

バージョン管理システム入門 (初心者向け)

Subversionの 「trunk」 「branches」 「tags」 の使い方

Subversionを使いこなせていますか？自由度の高いSubversionは、慣例に従うことも多く、慣例を理解してうまくリポジトリを構築する必要があります。「trunk」「branches」「tags」もその中のひとつです。作ったばかりのリポジトリはまっさらなので、まずはこの3つを作成するとよいでしょう。

この記事では、解説と合わせて、チュートリアル形式で「trunk」「branches」「tags」の作成を行っていきます。ぜひあなたも一緒にやってみてください。

trunk (トランク) は、開発の中心 (メインライン) となるディレクトリで、ほとんどのファイルをここに格納します。通常は開発中の最新バージョンを格納しておきます。Gitでいう、masterブランチのようなものです。後述するbranchesのひとつと考えることもできます。最低限トランクさえあれば開発を進めていくことができますので、初心者の方はまずここからはじめてみましょう。

Subversionのインストール

まだSubversionをインストールしていない方は、下記の公式サイトから、各プラットフォーム用のバイナリパッケージをダウンロードして、インストールしておいてください。

- バイナリパッケージのダウンロード : <https://subversion.apache.org/packages.html>

新しいリポジトリの作成

それでは、はじめに新しいリポジトリを作成しましょう。それには、コマンドプロンプト（コンソール）から、次のコマンドを入力します。ここでは、リポジトリ名を「tuto-repo」としています。なお、カレントディレクトリは適宜任意のディレクトリに移動しておいてください。

```
svnadmin create tuto-repo
```

コマンドを実行すると、新しいリポジトリが作成されます。まだ中身はなにもありません。

trunkを作ってみよう

trunkの作成

次に、このリポジトリの中にtrunkディレクトリを作ってみましょう。次のコマンドを使って、ディレクトリの作成とコミットを同時に行います。なお、リポジトリのパスは適宜読み替えてください。形式が一般的なパス表記と異なっていることに注意が必要です。

```
svn mkdir file:///C:/Users/Test/Desktop/tuto-repo/trunk -m trunkの作成
```

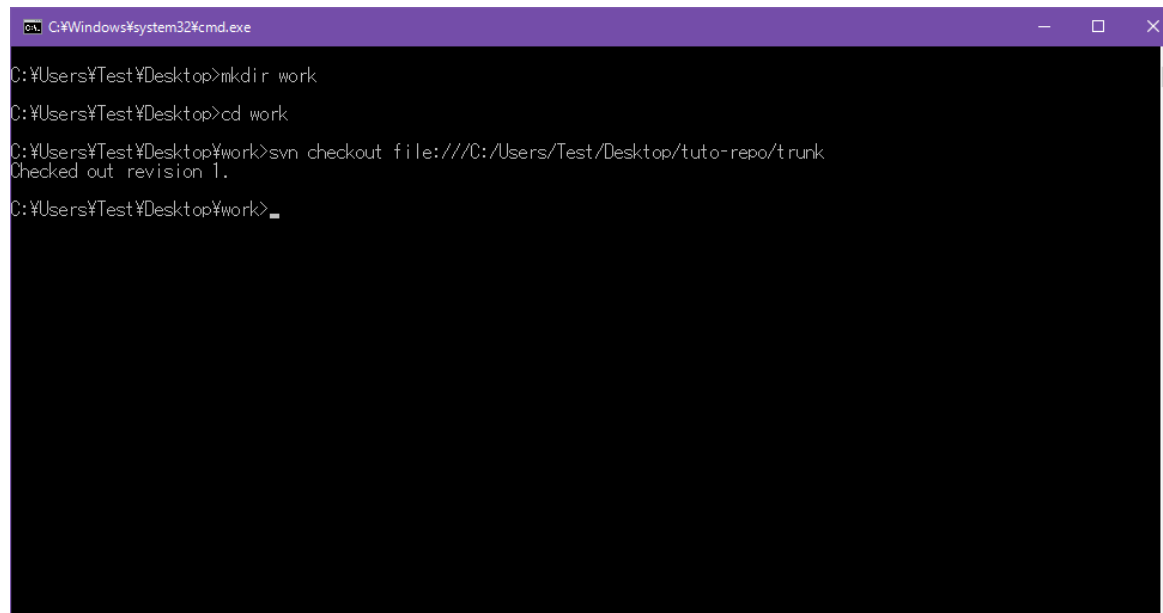


```
C:\Windows\system32\cmd.exe
C:\Users\Test\Desktop>svn mkdir file:///C:/Users/Test/Desktop/tuto-repo/trunk -m trunkの作成
Committed revision 1.
C:\Users\Test\Desktop>
```

trunkのチェックアウト

それでは、今しがた作成したtrunkにコミットしてみましよう。そのためには、まず作業コピーをチェックアウトする必要があります。「work」という名前の新しいディレクトリを作成して、そこに作業コピーをチェックアウトしましょう。

