

TortoiseSVN

Kullanım Klavuzu

Telif hakkı (c)2008 İsmail SEZEN.

Bu belgenin, GNU Özgür Belgeleme Lisansı, Sürüm 1.2 veya Özgür Yazılım Vakfı tarafından yayımlanmış daha yeni sürümlerindeki koşullara uygun şekilde; değişmeyen bölümler, ön kapak ve arka kapak metni olmaksızın, kopyalanması, dağıtılması ve/veya değiştirilmesine izin verilmiştir. Lisansın bir kopyası "[GNU Free Documentation License](#)" adlı bölüme eklenmiştir.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being just "GNU Manifesto", with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "[GNU Free Documentation License](#)".

İçindekiler Dizini

1.Subversion – (SVN).....	4
1.1.Subversion 'a Giriş.....	4
1.2.Subversion Nasıl Çalışır?.....	4
2.TortoiseSVN.....	4
2.1.Giriş.....	4
2.2.TortoiseSVN Kurulumu.....	5
2.3.TortoiseSVN ve SubVersion Kullanımı.....	5
a)Depo Oluşturma (Creating Repository).....	5
b)Depoya Klasör Ekleme (Import).....	6
c)Çalışma Kopyası Oluşturmak (Checkout).....	8
d)Çalışma Kopyasına dosya veya klasör eklemek (Add).....	11
e)Çalışma Kopyasındaki bir dosyaya veya klasörü silmek (Delete).....	12
f)Dosya veya klasörleri yok saymak (Ignore).....	13
g)Değişiklikleri geri almak (Revert).....	14
h)Değişikliklerin depoya teslim edilmesi (Committing).....	15
i)Çalışma Kopyasını Güncellemek (SVN Update).....	17
j)Depo Günlüğünü Takip Etmek (Show Log).....	17
k)Çalışma kopyasını temizlemek (Cleanup).....	19
l)Çakışmaları düzeltmek (Resolving Conflicts).....	19
3.İyi Alışkanlıklar.....	20
3.1.Depo klasör düzeni.....	20
3.2.Projeniz üzerinde çalışmaya başlamadan önce güncelleyin.....	21
3.3.Çalıştığınız bölümleri diğer kullanıcılara bildirin.....	21
3.4.Teslim işlemi yaparken daima açıklayıcı ve kısa notlar yazın.....	21
3.5.Çalışmanızı bitirmeden depoya teslim işlemi yapmayın.....	22
3.6.Sürümlenmemiş dosyaları unutmayın.....	22
4.Kaynaklar.....	23
5.İletişim.....	23
6.GNU Özgür Belgeleme Lisansı.....	24

1. Subversion – (SVN)

1.1. Subversion 'a Giriş

Subversion; “Tigris.org: Açık Kaynak Yazılım Mühendisliği Araçları” sitesi ve CollabNet şirketi adı altında [Http://subversion.tigris.org/](http://subversion.tigris.org/) internet adresinde yayınlanan açık kaynak kodlu bir **Sürüm Kontrol Sistemi**'dir. Subversion 'ı, yayıncılarının da belirttiği gibi kimseye haber vermeden istenilen şekilde geliştirilebilir, dağıtılabilir ve kullanabilirsiniz. Subversion 'ın temel yapılış amacı başka bir sürüm kontrol sistemi olan CVS 'nin eksik yanlarını tamamlayarak, CVS 'nin kullanıcılarını kendi taraflarına çekmek.

Subversion 2000 yılından bu yana geliştirilmektedir ve halen bu gelişim süreci devam etmektedir. Fakat temel yapısı 2001 yılında oluşturulduğu için şu zamanda yapmış olduğunuz çalışmalar için yeterli kararlığa sahiptir.

Subversion birçok işletim sistemi üzerinde sorunsuz çalışmaktadır. Subversion ANSI C ile yazıldığından dolayı APR (Apache Portable Runtime) 'nin çalıştığı bütün sistemlerde çalışır.

Subversion 'ın iki türlü kullanımı vardır. Birincisi depo (Repository) olarak kullanıldığı sunucu kullanımı bir diğeri ise depo'daki dosyalara erişim için kullanılan istemci kısmıdır. Subversion 'ın proje durumunu [Http://subversion.tigris.org/project_status.html](http://subversion.tigris.org/project_status.html) adresinden takip edebilirsiniz.

Subversion verilerinizi saklarken [berkeleydb](#) ve [fsfs](#) veri saklama sistemlerini kullanır. Subversion 'ı her türlü proje ve dosya için kullanabilirsiniz.

1.2. Subversion Nasıl Çalışır?

Subversion, yaptığınız projeler için bir depolama aracıdır ve komut satırından çalışan bir uygulamadır. Subversion ile ister kendi bilgisayarınızda isterseniz de uzak bir bilgisayarda depo oluşturup, yapmış olduğunuz çalışmaları bu depoda saklayabilirsiniz. SubVersion 'ın görevi, çalışmalarınızı, yaptığınız her değişikliği bir sürüm numarası vererek depoda saklamaktır. Böylece istediğiniz zaman önceki bir sürüme geri dönebilir, nerede, ne zaman, ne değiştirdiğinizi görebilir, çalışmalarınıza bu şekilde yön vererek ilerleyebilirsiniz. Subversion 'ın sunucu özelliğini kullanarak uzak bir kaynaktan ya da kendi bilgisayarınızda depo oluşturabilir ve istemci özelliği ile de Subversion 'ın oluşturmuş olduğu depodaki dosyalara ulaşabilirsiniz.

2. TortoiseSVN

2.1. Giriş

TortoiseSVN projesi [Http://tortoisesvn.tigris.org/](http://tortoisesvn.tigris.org/) sitesinde yürütülmektedir. Yukarıda SubVersion 'ın bir komut satırı uygulaması olduğundan bahsetmiştik. Yine aynı ekip tarafından geliştirilen TortoiseSVN uygulaması, SubVersion 'ın bu eksikliğini kapatır ve aynı ekip tarafından geliştirildiğinden SubVersion ile hiçbir uyum problemi yoktur. Birçoğumuzun bilgisayar ile geçmişi uzun bir süreye dayanıyordur. Subversion 'da normalde konsol ekranı üzerinde belli komutlarla çalışan bir sistemdir. Ama biz sevgili Windows kullanıcıları için geliştirilen TortoiseSVN sayesinde bu zorlukları yaşamıyoruz. TortoiseSVN Windows kabuğu ile bütünleşerek, birkaç tıklamayla bizim yerimize gerekli komutları Subversion 'a gönderip gerekli işlemleri yapıyor.

2.2. TortoiseSVN Kurulumu

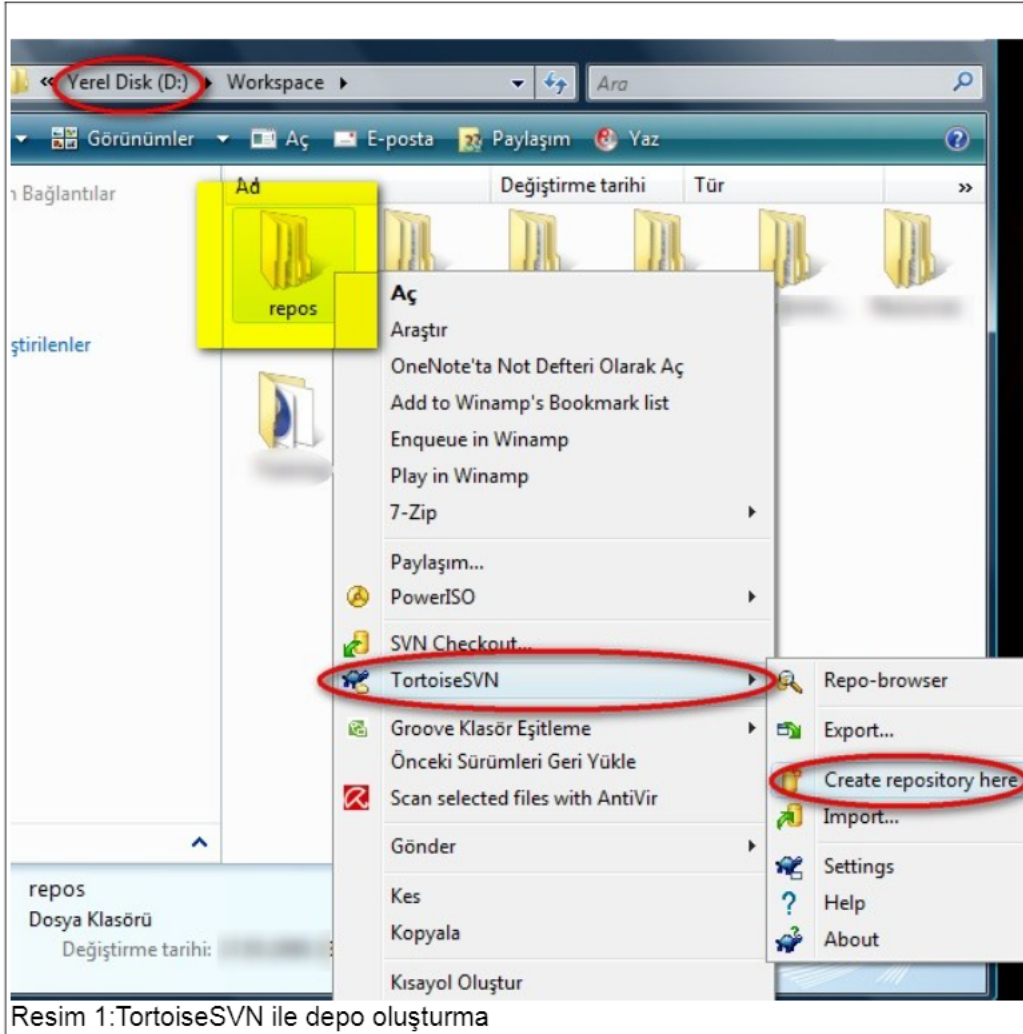
TortoiseSVN programını [Http://tortoisesvn.net/downloads](http://tortoisesvn.net/downloads) adresinden indirebilirsiniz. Yine aynı bağlantı altında programa ait Türkçe dil paketi bulunmaktadır. Programı indirdikten sonra her zamanki kurulum sırasını takip ederek bilgisayara yükledikten sonra bizden bilgisayarı kapatıp açmamızı isteyecek. Program varsayılan olarak C:\Program Files\TortoiseSVN klasörüne kuruluyor.

2.3. TortoiseSVN ve SubVersion Kullanımı

TortoiseSVN programının kullanımını aşağıda gruplandırılmış ana başlıklar altında inceleyelim.

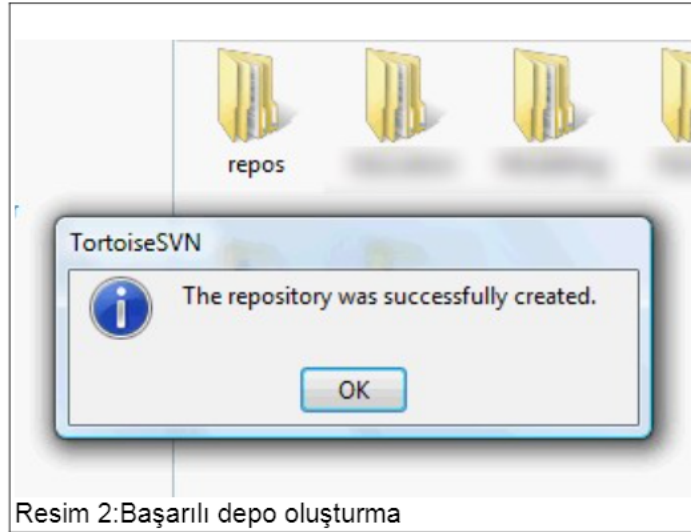
a) Depo Oluşturma (Creating Repository)

Depo oluşturmak için öncelikle depoya ev sahipliği yapacak olan bir klasör oluşturmalıyız. Bu klasörün depo olarak kullanılabilmesi için boş ve yazma korumasının olmaması şarttır. Bununla birlikte, depo klasörümüzü işletim sisteminin kurulu olmadığı ikinci bir yerel diskte oluşturmak iyi bir seçim olacaktır. Zira, depo oluşturmanın amacı, zaten takip ettiğimiz proje veya belgeleri güvenli bir yerde saklamayı da içermektedir. Oluşturduğumuz klasöre “repos” adını verelim. Daha sonra oluşturduğumuz bu klasöre sağ tıklayarak TortoiseSVN menüsünden “Create Repository Here” seçeneğini tıklayın.



