeverwaltung ist ein Webservice, daher stehen für einige weitere Entwicklungsumgebungen Plugins zur Verfügung.

Bsp. System

Eclipse
OpenTF
Microsoft Visual Studio Team Explorer 2010
codename "Eaglestone"

Url

<http://vstseclipse.sourceforge.net/>
<http://code.google.com/p/opentf/>
Eclipse plug-in and cross-platform command-line client for
Visual Studio 2010 Team Foundation Server

TFS Web Service List

Build Service

V1.0

<http://MyTFS:8080/Build/V1.0/BuildController.asmx>

<http://MyTFS:8080/Build/V1.0/BuildStore.asmx>

<http://MyTFS:8080/Build/V1.0/Integration.asmx>

<http://MyTFS:8080/Build/V1.0/PublishTestResultsBuildService.asmx>

<http://MyTFS:8080/Build/V1.0/PublishTestResultsBuildService2.asmx>

V2.0

<http://MyTFS:8080/Build/V2.0/BuildService.asmx>

<http://MyTFS:8080/Build/V2.0/Integration.asmx>

Services Service

V1.0

<http://MyTFS:8080/Services/v1.0/AuthorizationService.asmx>

<http://MyTFS:8080/Services/v1.0/CommonStructureService.asmx>

<http://MyTFS:8080/Services/v1.0/EventService.asmx>

<http://MyTFS:8080/Services/v1.0/GroupSecurityService.asmx>

<http://MyTFS:8080/Services/v1.0/ProcessTemplate.asmx>

<http://MyTFS:8080/Services/v1.0/projectMaintenance.asmx>

<http://MyTFS:8080/Services/v1.0/Registration.asmx>

<http://MyTFS:8080/Services/v1.0/ServerStatus.aspx>

V2.0

<http://MyTFS:8080/Services/v2.0/GroupSecurityService2.aspx>

Version Control Service

V1.0

<http://MyTFS:8080/VersionControl/v1.0/Administration.aspx>

<http://MyTFS:8080/VersionControl/v1.0/Integration.aspx>

<http://MyTFS:8080/VersionControl/v1.0/ProxyStatistics.aspx>

<http://MyTFS:8080/VersionControl/v1.0/Repository.aspx>

Warehouse Service

V1.0

<http://MyTFS:8080/Warehouse/v1.0/warehousecontroller.aspx>

WorkItem Tracking Service

V1.0

<http://MyTFS:8080/WorkItemTracking/v1.0/ClientService.aspx>

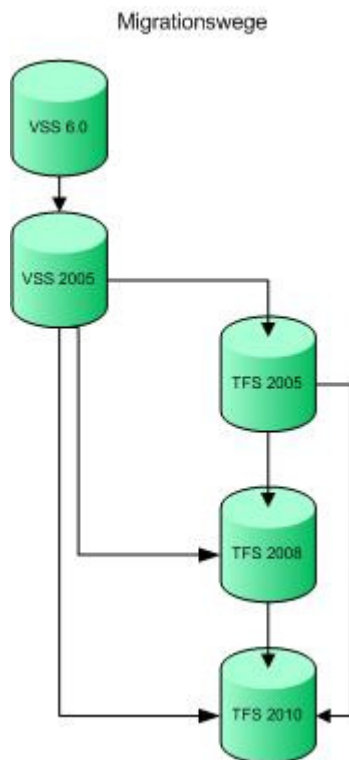
<http://MyTFS:8080/WorkItemTracking/v1.0/ConfigurationSettingsService.aspx>

<http://MyTFS:8080/WorkItemTracking/v1.0/ExternalServices.aspx>

<http://MyTFS:8080/WorkItemTracking/v1.0/Integration.aspx>

<http://MyTFS:8080/WorkItemTracking/v1.0/SyncEventsListener.aspx>

Projektmigration von VSS zu TFS



Migration VSS -> TFS

Für die Migration von VSS nach TFS ist VSS Server Version 2005 notwendig.

Grund: VS 2005 hat eine erweiterte API, mit der die Analyse und die Migration von VSS nach TFS erst möglich wird.

Die in VSS abgelegten Projekte können historisch komplett in TFS übernommen werden, also alle Files aller Versionen, der Beschreibungen und der Labels.

Die einzige Einschränkung für eine Übernahme ist die Verwendung von *Shares*, wie sie in VSS erlaubt waren. Mittels dieser Shares konnten gemeinsame Files projektübergreifend abgelegt werden, mit allen Nachteilen, die sich daraus ergeben können. Diese Shares sind zuvor in VSS aufzuheben.

VSS hat eine eigenständige Benutzerverwaltung und Rechteverwaltung das in einer Datei am Filesystem abgelegt wurde, TFS nutzt die User des Active Directories.

Die Migration erfolgt mittels Tool (vssconverter.exe), diese Schritte sind durchzuführen:

- Prüfung des Projekts in VSS (Alle Versionen konsistent, keine Shares vorhanden)
- Erstellen eines Mapping Files VSS / TFS (Welcher VSS User/Rechte zu Active Directory User/TFS Rechten, VSS Projekt zu TFS Projekt)
- Starten der Migration

In diesem Bsp: liegt die VSS DB im Verzeichnis: c:\VSSstest

Zuerst die Fehleranalyse durchführen: `analyze -v4 c:\VSSstest\data\`

`c:\Programme\Microsoft Visual SourceSafe\analyze.exe` (cmd auf Installpfad VSS erstellen)

Wenn kein Fehler ist schaut das Ergebnis so aus:

Building the project list.

Checking cross file relationships.

Checking parent/child relationships.

Started analyzing user management system.

Successfully completed analyzing the user management system.

Validating the Rights system.

Running advanced reference verification.

Validating references.

Wenn Fehler gemeldet werden, eine Reparatur durchführen:

`analyze -f -v4 c:\VSSstest\data\`

```
VSSConverter analyze settings.xml .... Starts Analysis  
VSSConverter migrate settings.xml .... Starts Migration
```

Migration VSS -> TFS: Aufbau des Mapping Files

Fett markierte Werte sind zu ersetzen:

```
<?xml version="1.0" encoding="utf-8"?>
<SourceControlConverter>
  <ConverterSpecificSetting>
    <Source name="VSS">
      <VSSDatabase name="\VSSDB\vss"></VSSDatabase>
      <UserMap name="Usermap.xml"></UserMap>
    </Source>
  <ProjectMap>
    <Project Source="$FolderA" Destination="$TeamProjectA"></Project>
    <Project Source="$FolderB" Destination="$TeamProjectB/ProjectB"></Project>
  </ProjectMap>
</ConverterSpecificSetting>
<Settings>
  <TeamFoundationServer name="server_name" port="port_number" protocol="protocol"></TeamFoundationServer>
</Settings>
</SourceControlConverter>
```

Nach der durchgeführten Analyse wird die Datei usermap.xml erstellt. Sie beinhaltet die User, die in VSS zu dem angegebenen Projekt involviert waren. Die rechte Seite des Mappings bleibt gleich, diese ist nun mit den Usern des Active Directories abzugleichen und zu befüllen.

```
<?xml version="1.0" encoding="utf-8" ?>
<UserMappings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <UserMap From="VSSuser" To="Domain\User" />
</UserMappings>
```