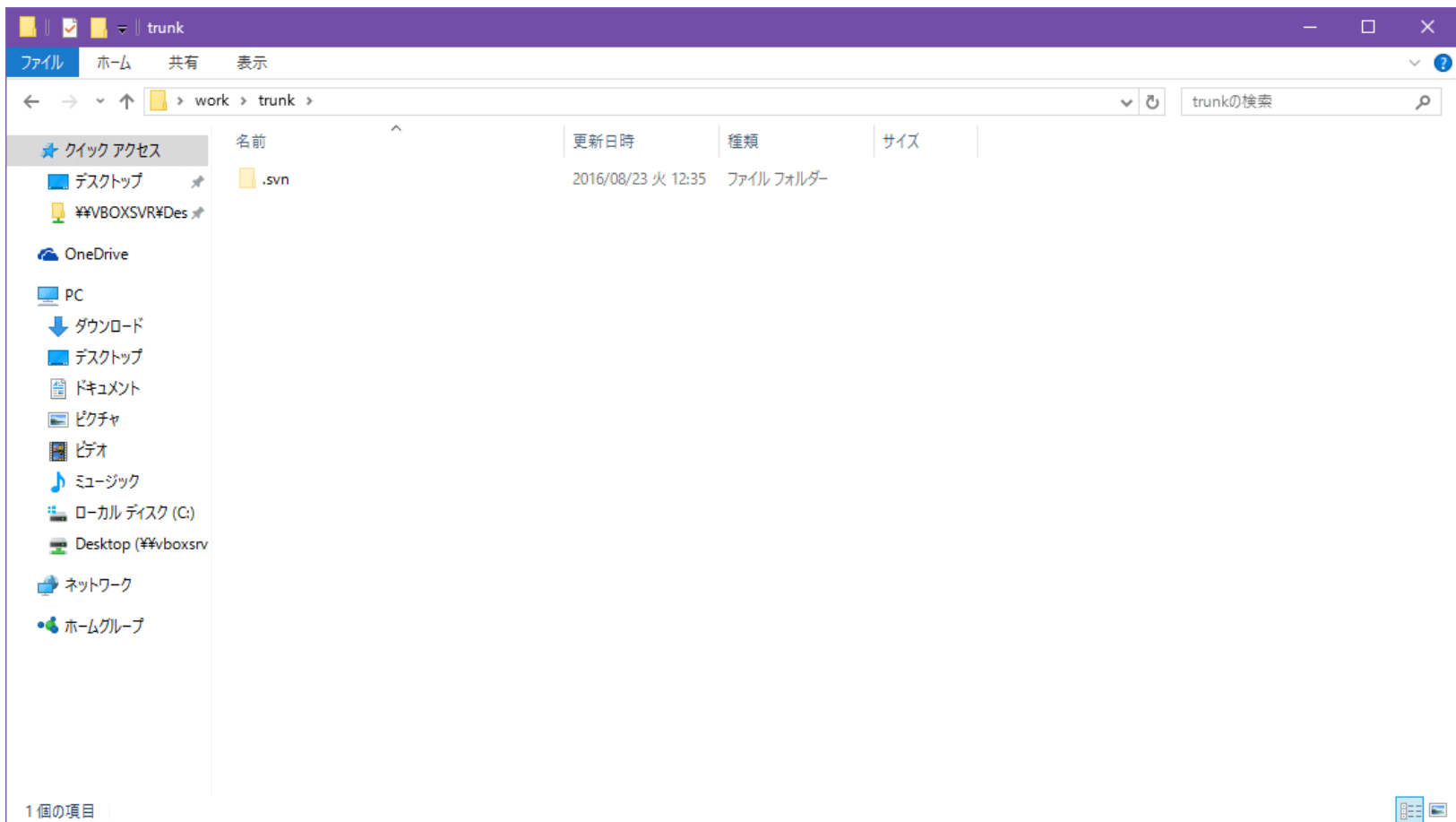
```
mkdir work  
cd work  
svn checkout file:///C:/Users/Test/Desktop/tuto-repo/trunk
```



```
C:\Windows\system32\cmd.exe  
C:\Users\Test\Desktop>mkdir work  
C:\Users\Test\Desktop>cd work  
C:\Users\Test\Desktop\work>svn checkout file:///C:/Users/Test/Desktop/tuto-repo/trunk  
Checked out revision 1.  
C:\Users\Test\Desktop\work>
```

trunkを作ってみよう

これで作業コピーのチェックアウトができました。「work」ディレクトリを確認すると、次のように空の「trunk」ディレクトリ（.svn隠しディレクトリはあります）が作られていますね。