Resim 1: TortoiseSVN ile depo oluşturma

Bu komutu verdikten sonra “The Repository was successfully created.” iletişim kutusu deponun başarı bir şekilde oluşturulduğunu bize bildirir. TortoiseSVN varsayılan olarak `fsfs` tipinde bir depo oluşturur.



Resim 2:Başarılı depo oluşturma

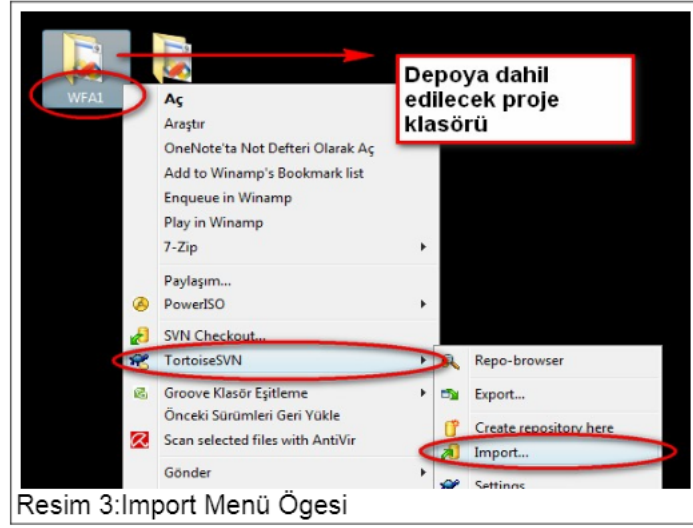
Eğer “repos” klasörünü açacak olursanız içerisinde bazı dosya ve klasörlerin bulunduğunu göreceksiniz. ReadMe.txt dosyasında da yazdığı üzere bu klasördeki dosyalara kesinlikle el ile müdahale etmeyin, silmeyin ve değiştirmeyin. Bu klasöre sadece SubVersion ya da TortoiseSVN üzerinden erişmeniz deponun sağlığını koruyacaktır.

Ek olarak, isterseniz her proje için yukardaki yöntemi izleyerek ayrı ayrı depolar oluşturabilir ya da tek bir depoyu bütün projelerinizi takip etmek için kullanabilirsiniz. Çalışma verimliliğiniz açısından ikisinin bir üstünlüğü yoktur. Eğer bir tek depoyu bütün projeleriniz için kullanırsanız, bu depodaki projelerin tamamı aynı *sürüm numaralandırması* paylaşır. Açıkçası, sürüm numaralandırmasının projelerin gelişimi açısından pek bir ehemmiyeti yoktur. Sadece depoya yapılan teslim (Committing) işlemlerinin birbirinden ayırma vazifesi götür. her bir projeniz için farklı depolar kullanmanız durumunda ise her proje için bir depo oluşturmanız gerekir. Bu da bir tek depo takip etmek yerine, birden fazla depo takip etmeniz anlamına gelir. Fakat, her bir projenize ait sürüm geçmişini (Log/History) verimli şekilde takip edebilmek için, her bir proje için bir depo oluşturmak faydalıdır. Açıkçası, ben her bir proje için ayrı bir depo oluşturmayı tercih ediyorum, ve böylece projeleri mantıksal olarak ayırmanın yanında fiziksel olarak da ayırmış oluyorum. Sürüm numarası konusunda [Değişikliklerin depoya teslim edilmesi \(Committing\)](#) bölümünde daha geniş bahsedeceğiz.

b) Depoya Klasör Ekleme (Import)

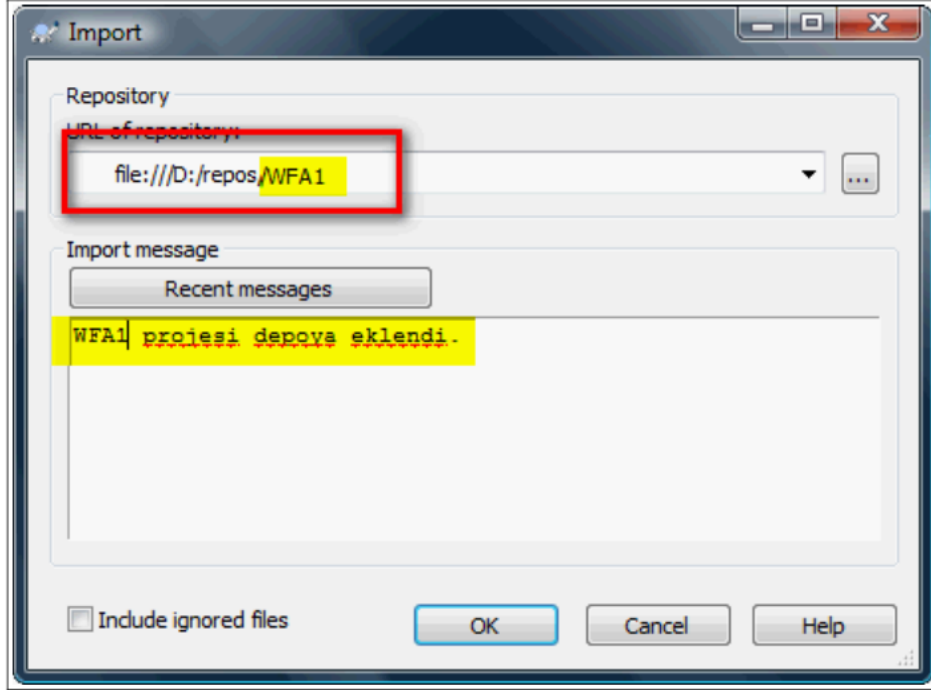
Daha önce oluşturduğunuz deponuza herhangi bir çalışmanızı eklemek için “Import” komutu kullanılır. Import işleminden önce depoya eklemek istediğiniz projenizin klasör yapısını belli standartlara uydurmak ve netleştirmek önemlidir. Ancak biz şimdilik bunu göz ardı edeceğiz. Daha sonra iyi alışkanlıklar bölümünde depoya gönderilmesi gereken klasör yapısının nasıl olması gerektiğinden bahsedeceğiz.

Örneğin, halihazırda kayıtlı 2 adet .net projemizin olduğunu varsayalım. Örnek projeler varsayılan olarak Windows Forms Application projesi olsun ve bu projeleri oluşturduğumuz depolara nasıl dahil edeceğimizi görelim. Öncelikle WFA1 ismini verdiğim 1 numaralı Windows Forms Application projesini depoya dahil edeceğim. Bunun için proje klasörüne sağ tıklıyorum ve sistem menüsünden TortoiseSVN ve oradan “Import...” ögesini seçiyorum.



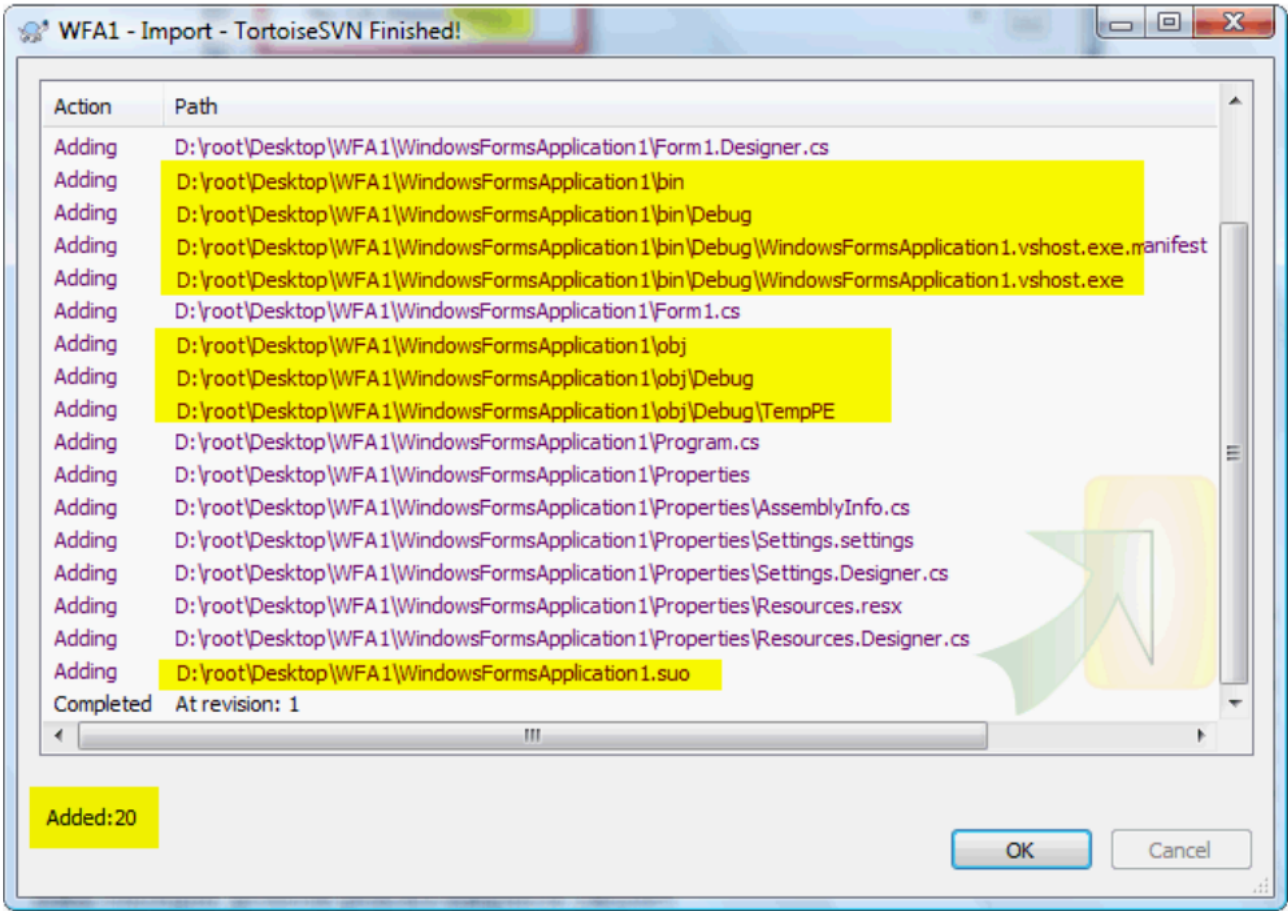
Resim 3:Import Menü Ögesi

Şimdi WFA1 isimli projemizi işletim sisteminin kurulu olmadığı farklı bir yerel sürüctide oluşturmuş olduğumuz depoya dahil edelim. Ben sabit diskimin D: bölümünde oluşturduğum repos isimli klasöre projemi dahil edeceğim.



Resim 4:Import Penceresi

Import ögesine tıkladıktan sonra açılan Import penceresinde “**URL of Repository**” *metin kutusuna yerel depo* 'mun adını yazıyorum. Burada dikkat edilmesi gereken nokta sağ tıkladığımız proje klasörünün adını da bu yola eklememiz gerektiği. Eğer eklemesek, klasör WFA1 klasörü değil, sadece içerisindeki dosyaların depoya eklenmesine sebep oluruz. Bu noktada yerel depo isimlendirmesini kullanmamın sebebi İnternet ya da yerel ağ içerisinde de depolar tanımlayıp bunlara TortoiseSVN ile ulaşabilmenizdir. Biz depomuzu kendi bilgisayarımızda tanımladığımız için yerel depo tanımını kullanıyoruz. “**Import message**” metin kutusuna da yaptığımız dahil etme işlemi açıklayan bir metin ekliyoruz. “**Ok**” butonuna bastığımızda dosyaların depoya teker teker dahil edildiğini gösteren pencere karşımıza çıkıyor.(Resim 5)



Resim 5: Dosya Import süreci

Bu iletişim penceresinde proje klasöründeki hangi dosyaların depoya dahil edildiğini görebilirsiniz. Bu noktada değinmek istediğim ufak bir nokta var. Resim-5 de sarı renk ile vurguladığım dosyalar Visual Studio 'nun derleme aşamasında kullandığı dosyalardır ve depoya gönderilmesine esasen gerek yoktur. Bu konuya yine İyi alışkanlıklar başlığı altında değineceğiz.