trunkを作ってみよう

trunkへのコミット

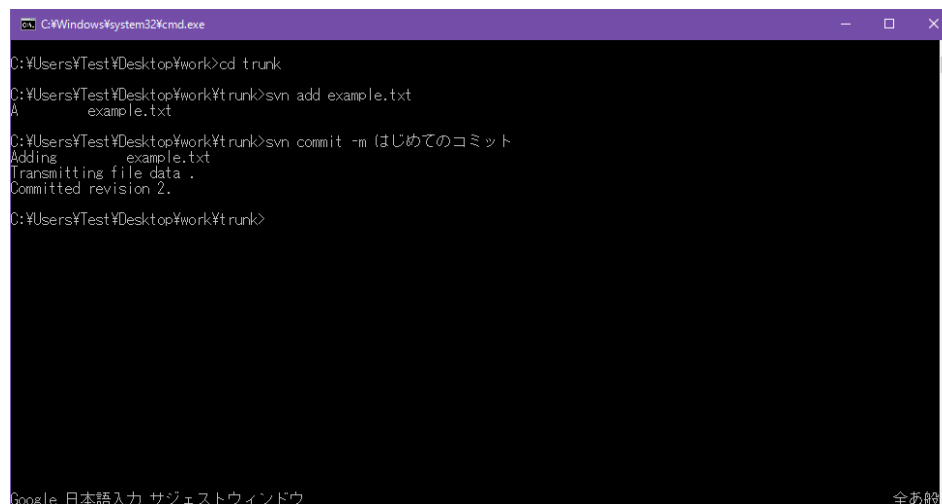
次に、下記の内容の「example.txt」を「trunk」ディレクトリ内に作成してください。

```
hogehoge  
fugafuga
```

ファイルを保存したら、次のコマンドを使ってファイルをコミットしましょう。

```
cd trunk  
svn add example.txt  
svn commit -m はじめてのコミット
```

これでうまくコミットできましたね。
このように、コミットを重ねていく
ことで開発を進めていきます。次は、
このリポジトリに「branches」を
作ってみましょう。



```
C:\Windows\system32\cmd.exe  
C:\Users\Test\Desktop\work>cd trunk  
C:\Users\Test\Desktop\work\trunk>svn add example.txt  
A  
example.txt  
C:\Users\Test\Desktop\work\trunk>svn commit -m はじめてのコミット  
Adding example.txt  
Transmitting file data .  
Committed revision 2.  
C:\Users\Test\Desktop\work\trunk>
```

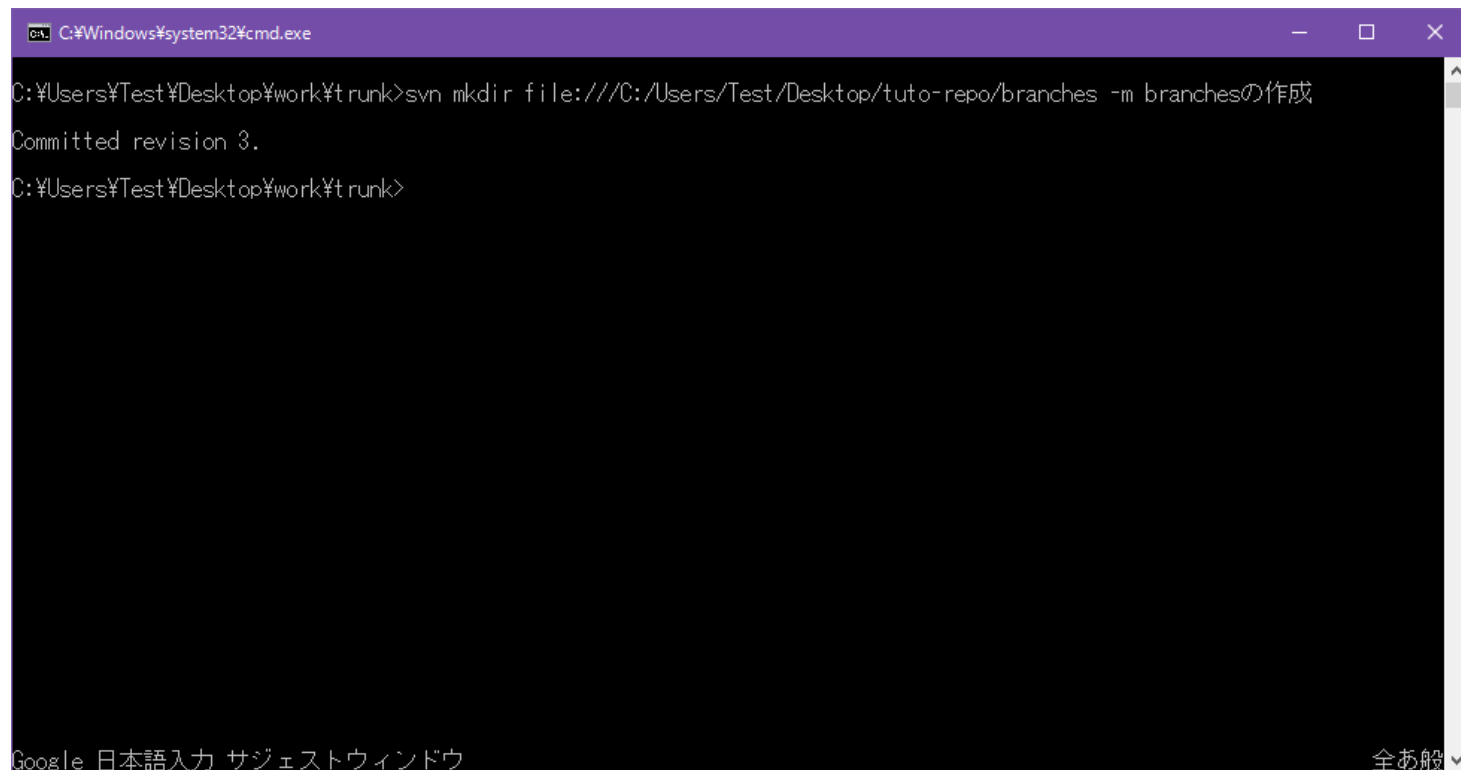
branches (ブランチ) は、トランクから分岐した別のツリーです。ブランチ (枝) という名前は、トランク (幹) から枝分かれすることからきています。トランクとは別に開発を進めたいときに、トランクからブランチを作ります。ブランチを作るということは、トランクディレクトリをブランチとして、「branches」ディレクトリ内にコピーすることです。別々のディレクトリとして存在しますので、もちろん履歴も別々のリビジョンを持つことになります。