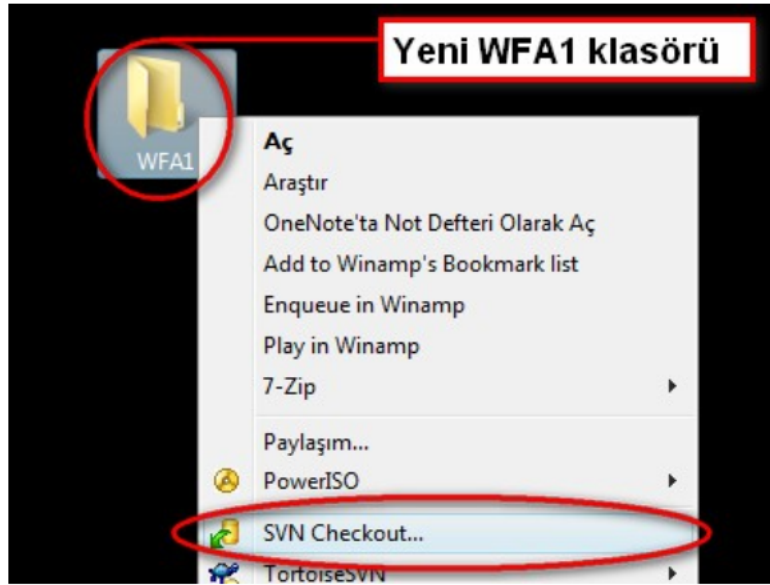
Bunlara ek olarak, dosya gönderme sürecinin sonunda, bu göndermeye ait sürüm numarasının kaç olduğu gösterilir. En altta yine sarı renkle vurgulanmış olarak kaç dosyanın depoya dahil edildiğini görebilirsiniz.

Son olarak artık proje klasörünüzü gönül rahatlığı ile silebilirsiniz. Çünkü artık projemizi depomuza gönderdik ve istediğimiz zaman bir kopyasını oluşturabiliriz.

c) Çalışma Kopyası Oluşturmak (Checkout)

Şimdi bir önceki başlıkta depoya eklediğimiz projenin bir *çalışma kopyası*'nı (Working Copy) oluşturalım. *Çalışma kopyaları* deponuzun ev sahipliği yaptığı projelerin birer yansımasıdır, kopyasıdır. Yani kullanıcılar, çalışmalarını bu kopyalar üzerinde yaparlar ve yaptıkları değişiklikleri depoya teslim ederler.(Commiting)

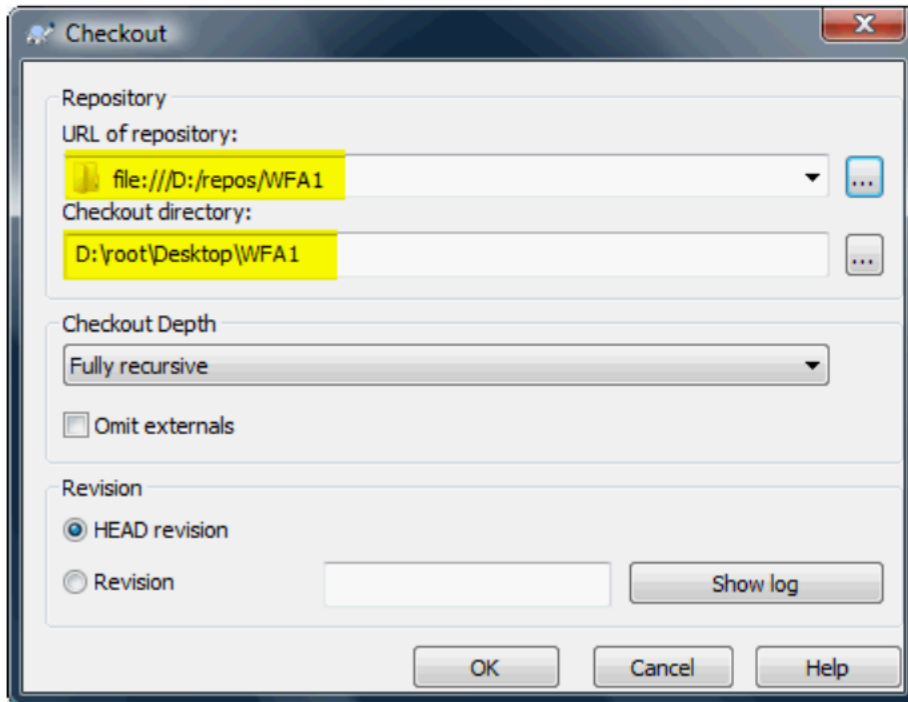
Çalışma Kopyası oluşturmak için projelerinizi takip ettiğiniz bir klasörde (Örneğin, *Projelerim Klasörü*) WFA1 isimli (İsterseniz farklı bir dosya adı verebilirsiniz) bir klasör oluşturun. Ben Masaüstümde bir klasör oluşturdum. Daha sonra bu klasöre sağ tıklayın ve **“SVN Checkout...”** ögesini seçin.(Resim-6)



Resim 6:SVN Checkout menü ögesi

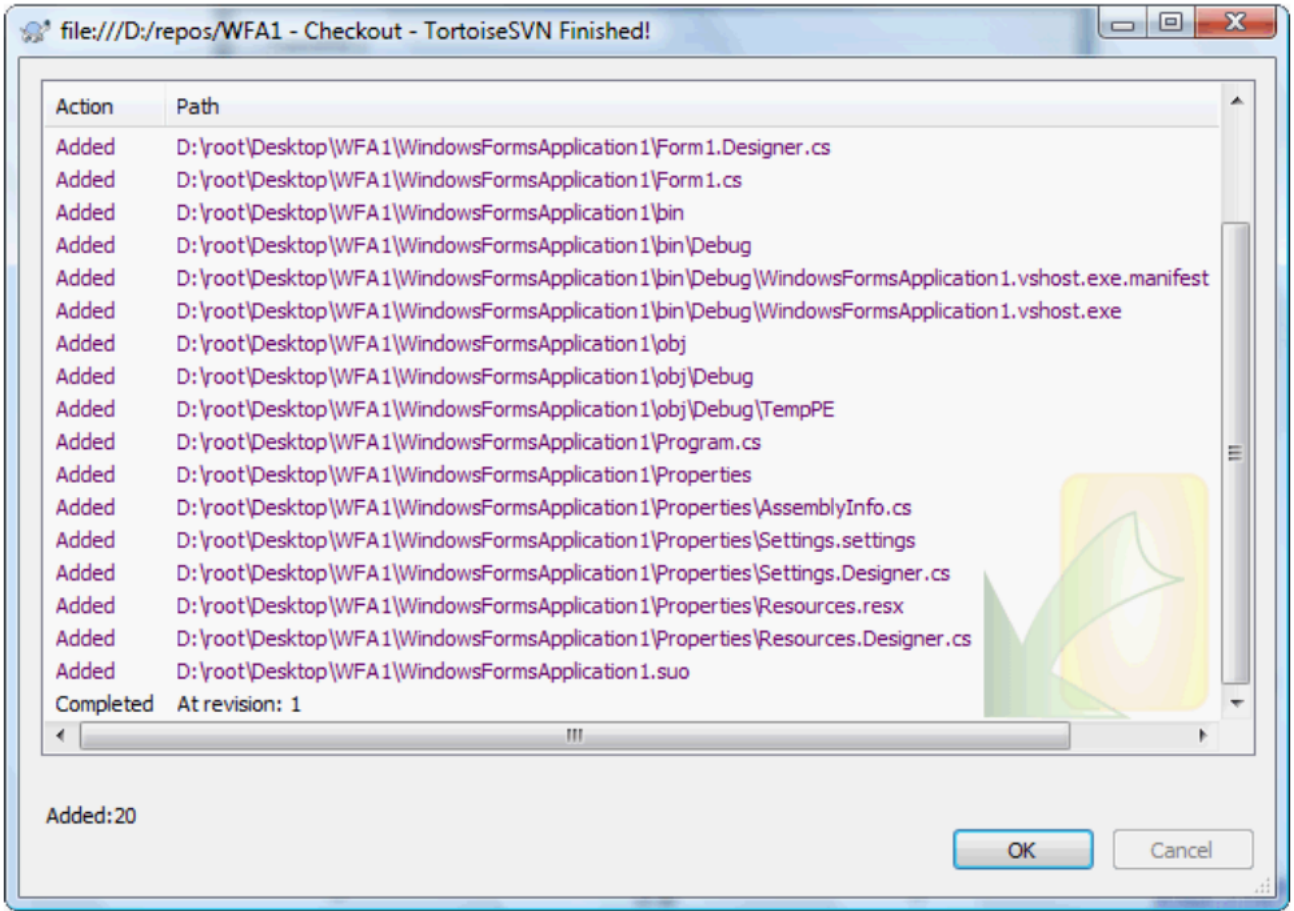
Tıkladıktan sonra karşımıza Checkout iletişim penceresi gelecek. (Resim-7) Bu pencerede “URL of Repository” kısmına daha önce Import işlemi esnasında hangi adresi yazdıysam yine aynı adresi yazıyorum. “Checkout directory” kısmında ise varsayılan olarak sağ tıkladığım dizinin adı çıkıyor.

Checkout penceresinde varsayılan revizyon “HEAD revision” yani en son yapılan değişiklikleri içeren revizyondur. Revizyon menüsünde isterseniz daha eski bir revizyonu kullanarak çalışma kopyası oluşturabilirsiniz. Eğer hangi revizyonun kopyasını oluşturacağımızı bilmiyorsanız, Show log butonunu kullanarak eski revizyonlara ait sürüm numaralarını görebilir ve seçeceğiniz revizyona karar verebilirsiniz.



Resim 7:Checkout iletişim penceresi

“Ok” butonuna tıkladığımızda depomuzdaki bütün dosyaların teker teker klasörümüze eklendiğini gösteren Checkout süreç penceresi karşımıza çıkar.(Resim-8) Gördüğümüz üzere depoya WFA1 isimli projemizi ilk dahil ettiğimizdeki tüm dosyalar, tekrar depodan *çalışma kopyamıza* indirilmiştir.



Resim 8:Checkout süreç penceresi

Çalışma Kopyası oluşturulduğunda artık klasörümüz diğer klasörlerden farklı ve özel bir klasördür. Klasörün sol alt köşesinde yeşil renkli bir çentik belirlemiştir. Bu onay çentiği bize çalışma



Resim 9:Çalışma Kopyası

kopyasındaki dosyaların artık *sürümlenmiş dosyalar* olduğunu ve kopya klasörümüzün hiçbir şekilde değiştirilmediğini gösterir. Yani bu klasör içerisindeki hiçbir dosyaya müdahalede bulunulmamıştır. Aynı yeşil renkli onay işareti, çalışma kopyası klasörünün içindeki depoya daha önce dahil edilmiş (Import) bütün klasör ve dosyalarda da vardır. Eğer projedeki

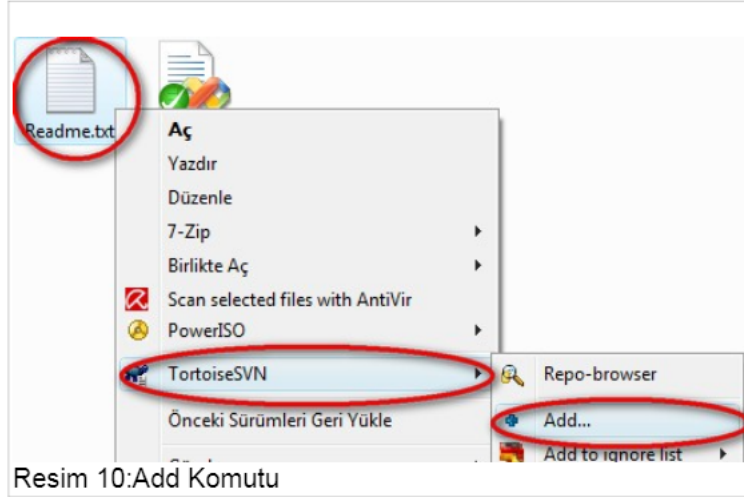
dosyaların birisi üzerinde değişiklik yaparsanız, dosya üzerindeki yeşil çentik kırmızı ünlem işaretine döndürür ve bu ünlem işareti çalışma kopyası klasörünün en üst ebeveynine kadar yansıtılır. Bu kırmızı renkli ünlem işareti çalışma kopyası klasörünüzde bir değişiklik yaptığımızın işaretidir.

Şimdi depomuza proje ekleme ve bu projenin çalışma kopyasını oluşturmayı öğrendiğimize göre ufak bir alıştırmaya bakalım. Varsayılan bir WFA2 projesi oluşturun ve bu projeyi aynı WFA1 projesini depoya eklerken yaptığımız adımları takip ederek “**D:\repos**” yerel depomuza dahil edin. Daha sonra WFA2 proje klasörünü silin ve istediğiniz bir yerde oluşturacağımız boş bir klasöre (*örneğin ,klasörün adı repos olsun*) direk yerel depo 'nun çalışma kopyasını oluşturun. Yani, Checkout işlemi esnasında “File:///D:/repos” yolunu girin. Yolun başına File:/// ön ekini eklemeniz zorunludur.

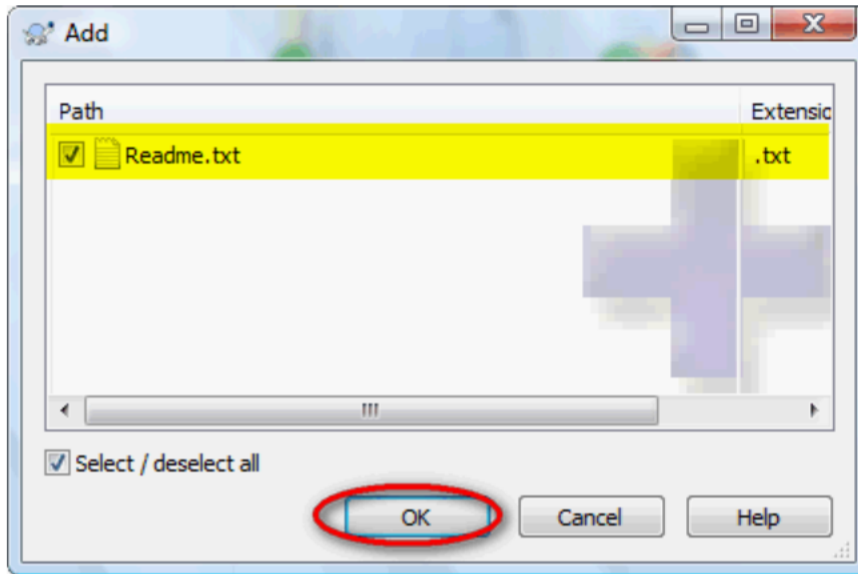
Checkout işleminden sonra oluşturduğunuz çalışma kopyasına bakacak olursanız, depodaki WFA1 ve WFA2 klasörlerinin her ikisinde bulunduğunu göreceksiniz. Yani, tek bir depo her iki projede ev sahipliği yapmaktadır. Fakat, WFA2 projesini depoya dahil ettiğinizde sürüm numarasının 2 olarak verildiğine dikkat edin. Bu her iki projenin de aynı sürüm numaralandırmasını paylaştığı anlamına gelir.

d) Çalışma Kopyasına dosya veya klasör eklemek (Add)

Halihazırdaki bir projeyi depomuza eklemeyi öğrendik. Peki depoya eklediğimiz bir projenin çalışma kopyasına yeni bir dosya eklemek istediğimizde ne yapacağız? Örneğin, WFA1 projemize Readme.txt isimli bir dosya eklemek istediğimizi farz edelim. WFA1 klasörü içerisinde dosyayı ilk oluşturduğumuzda dosyanın sol alt köşesinde yeşil renkli çentik yoktur. Çünkü dosya henüz depoya dahil edilmemiştir. TortoiseSVN 'ye dosyanın depoya dahil edilmesi gerektiğini bildirmek için, dosyaya sağ tıklanır ve TortoiseSVN menüsünden “Add...” komutu tıklanır.



Bu işlemin aynısını klasörler içinde geçerlidir. Yani projeye yeni eklediğiniz bir klasörü sağ tıklayıp yine “Add...” komutunu kullanabilirsiniz. Komuta tıkladığımızda karşımıza “Add” penceresi çıkacaktır. Bu pencerede, projeye dahil etmek istediğiniz dosyaları listelenir.



Eğer listedeki herhangi bir dosyayı *çalışma kopyası* 'na dahil etmek istemiyorsanız, sol taraftaki çentiği kaldırmanız yeterlidir. “Ok” butonuna tıkladığımızda dosyaların çalışma kopyasına eklendiğini gösteren pencerenin ardından, klasöre dönüp Readme.txt dosyasına baktığımızda, dosyanın kırmızı ünlem ile işaretlendiğini göreceksiniz. Bu çalışma kopyasının değiştiğini ve Çalışma tabanı ile aynı olmadığını göstergesidir. “Ok” butonuna bastığımızda Readme.txt dosyası depoya gönderilir ve artık üzerindeki kırmızı ünlem işareti, yeşil renkli çentik işaretine dönüşmüştür.

Peki bir klasördeki bazı özel klasörleri, proje gelişimi boyunca hiçbir şekilde teslim işlemine tabi tutmak istemiyorsunuz. Bu durum biraz sıkıntı yaratır. Çünkü yüzlerce dosyanın bulunduğu bir projeye ait çalışma kopyasında onlarca dahil edilmek istenmeyen dosya bulunabilir, ve depoya gönderme yapacağınız zamanlarda bu dosyalar listede görünerek kalabalık yaratırlar. Subversion 'un yaratıcıları bu durumu da hesaba katmışlar ve dosyaları yok saymanıza yardımcı olacak bir komut oluşturmuşlar. “[ignore](#)” komutu sayesinde istediğiniz özel dosya ve klasörlerin sürekli olarak depoya gönderilmesini engelleyebilirsiniz. Bu konuya “[Dosyaları yok saymak \(ignore\)](#)” bölümünde ayrıntılı olarak değineceğiz.

Dosya veya klasörlerinizi çalışma kopyasına ekledikten sonra depoya göndermek istiyorsanız, tek yapmanız gereken göndermek istediğiniz öğeye veya klasöre sağ tıklayıp “Commit” komutunu vermektir. Çalışma kopyasına eklenen dosyalar Değişiklikler listesinde “Added” şeklinde işaretlenmiş olarak görüleceklerdir.

Çalışma kopyasına bir ekleme yaptınız fakat, daha sonra bu eklemeyi yapmaktan vazgeçtiğiniz takdirde tek yapmanız gereken dosya ya da klasöre sağ tıklayıp yine TortoiseSVN menüsünden “Revert...” komutunu seçmektir. Ekleme yaptıktan hemen sonra dosyaya sağ tıklayıp, TortoiseSVN menüsünde “Undo Add...” menü öğesine de tıklayarak ekleme işlemi geri almanız mümkündür. “Undo Add...” komutuna tıkladığınızda yine “Revert” öğesine tıkladığınızda açılan pencerenin aynısı açılacaktır. Bu kısım için [Değişiklikleri Geri almak \(Revert\)](#) bölümüne bakın.

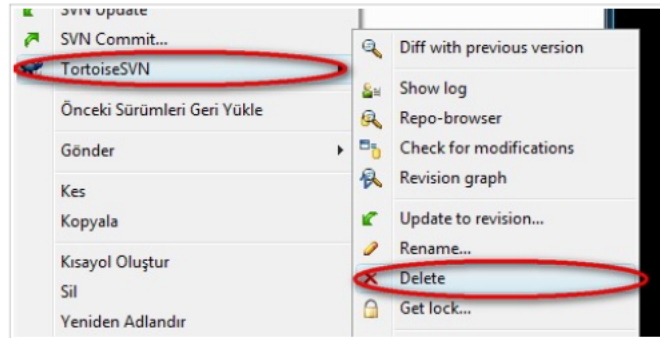
e) Çalışma Kopyasındaki bir dosyaya veya klasörü silmek (Delete)



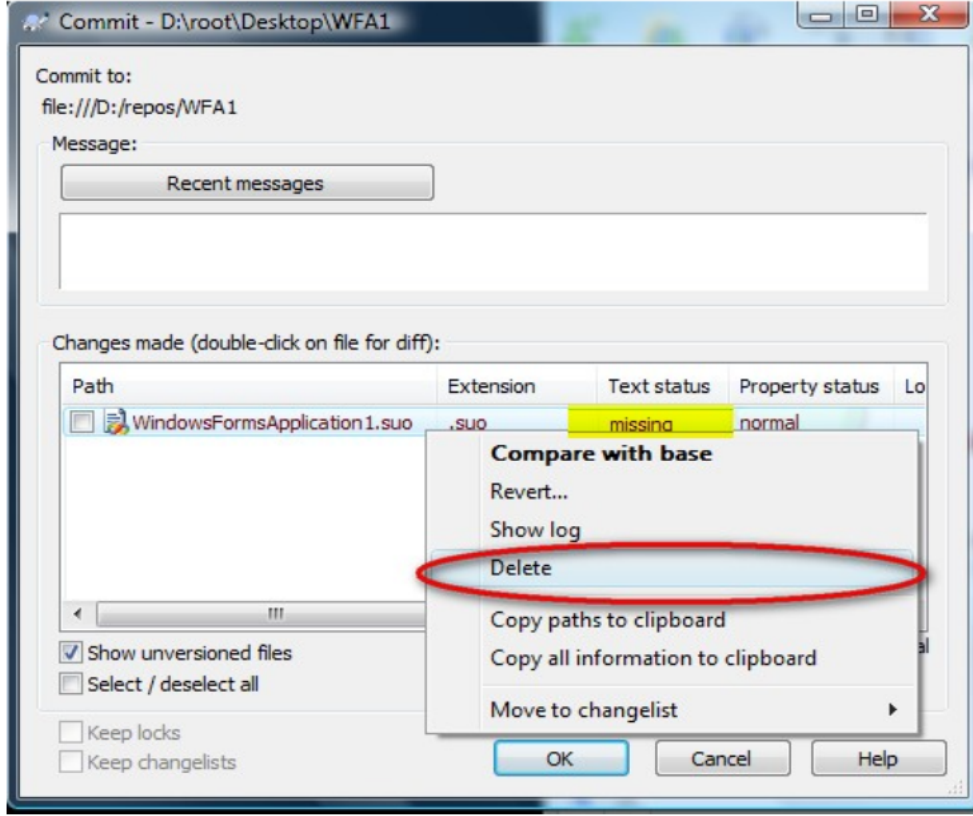
Oluşturmuş olduğunuz bir *çalışma kopyası* 'na dahil olmuş olan sürümlenmiş bir dosyayı veya klasörü silme işlemi, işletim sisteminin size sağladığından farklı bir yol ile silmeniz gerekir. Çünkü yeşil renkli çentik işaretine sahip olan sürümlenmiş dosyalar depoya dahil edilmiş dosyalardır ve TortoiseSVN ile silinmesi gerekir. TortoiseSVN yardımıyla bir dosyayı sildiğinizde, dosyanın normal bir şekilde sabit diskinizden silindiğini görürsünüz. Bir klasör sildiğinizde ise klasör sol alt köşesindeki kırmızı renkli bir çarpı ile işaretlenir. Dosyayı silmek için, silmek istediğiniz öğe sağ tıklanır, sistem menüsünden TortoiseSVN ve oradan Delete komutu seçilir.

Bu yöntem sayesinde dosya hem sabit diskinizden silinir hemde çalışma kopyanızdaki kayıtlardan silindi olarak işaretlenir. Yaptığımız bu değişikliği depoya teslim etmek (Commit) istediğinizde dosya listesinde silmiş olduğunuz dosya “silindi” olarak işaretlenmiş olduğunu görebilirsiniz.