branchesを作ってみよう

branchesの作成

はじめに、trunkと同様のコマンドを使って、branchesディレクトリを作ってみましょう。コミットも同時に行います。

```
svn mkdir file:///C:/Users/Test/Desktop/tuto-repo/branches -m branchesの作成
```

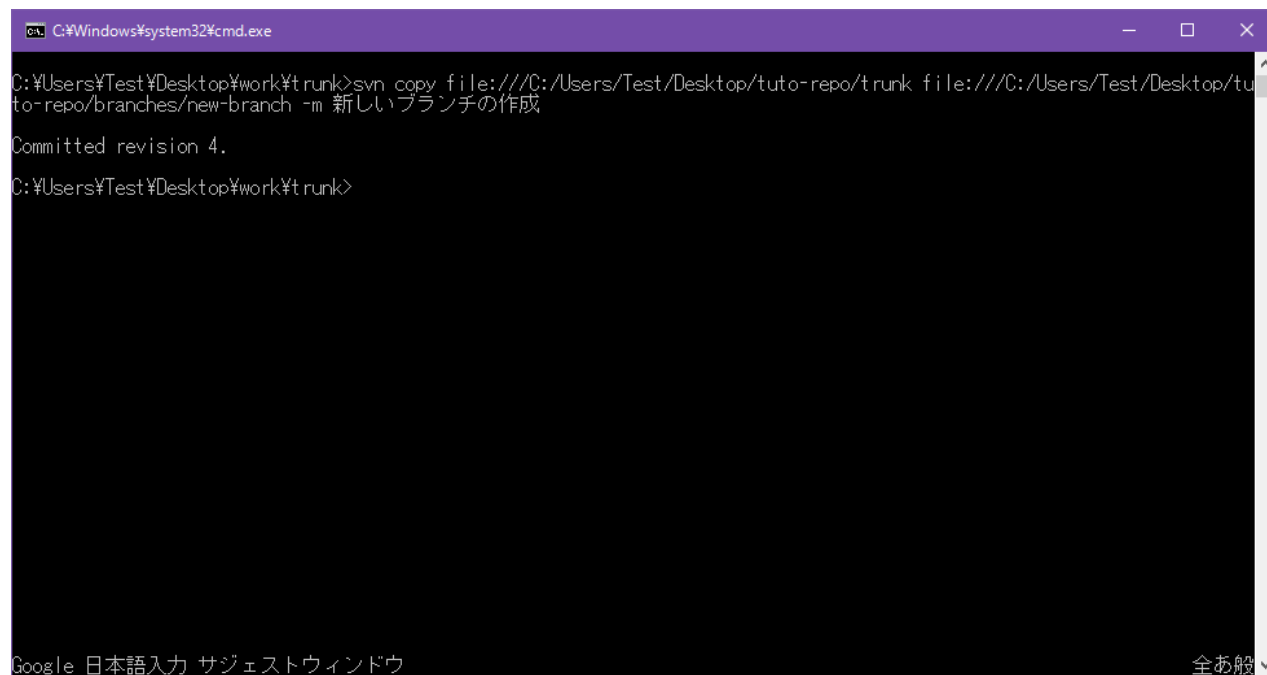


```
C:\Windows\system32\cmd.exe
C:\Users\Test\Desktop\work\trunk>svn mkdir file:///C:/Users/Test/Desktop/tuto-repo/branches -m branchesの作成
Committed revision 3.
C:\Users\Test\Desktop\work\trunk>
```

新しいブランチを作る

それでは、ディレクトリもできたことなので、トランクから新しいブランチを作ってみましょう。ちょっと長いコマンドになるので、コマンドの引数に誤りがないかよく確認してから実行してください。ここでは、ブランチ名を「new-branch」としています。

```
svn copy file:///C:/Users/Test/Desktop/tuto-repo/trunk file:///C:/Users/Test/Desktop/tuto-repo/branches/new-branch -m 新しいブランチの作成
```

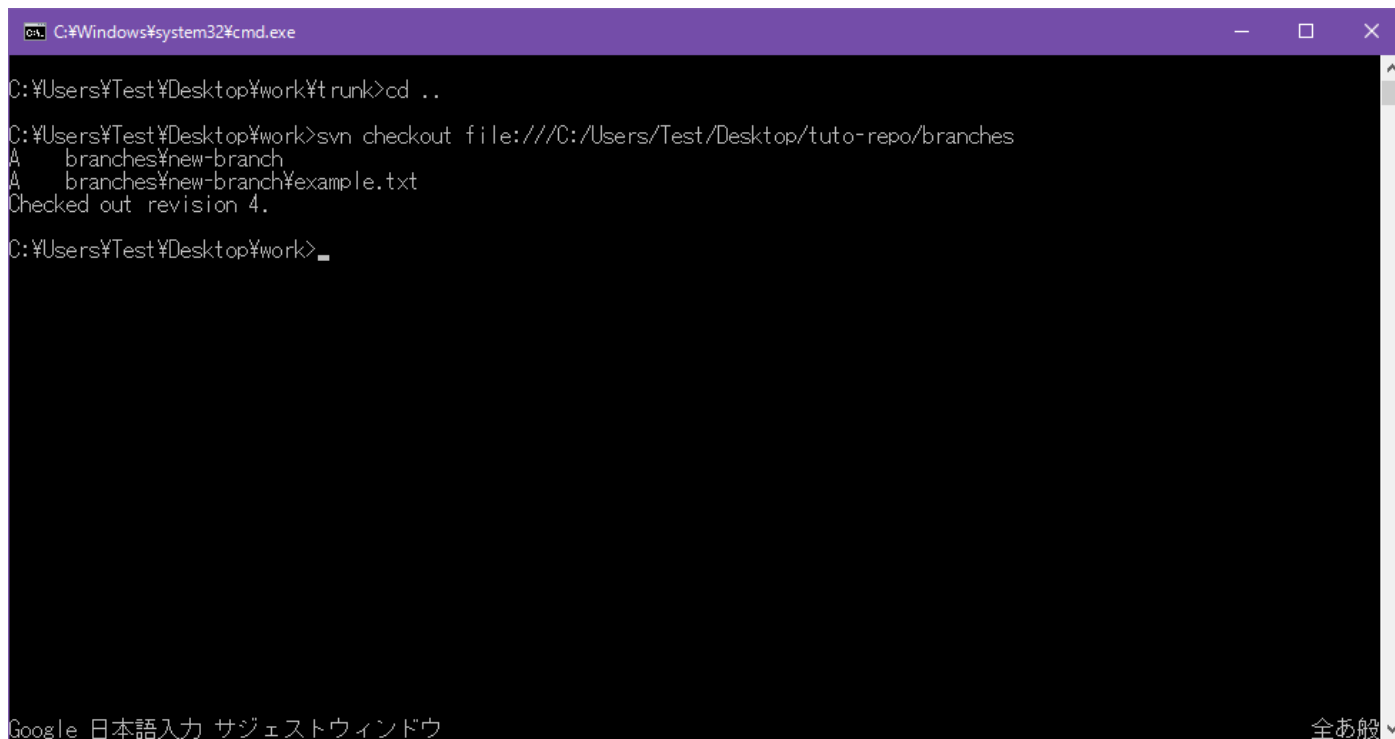


```
C:\Windows\system32\cmd.exe
C:\Users\Test\Desktop\work\trunk>svn copy file:///C:/Users/Test/Desktop/tuto-repo/trunk file:///C:/Users/Test/Desktop/tuto-repo/branches/new-branch -m 新しいブランチの作成
Committed revision 4.
C:\Users\Test\Desktop\work\trunk>
```

branchesを作ってみよう

このコマンドでなにが起こったかというと、現在の「trunk」ディレクトリが、「new-branch」として「branches」ディレクトリにコピーされ、コミットされました。これがブランチを作成するということです。うまく作成できているか、ブランチをチェックアウトして確認しましょう。

```
cd ..  
svn checkout file:///C:/Users/Test/Desktop/tuto-repo/branches
```

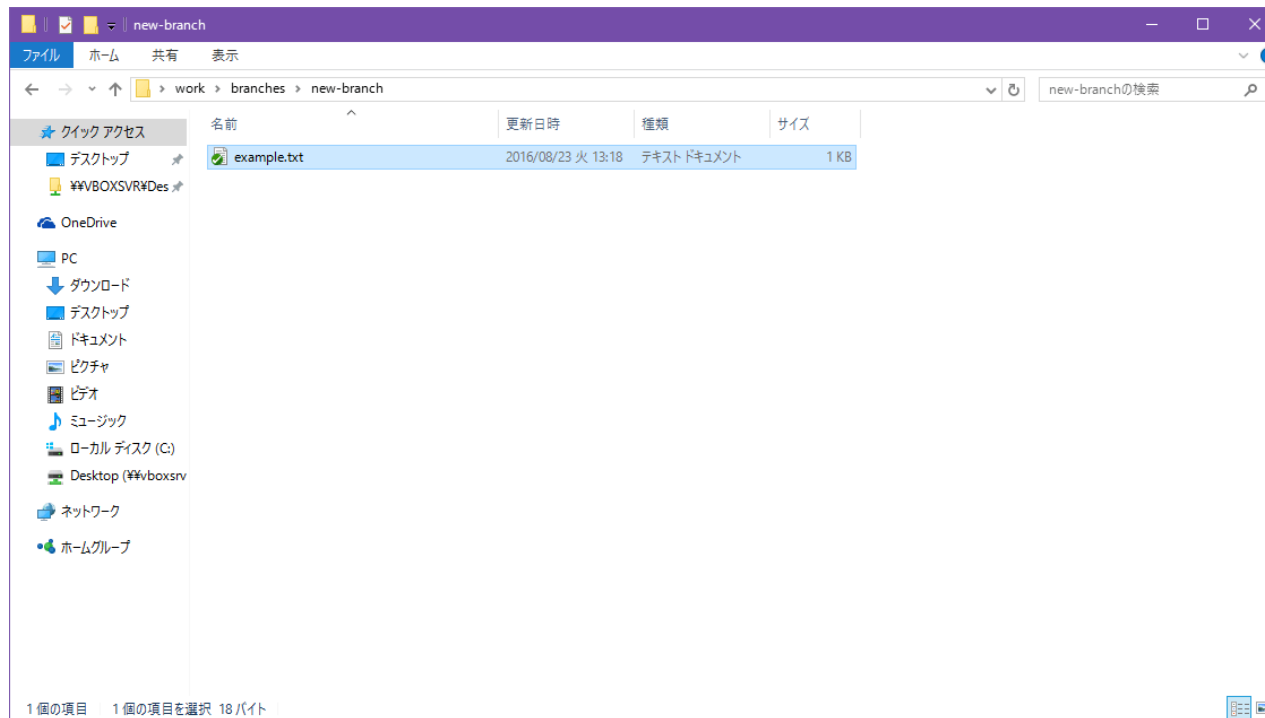


```
C:\Windows\system32\cmd.exe  
C:\Users\Test\Desktop\work>cd ..  
C:\Users\Test\Desktop\work>svn checkout file:///C:/Users/Test/Desktop/tuto-repo/branches  
A      branches/new-branch  
A      branches/new-branch/example.txt  
Checked out revision 4.  
C:\Users\Test\Desktop\work>
```