Peki çalışma kopyanızdaki bir dosyayı yanlışlıkla TortoiseSVN menüsünden farklı bir yöntem ile sildiniz! Bu durumda depoya teslim işlemi yapmak istediğinizde (Commit) silmiş olduğunuz dosyanın “Kayıp” (Missing) olarak işaretlendiğini göreceksiniz.(Resim-11) Bu durumda dosyayı gerçekten silmek istiyorsanız, Commit penceresinde dosyayı sağ tıklayıp “Delete” komutuna tıklayarak dosyayı silindi olarak işaretlemeniz yeterlidir. Böylece dosya “Silindi” olarak işaretlenecektir.



Resim 12:Delete Komutu



Resim 13:Commit penceresi

Fakat dosyanın yanlışlıkla silindiğini düşünüyorsanız, bu durumda dosyanın silindiği klasöre ya da Çalışma kopyası klasörüne sağ tıklayıp TortoiseSVN menüsünden “Revert” komutunu seçerek, silmiş olduğunuz dosyayı geri getirebilirsiniz. (Bkz: [Değişiklikleri geri almak \(Revert\)](#))

f) Dosya veya klasörleri yok saymak (Ignore)

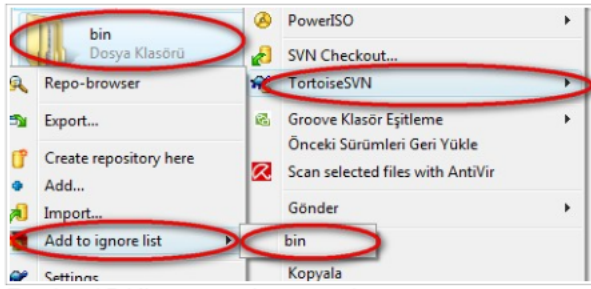
Gelelim çalışma kopyanızdaki sizin için önemli olmayan klasör ve dosyaların yok sayılmasına. WFA1 projemiz bir Visual Studio projesi olduğundan proje dizini altında /bin ve /obj uzantılı derleme klasörleri oluştururlar. Bu klasörler bizim yetkimiz dışında Visual Studio tarafından, projenin her derlenişinde otomatik olarak oluşturulduklarından ve her derlemede içerikleri değiştiğinden yerel depomuza eklenmeleri gerçekten gereksizdir. WFA1 projemizde biz ilk seferde bu klasörleri depoya dahil etmiştik. Açıkçası, bir projenin depoya bu şekilde dahil edilmesi kötü bir alışkanlıktır. İlerleyen bölümlerde projelerin depoya nasıl dahil ve teslim edilmesi gerektiğini açıklayacağım.

Şimdi deneme olması için, /bin, /obj ve .suo uzantılı dosyaların projeye dahil olmadığı yeni bir proje oluşturduğum ve bu projeyi depoya dahil (Import) ettim. Daha sonra masüstümde WFA3 isimli bir klasör oluşturup depodaki projenin kopyasını Checkout komutu ile bu klasöre indirdim. Projeyi çalıştırıp, /bin ve /obj klasörleri ile proje dizini altındaki .suo uzantılı dosyanın tekrar oluşturulmasını sağladım.



Resim 14:Add to Ignore List

Şimdi .suo uzantılı dosyanın proje dizininde yok saymak istiyorum. Bunun için dosyaya sağ tıklayıp TortoiseSVN menüsünden “Add to Ignore List” 'e geliyorum ve oradan dosya adına tıklıyorum. Eğer proje dizinindeki .suo uzantılı dosyaların tümünün yok sayılmasını istiyorsanız *.suo seçeneğini tıklamanız gerekiyor.



Resim 15:Klasörü yok saymak

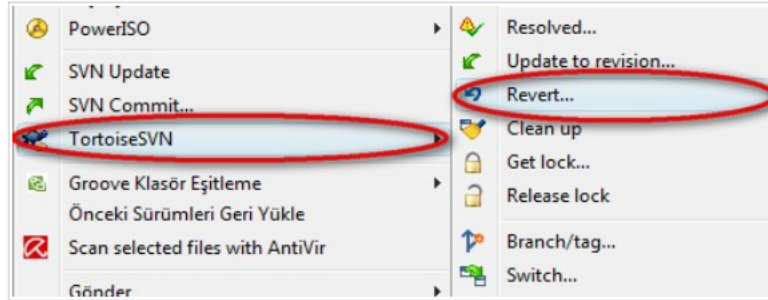
Bir klasörü yok saymak istediğinizde ise aynı dosyada yaptığımız işlemleri tekrarlamamız yeterlidir. Bu durumda “Add to Ignore List” ögesine tıkladığımızda karşınıza klasörün adı çıkacaktır.

Dosya ya da klasörleri yoksa yılanlar listesine eklediğinizde proje klasörünüzün yine sol alt köşesindeki kırmızı ünlem işareti ile mimlendiğini göreceksiniz. Bu işaret yine bize çalışma

kopyamızda değişiklik yapıldığını ve çalışma tabanımız ile aynı olmadığını gösterir. Ünlem işaretini kaldırmak için yapmamız gereken, çalışma kopyasına sağ tıklayıp “Commit” komutunu vererek değişiklikleri depoya teslim etmektir. Eğer yaptığımız değişikliklerden vazgeçtiysek, depoya teslim etmeden önce “Revert” komutunu kullanarak hepsini geri alabiliriz.

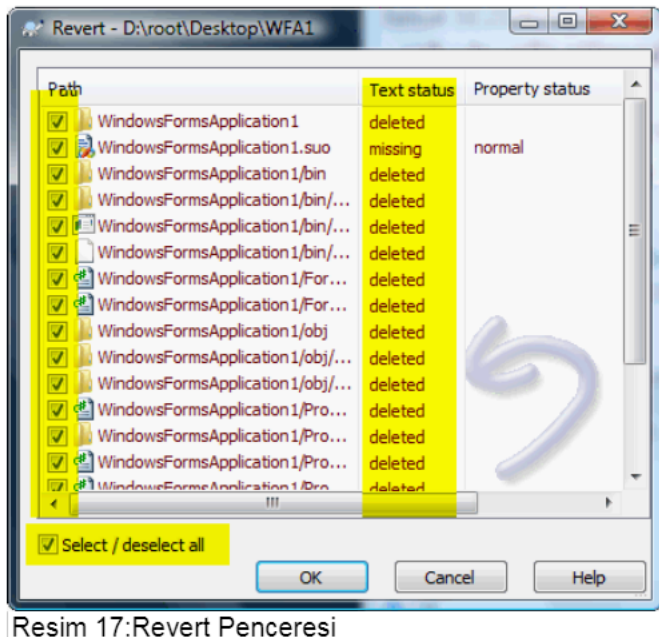
g) Değişiklikleri geri almak (Revert)

Bir dosyada değişikli yaptınız,sildiniz ya da yanlışlıkla sildiniz! Ne yapmalıyız? Bu durumda “Revert” komutu imdadımıza yetişir. Proje gelişim sürecinde TortoiseSVN 'nin en faydalı özelliklerinden birisi bence.



Resim 16:Revert komutu

Silinmiş veya değiştirilmiş olan bir dosyayı geri almak için dosyanın içinde bulunduğu klasöre sağ tıklayıp sistem menüsünden TortoiseSVN ve oradan “Revert” ögesine tıklıyoruz. Açılan pencerede dosya listesinden geri almak istediğimiz dosyaları seçerek “Ok” butonuna tıkladığımızda dosyalar değiştirilmeden önce hallerine geri dönerler.



Resim 17:Revert Penceresi

Tıkladıktan sonra “Revert” penceresi açılıyor ve bu pencerede depoya gönderilmemiş fakat üzerinde değişiklik yapılmış öğelerin listesini görüyoruz. İlk açılışta bütün öğeler işaretlenmemiş olarak geldiğinden isterseniz sadece istediğiniz öğeleri işaretleyip “Ok” butonuna basabilir ya da sol alt köşedeki “Select/deselect all” işaret kutusuna tıklayarak tümünün geri alınmasını sağlayabilirsiniz.

“Ok” butonuna tıkladıktan sonra seçtiğiniz tüm öğeler, değiştirilmedikleri ilk hallerine geri alınma sürecini gösteren iletişim penceresi açılır. Bu pencereyi kapattıktan sonra klasöre geri dönüp baktığımızda değişiklik yaptığımız dosyaların eski hallerine döndüklerini ve yine

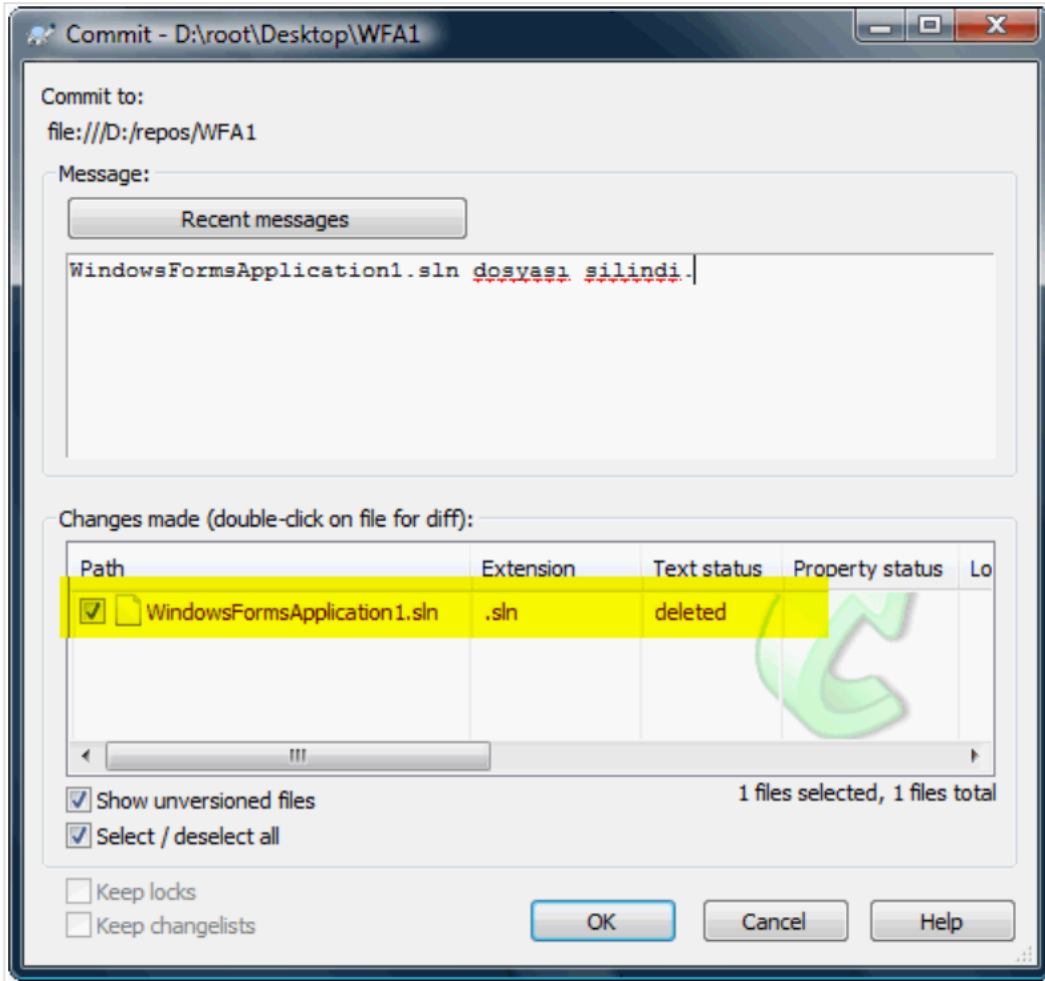
yeşil renkli çentik ile işaretlendiklerini göreceksiniz.