Google 日本語入力 サジェストウィンドウ 全あ般

branchesを作ってみよう

コマンドを実行したら、「work」ディレクトリを確認してみましょう。「branches」ディレクトリの中に「new-branch」ディレクトリが作られていますね。内容も「trunk」ディレクトリと同じになっています。



後はトランクと同じように、ファイルを追加したり、ファイルに変更を加えたりして、コミットすることができます。もちろん、その変更はブランチに対して行われるため、トランクには影響しません。うまくトランクとブランチを使い分けていきましょう。次に、類似の概念である「tags」について解説していきます。


tags (タグ) は、ある特定時点のトランクに名前をつけたものです。Subversionのリポジトリは履歴を保持しているため、いつでも過去のファイルを自由に取り出すことができます。しかしリビジョン番号では、取り出したい履歴がいくつのリビジョンなのかすぐにはわかりません。そこでタグとして名前をつけておくことで、必要なときにタグを参照してすぐに取り出せるようにします。タグの実態は、tagsディレクトリ内にコピーされたトランクディレクトリです。通常、一度作成したタグを変更することはありません。デフォルトでは、ファイルは読み取り専用になっていないので、誤って変更してしまわないように注意しましょう。

tagsを作ってみよう

tagsの作成

こちらもbranches同様、まずはディレクトリを作成しなければいけません。お馴染みのコマンドを使って、ディレクトリの作成とコミットを行いましょう。

```
svn mkdir file:///C:/Users/Test/Desktop/tuto-repo/tags -m tagsの作成
```

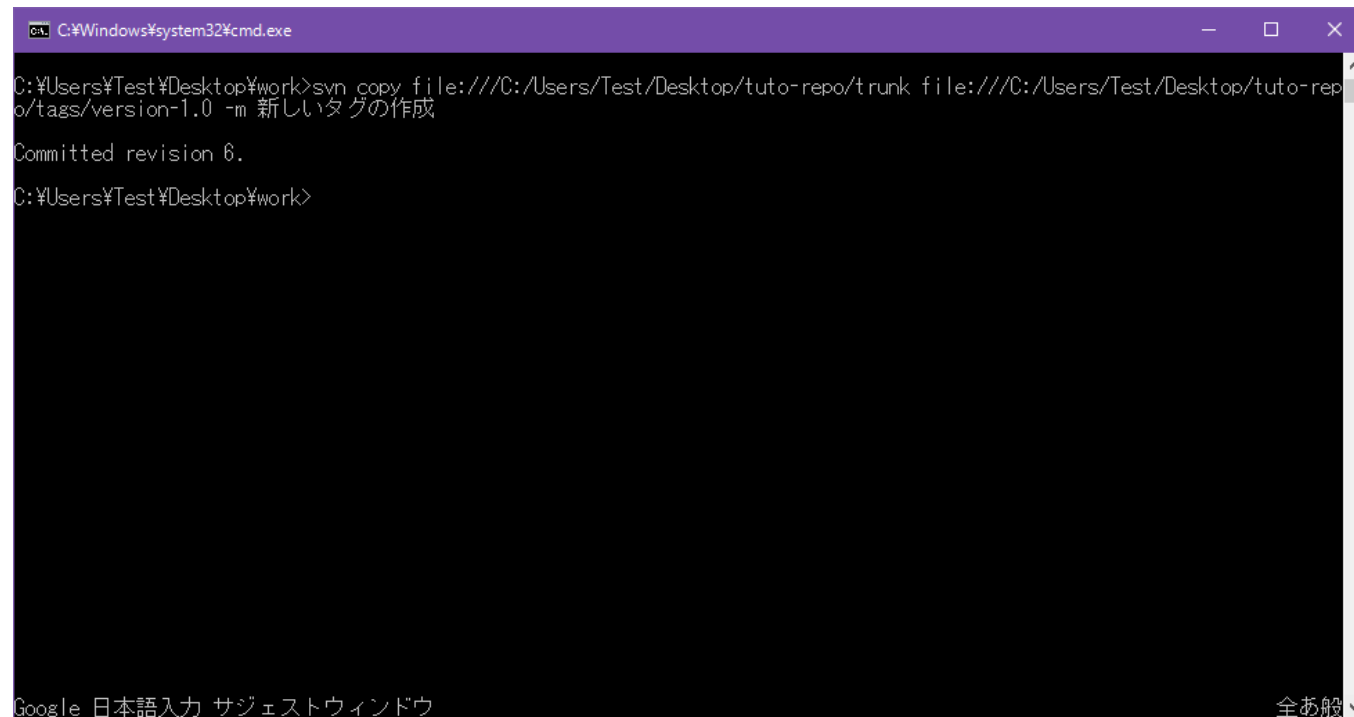


```
C:\Windows\system32\cmd.exe
C:\Users\Test\Desktop\work>svn mkdir file:///C:/Users/Test/Desktop/tuto-repo/tags -m tagsの作成
Committed revision 5.
C:\Users\Test\Desktop\work>
```