“Revert” komutuna isterseniz Commit penceresinden de ulaşabilirsiniz. Değiştirdiğiniz dosyaları depoya teslim etmeden önce bazı dosyaları eski haline döndürmeniz ve depoya gönderilmemesi gerektiğine karar verirsiniz, yapmanız gereken Commit penceresinde Değiştirilmiş dosyalar listesindeki bir dosyaya sağ tıklayıp Revert komutunu vermektir. Böylece dosya veya klasör eski haline döndürülür ve listeden kaldırılırlar.

h) Değişikliklerin depoya teslim edilmesi (Committing)

Peki, projemizde bir güncelleme yaptık. Bu değişiklikleri depoya nasıl yansıtacağız? Bu işleme “Committing” yani “*teslim etme*” denir. Burada yapılan çalışma kopyasında *değişikliğe* uğramış olan dosyaları depoya göndermektir. Bu cümledeki *değişiklik* terimi, dosya silmek, eklemek ve içeriğinde el ile değişiklik yapma hadiselerinin tamamını kapsamaktadır. Bu teslim işleminden sonra deponun sürüm numarası 1 arttırılır.

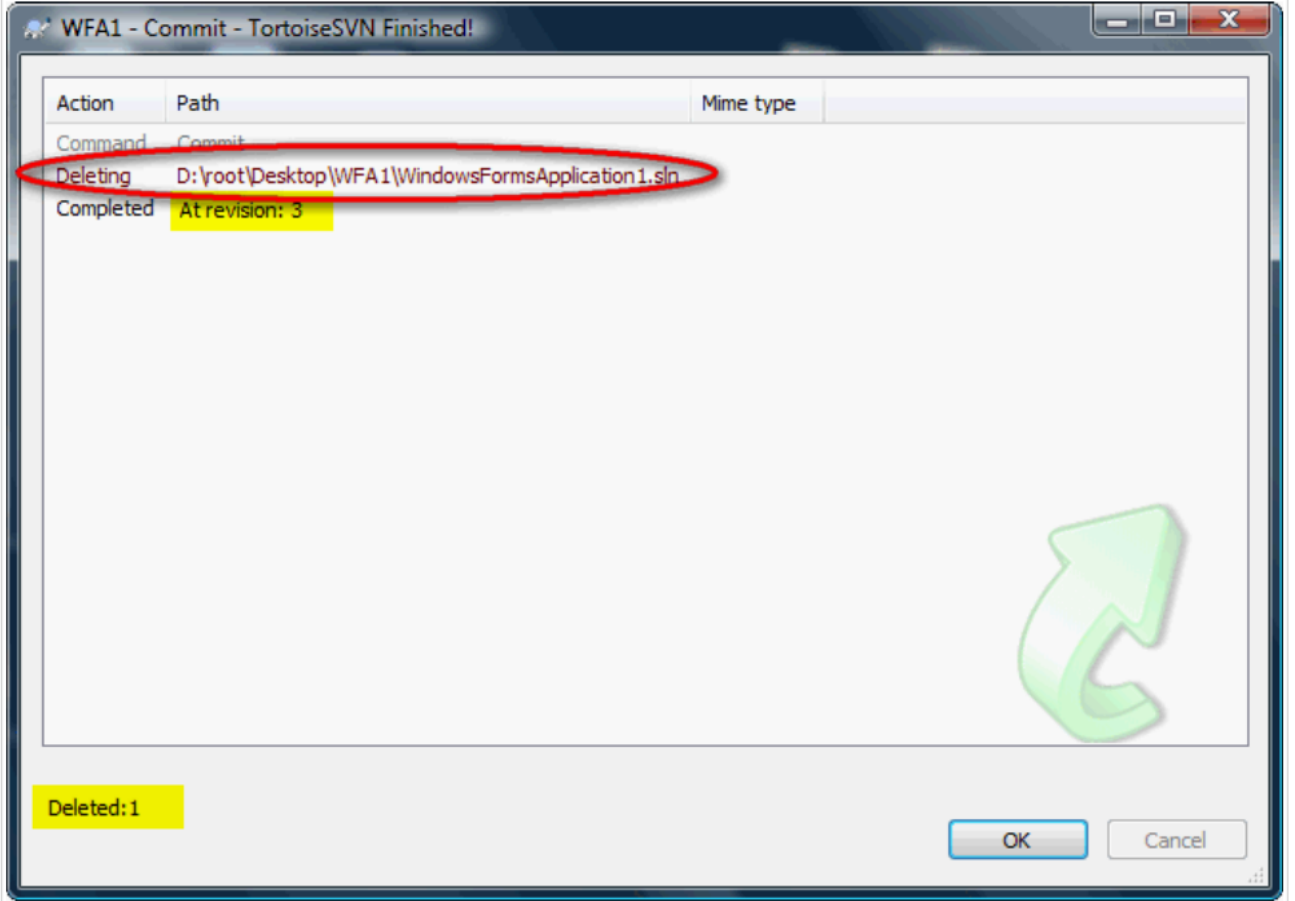
Örnek olarak, WFA1 çalışma kopyasındaki dosyalardan birisini [silelim](#) veya içindeki metinlerde değişiklik yapalım. Hangi dosya değiştirildi ise o dosyanın sol alt köşesindeki yeşil çentik işareti kırmızı ünlem işaretine döndürür. Eğer tek bir dosyayı değiştirdiysek ve depoya göndermek istiyorsak o dosyanın üzerine sağ tıklayıp sistem menüsünden “SVN Commit...” ögesine tıklamamız yeterlidir. Birden çok değişiklik yapmışsak ve dosyaların tamamını depoya göndermek istiyorsak WFA1 klasörüne sağ tıklayıp yine aynı menü ögesini seçerek de bu işlemi yapabiliriz. “SVN Commit...” ögesi tıklandığında karşımıza “Commit penceresi” açılacaktır.



Resim 18:Commit Penceresi

Commit penceresinde “Changes made” listesinde deęişiklik yapmış olduğumuz dosyaların bir listesini görebiliriz. Bu listede dosyaların hangi işleme tabi tutulduęu görülebilir. “Message” penceresi ise depoya göndermek istediğimiz deęişiklikler ile ilgili bilgileri yazmak içindir. Çoęu zaman depoya göndermeden önce dosyalarda ne gibi deęişiklikler yaptığımızı bu metin kutusuna yazmak iyi bir alışkanlıktır. Böylece ilerde herhangi bir revizyona geri dönmek istediğimizde bu metinlerden büyük ölçüde yardım alırsınız.

“Ok” butonuna tıkladığımızda deęişiklik listesindeki dosyaların depoya gönderilme sürecini gösteren dięer bir pencere karşımıza çıkar.

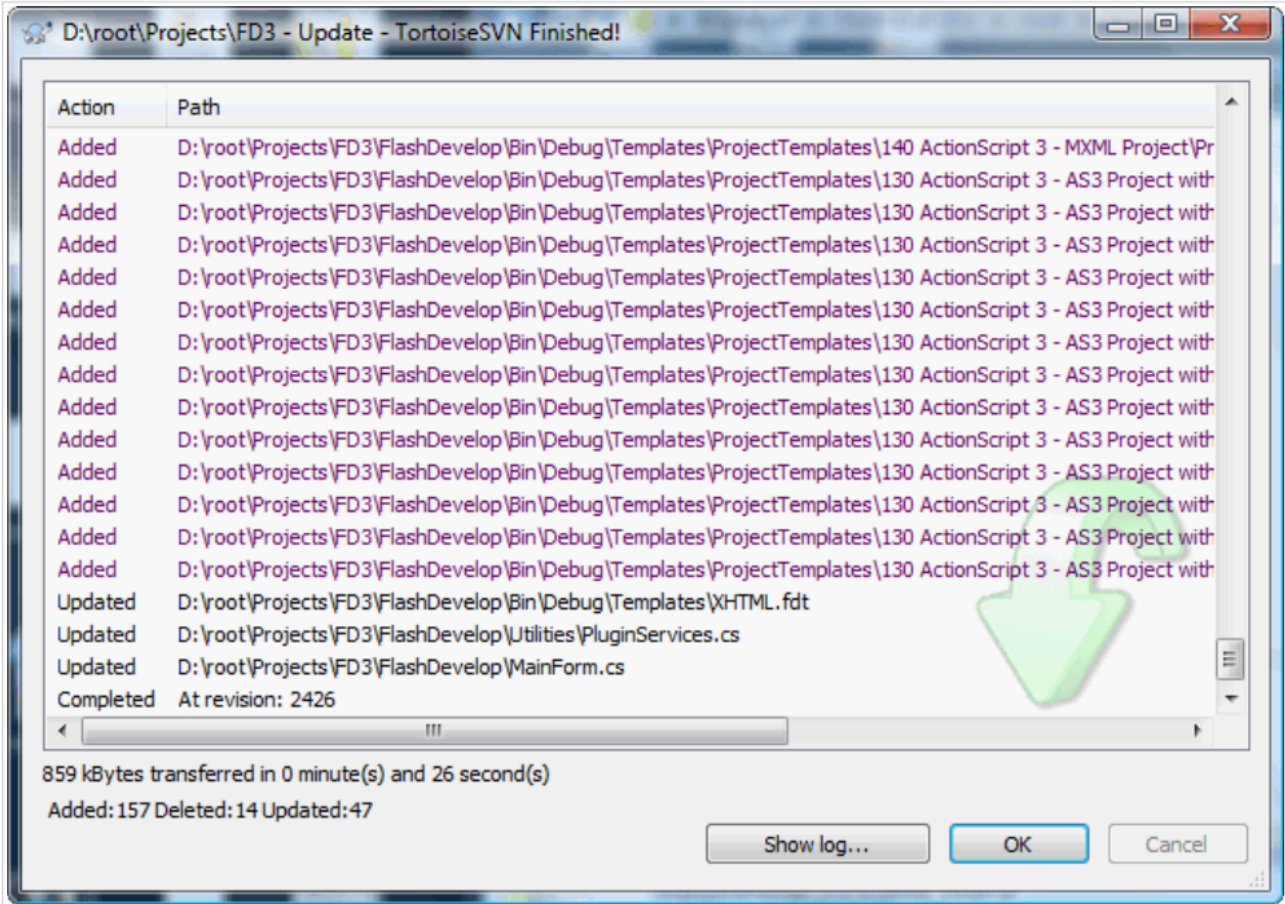


Resim 19:Depoya veri gönderme süreci

Bu pencerede deęiştirilmiş dosyaların depoya gönderilmesi sürecine tanık olursunuz. Her bir dosyanın hangi işleme tabi tutulduęu Action bölümünde yazılmaktadır. İşlem bittiğinde ise son olarak yapılan gönderme işlemine atanmış olan revizyon numarası gösterilir. Pencerenin sol alt tarafında Action bölümünde listelenen olayların istatistięi verilmektedir. Örneęin, kaç dosya silindi, deęiştirildi ya da eklendi gibi... “Ok” butonuna tıkladığımızda Commit işlemi tamamlamış oluyoruz.

i) Çalışma Kopyasını Güncellemek (SVN Update)

Şimdiye kadar yaptığımız örneklerde *yerel depo* kullandık. Fakat çoğu durumda depolar bir sunucuda tutulurlar ve birden fazla kullanıcı tarafından işlenirler. Peki biz diğer kullanıcıların depoda yaptıkları değişikliklerden nasıl haberdar olacağız? Bunun için proje üstünde çalışmaya başlamadan Çalışma kopyamıza sağ tıklayıp SVN Update komutunu vermek gerekir. Bu komuta tıkladığımızda TortoiseSVN depoya bağlanarak diğer kullanıcıların yaptığı güncellemeleri sizin çalışma kopyanıza indirir. TortoiseSVN 'nin üretilmesindeki en büyük etken de zaten budur. Böylece farklı bir çok kullanıcı uyumlu bir şekilde tek bir projeyi geliştirebilirler.