新しいタグを作る

それでは、トランクから新しいタグを作ってみましょう。ブランチ同様、先ほどの長いコマンドを使用します。間違えないように注意してください。ここでは、タグ名を「version-1.0」としています。

```
svn copy file:///C:/Users/Test/Desktop/tuto-repo/trunk file:///C:/Users/Test/Desktop/tuto-repo/tags/version-1.0 -m 新しいタグの作成
```

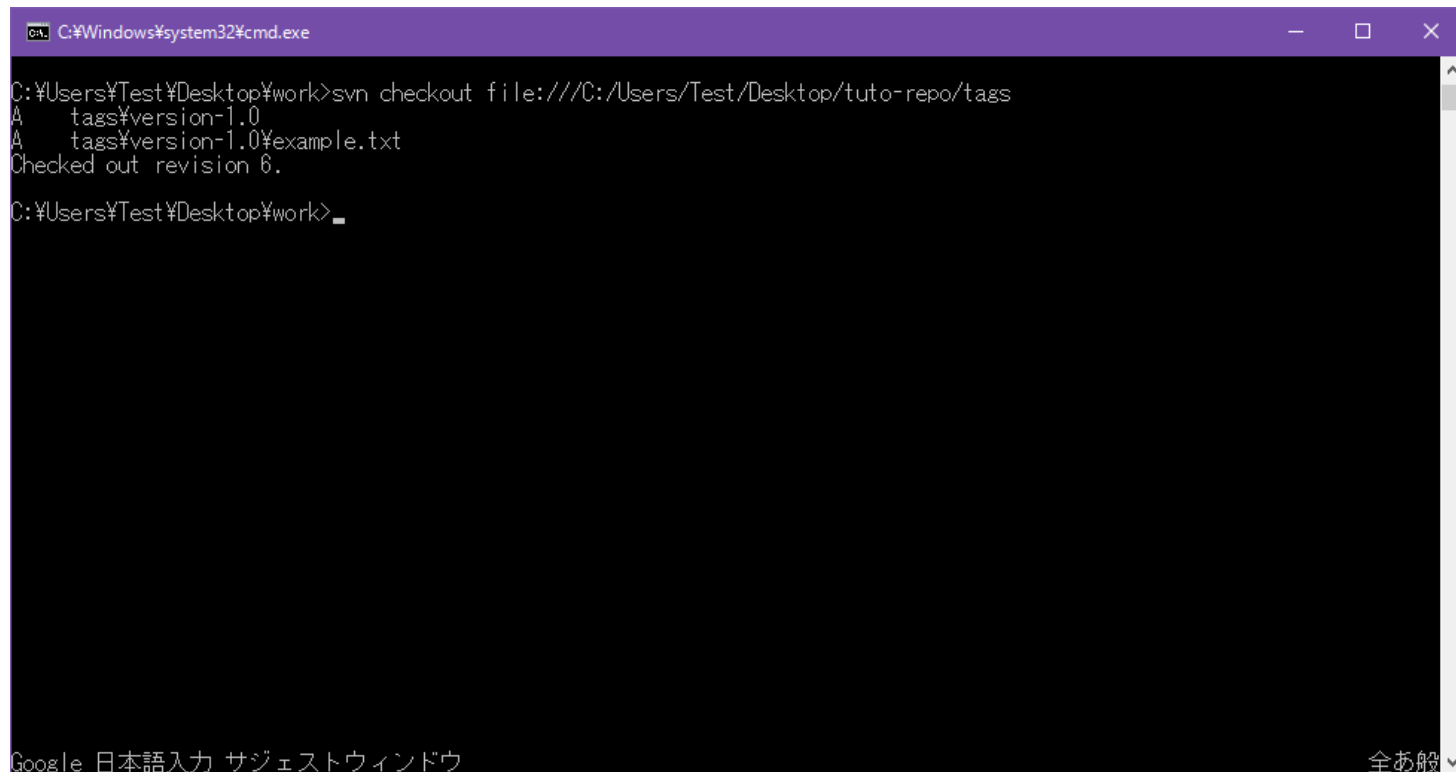


```
C:\Windows\system32\cmd.exe
C:\Users\Test\Desktop\work>svn copy file:///C:/Users/Test/Desktop/tuto-repo/trunk file:///C:/Users/Test/Desktop/tuto-repo/tags/version-1.0 -m 新しいタグの作成
Committed revision 6.
C:\Users\Test\Desktop\work>
```