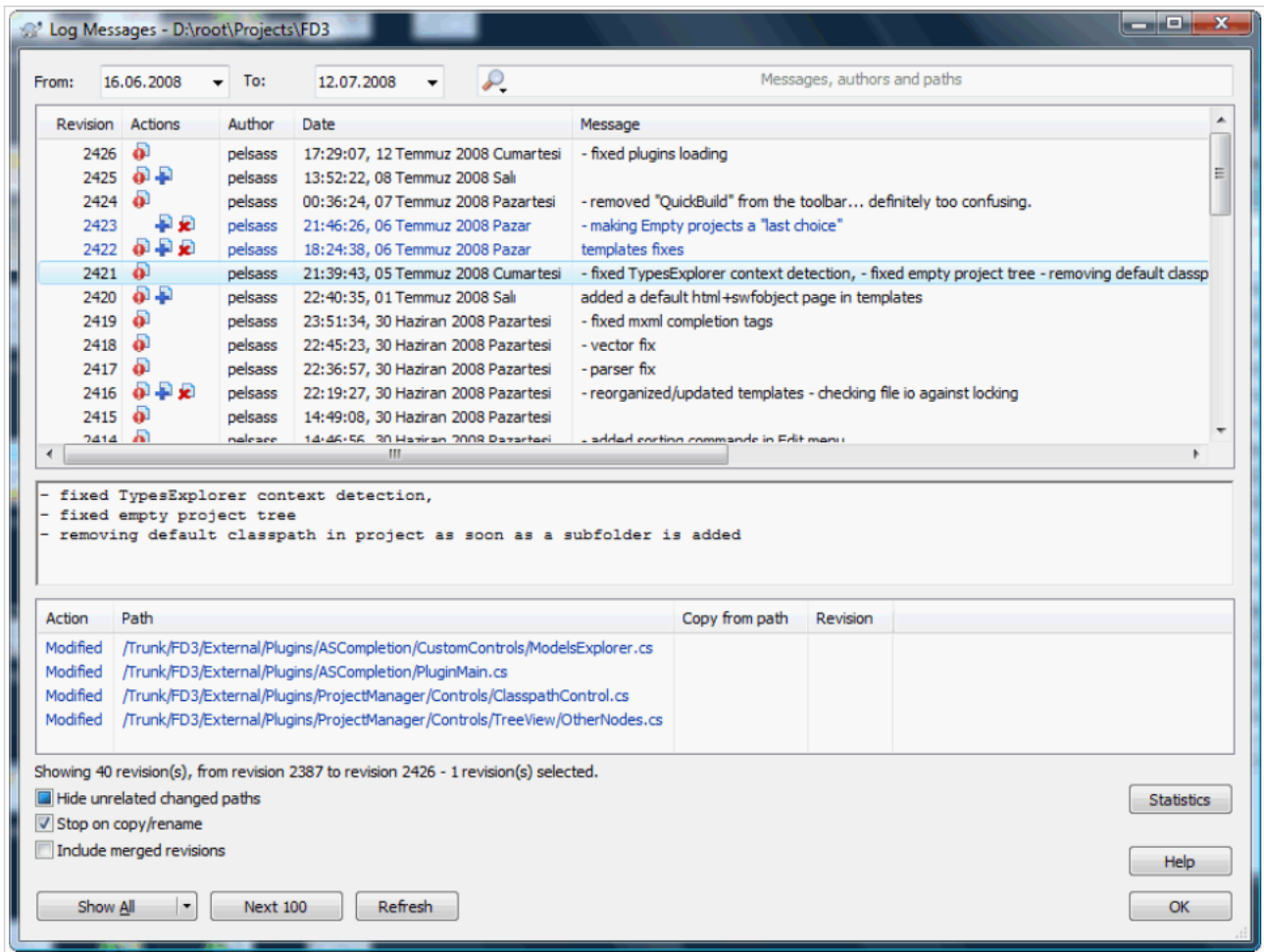


Resim 20:Çalışan Kopya Güncellemesi

“SVN Update” komutu verildikten sonra güncelleme penceresi açılır ve sunucudan indirilen dosyalar listelenmeye başlar. Action kısmında Hangi dosyanın ne işlemine tutulduğu görülebilir. Güncelleme işlemi bittiğinde “Show log...” butonu tıklanarak yapılan güncellemeler hakkında bilgi almak mümkündür.

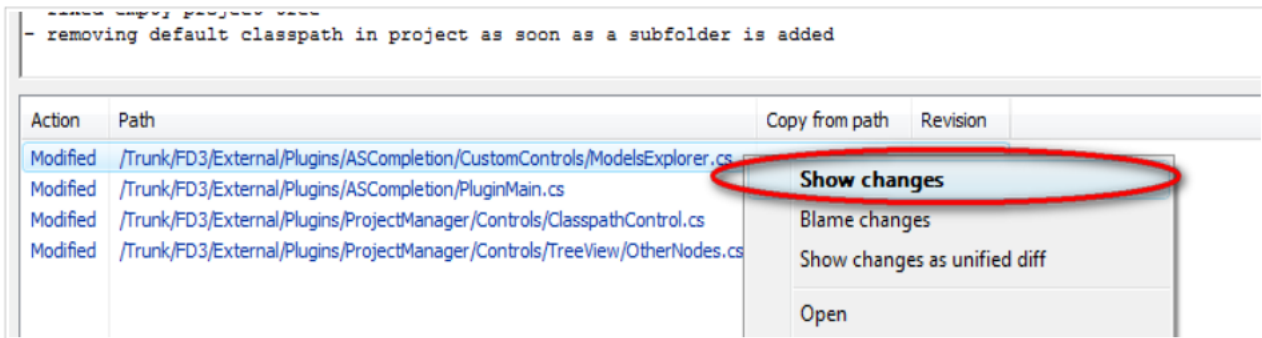
j) Depo Günlüğünü Takip Etmek (Show Log)

Günümüzde projelerin birden çok kişi tarafından yürütüldüğü düşünülürse, kimin ne yaptığını takip edebilmek açısından günlük tutmak çok önemlidir. Projede yaptığımız değişiklikleri deponuza yansıtarken (Committing) yazdığımız açıklamalar, sizin kullanıcı adınıza deponun günlük dosyalarına kayıt edilirler. Her bir depo kullanıcısının depoya gönderdiği revizyonlara ait günlük girdilerini görmek için, Çalışma kopyanıza sağ tıklayıp, TortoiseSVN menüsünden “Show Log” ögesini seçtiğinizde “Log message” penceresi açılır. (Resim-21)



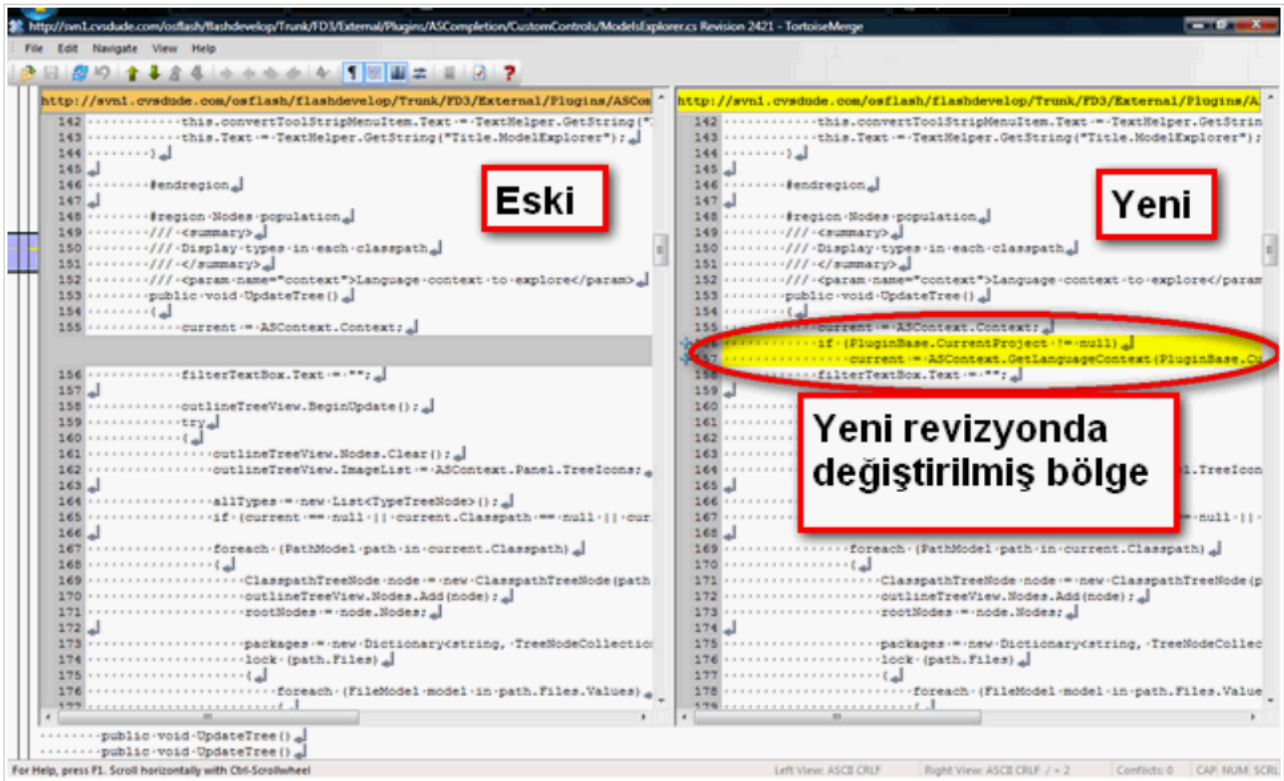
Resim 21:Günlük mesaj penceresi

“Log Message” penceresinde En üstte **From:** ve **To:** tarih kutularından istediğimiz tarih aralığını seçerek, o aralıktaki günlük girdilerini görebiliriz. Üst sağ taraftaki arama kutusuna ise bir kullanıcı adı ya da metin yazarak, belli kriterlere göre girdi araması yapabiliriz. Ortadaki metin kutusunda girdiye ait mesaj metni görülmektedir. Onun altındaki listede ise değiştirilmiş dosyaların bir listesini görüyoruz.



Resim 22:Show Changes

Bu listedeki bir dosyaya sağ tıklayıp açılan pencereden “Show Changes” ögesine tıklarsak, o dosyaya ait bir önceki revizyon ile şu andaki revizyon arasındaki farkı gösteren “TortoiseMerge” uygulaması çalışır ve iki revizyon arasındaki farkı gösterir. Böylece kullanıcının o dosya üzerinde ne gibi bir değişiklik yaptığını anlayabiliriz.



Resim 23:TortoiseMerge Penceresi

k) Çalışma kopyasını temizlemek (Cleanup)

Sunucu problemleri sebebiyle bir Subversion komutu başarıyla tamamlanamadıysa, çalışma kopyanız tutarsız durumda kalmış olabilir. Bu durumda tutarsız durumdaki çalışma kopyasına sağ tıklayıp TortoiseSVN menüsünden Cleanup komutu kullanılmalıdır.

Cleanup komutunun bir diğer kullanışlı özelliği de, eğer bir dosyanın tarihi değişmiş, fakat içeriği değişmemişse, Subversion, dosyanın önceki kopyasıyla byte-byte karşılaştırma yapmadan gerçekten değiştirilmiş olup olmadığını söyleyemez. Eğer tutarsız durumda kalmış olan birçok dosyanız varsa, bu durumda, TortoiseSVN' nin çalışma kopyasının durumunu belirleme işlemi çok yavaşlar ve iletişim formlarının cevap verme süresi yavaşlar. Cleanup komutunu, çalışma kopyası üzerinde çalıştırmak, bu bozuk olan zaman damgalarını düzeltir ve durum kontrolünü tam hıza ulaştırır. (TortoiseSVN – Help / 5.16 Cleanup)

l) Çakışmaları düzeltmek (Resolving Conflicts)

Çakışma olayı, eğer aynı projede çalışan iki geliştirici bir dosyadaki aynı satırları değiştirdiğinde ortaya çıkar. A geliştiricisi file.txt dosyasındaki söz konusu satırları değiştirir ve depoya teslim eder. B geliştiricisi de aynı zamanda depodan güncelleme yapmadan aynı satırları değiştirmişse, depoya teslim etmek istediğinde çakışma olayı meydana gelir. Subversion projeniz hakkında bir bilgiye sahip olmadığı için, çakışmanın düzeltilmesini geliştiriciye bırakır. Eğer bir çakışma rapor edildiğinde, şüpheli olan dosya açılmalı ve <<<<<<<< ile başlayan satırlar aranmalıdır. Çakışmalar aşağıdaki şekilde mimlenir.

```

<<<<<<< filename
      your changes
=====
      code merged from repository
>>>>>>> revision

```

Ayrıca, Subversion her bir çakışmış dosya için, halihazırdaki dizine 3 adet dosya ekler.

filename.ext.mine :

Bu çalışma kopyanızı güncellemeden önce çalışma kopyanızda mevcut olan dosyadır ve çakışma mimleri yoktur. Bu dosya içinde son yaptığınız değişiklikleri barındırır.

filename.ext.OLDREV :

Bu, çalışma kopyanızı güncellemeden önceki TABAN revizyonu işaret eden dosyadır. Başka bir deyişle, dosyada son düzenlemelerinizi yapmadan önce depodan indirdiğiniz dosyadır.

filename.ext.NEWREV :

Çalışma kopyanızı güncellediğiniz sırada bilgisayarınızda kurulu Subversion uygulamasının sunucudan aldığı dosyadır. Bu dosya, deponun ana revizyonundaki (HEAD revision) dosyaya karşılık gelir.