tagsを作ってみよう

このコマンドでも、ブランチの時と同じことが起こります。現在の「trunk」ディレクトリが「version-1.0」ディレクトリとして、「tags」ディレクトリにコピーされ、コミットされました。うまくいっているか、タグをチェックアウトして確認してみましょう。

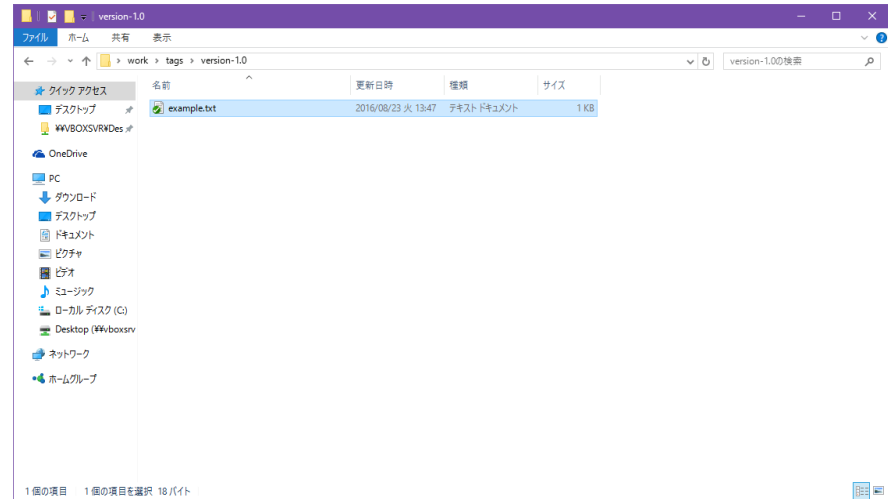
```
svn checkout file:///C:/Users/Test/Desktop/tuto-repo/tags
```



```
C:\Windows\system32\cmd.exe
C:\Users\Test\Desktop\work>svn checkout file:///C:/Users/Test/Desktop/tuto-repo/tags
A      tags\version-1.0
A      tags\version-1.0\example.txt
Checked out revision 6.
C:\Users\Test\Desktop\work>_
```

tagsを作ってみよう

「work」ディレクトリを確認すると、「tags」ディレクトリの中に「version-1.0」ディレクトリが作成され、「trunk」ディレクトリの内容がコピーされていますね。



すでにお気づきかもしれませんが、ブランチを作成する流れとタグを作成する流れは、対象とするディレクトリが違っただけで、それ以外は同じです。なぜかという、ブランチとタグ（およびトランク）はSubversionにとっては概念にすぎず、なんら特別なものではないからです。あくまで慣例として、ブランチとタグが定義されているだけで、内部的には違いはありません。

そのため、ユーザー側がしっかりと役割を理解して使い分ける必要があります。ブランチはトランクとは別の作業をするディレクトリ、タグはある時点のトランクを保管しておくためのディレクトリと覚えておきましょう。間違ってもタグに対してコミットしてはいけません。リポジトリフックというものを使って、タグに対するコミットを防ぐ方法もあります。とはいえ、少人数で使う場合には、全員に周知するだけでも問題ないでしょう。

「trunk」「branches」「tags」は、Subversionの基本となるものです。使い方をしっかりと覚えておきましょう。Subversionでは、リポジトリをどう使うかはユーザーに委ねられています。自由に操作できることは確かですが、慣例となっているベストプラクティスに従うことで、よりよいリポジトリになっていくことでしょう。

- Subversion によるバージョン管理 (1.4)
- Apache Subversion

社内サーバにリモートリポジトリを作るのも一つですが、「開発にまつわる面倒事」をこの際全部、tracpath（トラックパス）に任せてみませんか？

バージョン管理サービス・プロジェクト管理サービスの「tracpath（トラックパス）」では、ユーザー5名、リポジトリ数3つまで、永久無料で利用可能です。

さっそく実務でも使ってみましょう。

自らも開発を行う会社が作ったからこそ、開発チームの「作る情熱」を支える、やるべきことに集中出来るサービスになっています。

エンタープライズ利用が前提のASPサービスなので、セキュリティも強固です。



 **Git / Subversion / Mercurial を即チームに導入！**

エンタープライズ開発でも利用出来るシステム稼働率 99.9% の確かな安定性と、プライバシーマーク取得の安全性で、多くの法人ユーザーにも安心してご利用頂いております。

99.9%
システム稼働率
確かな安定性

プライバシーマーク取得

まずは、1プロジェクト、5ユーザーの永久無料のプランで、お試しください！

無料で作ってみる >