Çakışmayı çözmek için harici ya da TortoiseSVN-Conflict Editor 'ü çalıştırabilir veya başka herhangi bir editör kullanabilirsiniz. Kodların nasıl görünmesi gerektiğine karar verip, gerekli değişiklikleri yaparak dosyayı kaydetmelisiniz.

Çakışmayı çözdükten sonra, TortoiseSVN – Resolved komutunu çalıştırıp yaptığımız değişiklikleri depoya teslim edebilirsiniz. Burda önemli olan nokta, Resolve komutu gerçekte çakışmaları çözmez. Çakışmayı sizin çözmeniz gerekir. Resolve komutu, sadece, yaptığımız değişiklikleri depoya teslim edebilmeniz için *filename.ext.mine* ve *filename.ext.r** olan dosyaları klasörden kaldırır.

Eğer ikili formatta çakışan dosyalarınız var ise, Subversion dosyaları birleştirmeye çalışmaz. Yerel dosya değiştirilmeden (Sizin son yaptığımız değişikliklerle birlikte) kalır ve dizinde sadece *filename.ext.r** dosyaları oluşturulur. Eğer kendi yaptığımız değişikliklerden vazgeçmek ve depodaki sürümü tutmak isterseniz, sadece Revert komutunu kullanın. Eğer kendi sürümünüzü tutmak depodaki sürümün üzerine yazmak isterseniz, Resolved komutunu kullanın ve çalışma kopyanızı depoya teslim edin.

Eğer ebeveyn klasöre sağ tıklayıp TortoiseSVN - Resolved komutunu kullanırsanız, klasör içindeki çakışma olan çoklu dosyalar için çözümlene yapabilirsiniz. Bu komut size dizindeki çakışma olan dosyaların bir listesini içeren pencereyi gösterecektir. Bu pencerede istediğiniz dosyaları işaretleyerek, çakışmayı çözümlendi olarak işaretleyebilirsiniz.

3. İyi Alışkanlıklar

3.1. Depo klasör düzeni

Daha önce bir projenin depoya nasıl dahil edileceğini gördük. Ancak program geliştirme sürecinin iyi takip edilmesi ve Subversion' un birçok kullanışlı özelliklerinden hat safhada faydalanabilmek için projemizi depoya göndermeden önce projenin içinde barınacağı özel bir klasör düzeninin depoya dahil edilmesi gerekir.

Bir depoyu düzenlemek için tavsiye edilen bazı standart yollar vardır. Çoğu insan, projesinin ana gelişim sürecini takip etmek için “trunk”, yan kopyalarını takip etmek için “branches” ve Etiketlenmiş kopyaları takip etmek için de “Tags” isimli klasörler oluştururlar. Eğer deponuz sadece bir tek proje içeriyorsa, insanlar genellikle aşağıdaki üst seviye dizinlerini oluştururlar.

```
/trunk  
/branches  
/tags
```

Eğer deponuz birden fazla proje barındıracaksa, insanlar genellikle düzenlerini genel şemayı baz alarak kurarlar.

```
/trunk/paint  
/trunk/calc  
/branches/paint  
/branches/calc  
/tags/paint  
/tags/calc
```

Ya da projeye göre dizinleme yaparlar.

```
/paint/trunk  
/paint/branches  
/paint/tags  
/calc/trunk  
/calc/branches  
/calc/tags
```

Eğer projeler yakinen alakalı değilse ve üzerlerinde ayrı ayrı çalışılıyorsa, projeye göre dizinleme yapmak daha anlamlıdır. Deponun kopyasını tek seferde almak isteyeceğiniz ya da tek bir dağıtım paketine bağlı ilişkili projeler için temel şemaya göre düzenleme yapmak daha iyidir. Böylece herhangi bir depoyu indirmek istediğinizde tek bir “*trunk*” dizininiz olacak ve alt projeler arasındaki ilişkiler daha anlaşılır olacaktır.

3.2. Projeniz üzerinde çalışmaya başlamadan önce güncelleyin.

Çalışma kopyanız üzerinde işlem yapmaya başlamadan önce deponuzdan güncelleme yapın. Böylece haberiniz olmadan diğer geliştiriciler tarafından depoya yapılmış olan teslim işlemlerinden haberdar olmuş ve mümkün mertebe çakışma durumuna maruz kalmamış olursunuz.

3.3. Çalıştığınız bölümleri diğer kullanıcılara bildirin.

Projede bir bölüm üstünde çalışma yaparken diğer aktif geliştiricilere ne üzerinde çalışacağınız konusunda bilgi verin. Böylece diğer kullanıcılar sizin ilgilendiğiniz bölümlere ait dosyalarda eğer çok gerekli değilse değişiklik yapmazlar. Bu sayede, yine sıkıntılı bir durum olan çakışma durumuna düşmemiş olursunuz.

3.4. Teslim işlemi yaparken daima açıklayıcı ve kısa notlar yazın.

Projenizde bir değişiklik yaptınız ve depoya teslim edeceksiniz. Kendinize bir not alma düzeni

seçin ve dosyalarınızı depoya teslim ederken “Message” metin kutusuna sürekli bu düzene göre kısa ve **açıklayıcı** notlar alın. Genel ifadeler kullanmaktan kaçının. Belirleyici olun. 3 ay sonra deponuzun günlük kayıtlarımıza baktığımızda ben bu teslim işlemini neden yapmışım diye düşünmeyin.

Örneğin;

- *Projeye dosyalar eklendi.*
- *Bazı doyalar düzenlendi ve silindi.*

gibi genel ifadeler kullanmak yerine;

- Main.csproj/MainForm.cs dosyasında program minimize olarak başlatıldığında menünün görüntülenmesini engelleyen bir hata düzeltildi.
- Projede resim küçültme, icon dosya tipine çevirme, grileştirme gibi bitmap işlemlerinde kullanılmak üzere MyProject.Core.csproj/ImageManager.cs : dosyası eklendi.
- Uyumluluk için MenuYoneticisi.cs dosyası MenuManager.cs olarak değiştirildi.
- LoginForm.cs formu projeden kaldırıldı. Yerine LoginUI.cs grafik arayüzü eklendi ve arayüzün StandartForm.cs formuna otomatik yüklenmesi sağlandı.

ifadeler kullanın. Böylece hem diğer geliştiriciler yaptığımız değişikliklerden haberdar olurlar hemde ileriki bir zamanda projeden sildiğiniz ya da değişiklik yaptığımız bir dosyayı tekrar incelemek istediğinizde günlük kayıtlarında arama yaparak kolayca dosyanın silinmediği sürümüne ulaşabilirsiniz.

3.5. Çalışmanızı bitirmeden depoya teslim işlemi yapmayın.

Sizin tam anlamıyla tamamlamadığımız eksikliklerden dolayı diğer geliştiricilerin projeye devam edememesi ne kadar hoş olur? :) Bu yüzden, özellikle programlamanın gerektirdiği projelerde üzerinde çalıştığımız kodları mutlaka tamamlayın. Eğer vaktiniz yok ise, projenin derleme sürecine engel olmayacak biçimde kodlarınızı bağlayın ve depoya teslim penceresinde neleri eksik bıraktığımızı ve neleri tamamlayacağımızı **açıklayıcı** biçimde kısa notlar düşünün. Unutmayın! Bu ufak notlar hem sizin, hem de diğer geliştiriciler için.

3.6. Sürümlenmemiş dosyaları unutmayın.

Projenizi geliştirirken mutlaka yeni belgeler ekleyeceksiniz. Örneğin Visual Studio ile çalışırken bir yeni bir form oluşturduunuz, ya da yeni bir Class dosyası. Ya da bir forma ImageCollection eklediniz ve Visual Studio sizin haberiniz olmadan forma bir Resource.resx dosyası bağladı. Bu tür durumları sürekli aklınızda bulundurun ve sürekli kontrol edin. Ya bir dosyayı Visual Studio ile ekledikten hemen sonra explorer ortamında sağ tıklayıp *TortoiseSVN-Add...* komutunu kullanın ya da teslim penceresinde (Commit) “*Show Unversioned Files*” işaret kutusunu tıklayarak çalışma kopyanızdaki sürümlenmemiş dosyaları görün. Eğer listede sürümlenmemiş fakat depoya gönderilmesi gerekli bir dosya görürseniz, tek yapmanız gereken dosyaya sağ tıklayıp “*Add*” komutunu kullanmak. Böylece dosya depoya teslim edilecek şekilde işaretlenir.

4. Kaynaklar

- TortoiseSVN Help Documents
- <http://www.mutasyon.net/makaleoku.asp?id=774>
- [http://blog.pragmasql.com/page/Subversion-Kullanc4b1mc4b1-\(TurkceTurkish\).aspx](http://blog.pragmasql.com/page/Subversion-Kullanc4b1mc4b1-(TurkceTurkish).aspx)
- <http://tortoisesvn.tigris.org/faq.html>

5. İletişim

İsmail SEZEN

sezenismail@gmail.com

Lütfen, belgeye eklenmesinin iyi olacağını düşündüğünüz diğer konuları, yanlış olduğunu düşündüğünüz bölümleri ya da imla hatalarını bildirin. Teşekkürler.

6. GNU Özgür Belgeleme Lisansı

GNU Free Documentation License
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that

work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple

HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have

printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities

- responsible for authorship of
the modifications in the Modified
Version, together with at least five of the principal authors of the
Document (all of its principal authors, if it has fewer than five),
unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the
Modified Version, as the publisher.
 - D. Preserve all the copyright notices of the Document.
 - E. Add an appropriate copyright notice for your modifications
adjacent to the other copyright notices.
 - F. Include, immediately after the copyright notices, a license notice
giving the public permission to use the Modified Version under the
terms of this License, in the form shown in the Addendum below.
 - G. Preserve in that license notice the full lists of Invariant Sections
and required Cover Texts given in the Document's license notice.
 - H. Include an unaltered copy of this License.
 - I. Preserve the section Entitled "History", Preserve its Title, and add
to it an item stating at least the title, year, new authors, and
publisher of the Modified Version as given on the Title Page. If
there is no section Entitled "History" in the Document, create one
stating the title, year, authors, and publisher of the Document as
given on its Title Page, then add an item describing the Modified
Version as stated in the previous sentence.
 - J. Preserve the network location, if any, given in the Document for
public access to a Transparent copy of the Document, and likewise
the network locations given in the Document for previous versions
it was based on. These may be placed in the "History" section.
You may omit a network location for a work that was published at
least four years before the Document itself, or if the original
publisher of the version it refers to gives permission.
 - K. For any section Entitled "Acknowledgements" or "Dedications",
Preserve the Title of the section, and preserve in the section all
the substance and tone of each of the contributor acknowledgements
and/or dedications given therein.
 - L. Preserve all the Invariant Sections of the Document,
unaltered in their text and in their titles. Section numbers
or the equivalent are not considered part of the section titles.
 - M. Delete any section Entitled "Endorsements". Such a section
may not be included in the Modified Version.
 - N. Do not retitle any existing section to be Entitled "Endorsements"
or to conflict in title with any Invariant Section.
 - O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or
appendices that qualify as Secondary Sections and contain no material
copied from the Document, you may at your option designate some or all
of these sections as invariant. To do this, add their titles to the
list of Invariant Sections in the Modified Version's license notice.
These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number.

Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this

License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4.

Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.