

subversion

[[subversion](#)][[Eclipse](#)]

SVNBOOK

- <http://psyto.s26.xrea.com/misc/svnbook/>

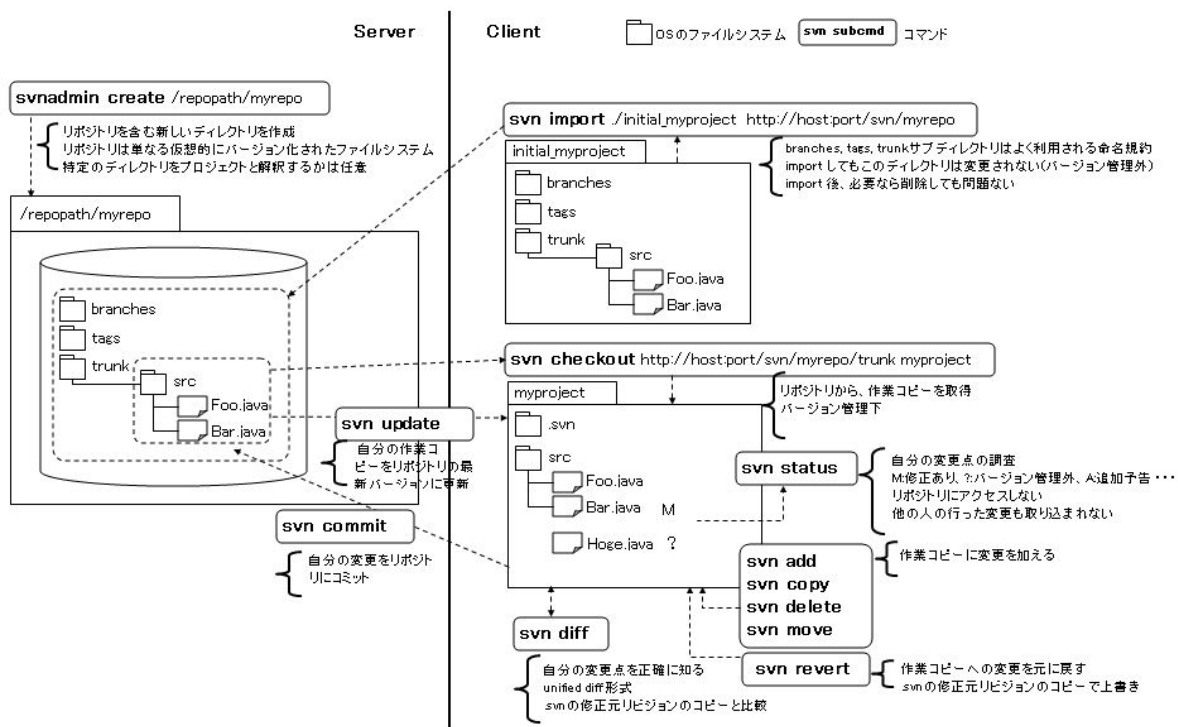
Subversion Download

- <http://subversion.tigris.org/getting.html>
- <http://subversion.apache.org/packages.html>

- [subversion の導入](#)

クイックスタート

基本の概要



導入

- [subversion の導入](#)

リポジトリ

リポジトリとは

- 単なる仮想的にバージョン化されたファイルシステム
- どんなデータも含むことのできる大きなツリー構造

リポジトリ作成

```
# svnadmin create /var/www/svn/myrepo
# chown -R apache:apache myrepo
```

インポート

```
import [<パス>] <URL>
```

```
c:\work>tree /f myproject
フォルダー パスの一覧
ボリューム シリアル番号は 00000002 F2F7:41B7 です
C:\WORK\MYPROJECT
    branches
    tags
    trunk
        Bar.java
        Foo.java
```

```
c:\work>svn import ./myproject http://192.168.24.14/svn/myrepo/ --username pirotto --password north123
```

```
-- この行以下は無視されます --
```

```
A    myproject
```

```
ログメッセージの変更や指定がなされていません
中断 (a), 続行 (c), 編集 (e):
```

```
c
追加しています          myproject%ttrunk
追加しています          myproject%ttrunk\Foo.java
追加しています          myproject%ttrunk\Bar.java
追加しています          myproject%tbranches
追加しています          myproject%tags
```

```
リビジョン 1 をコミットしました。
```

- これで、ツリー構造がデータベースに取り込まれた。
- もとのファイル群は削除しても問題ない。

ツリー構造

- ツリー構造は branches, tags, そして trunk という名前の三つの最上位ディレクトリを含む必要があります。
- trunk ディレクトリはすべてのデータを含んでいるはずですが、branches と tags ディレクトリは空です
- branches, tags, trunk サブディレクトリは実際には Subversion に必要なものではありません。後で利用する時におそらくもっとも便利になるように考えられた、よく利用される命名規約にすぎません。

チェックアウト

- リポジトリのデータを操作するためには、このデータのために、一種の個人用の作業領域となる新しい「作業コピー」を作る必要がある

```
URL[@REV]... [PATH]
```

```
c:\work>svn checkout http://192.168.24.14/svn/myrepo/trunk myproject
A    myproject\Foo.java
A    myproject\Bar.java
リビジョン 1 をチェックアウトしました。
```

ローカルリポジトリの場合、`file:///path`

編集の確認

- Foo.java を編集
- Hoge.java を追加

svn status

```
c:¥work¥myproject>svn status
?      Hoge.java
M      Foo.java
```

svn add

- 作業コピーに追加したファイルを、バージョン管理下へ置くことを予約する (commit で反映される)

```
c:¥work¥myproject>svn add Hoge.java
A      Hoge.java
```

svn diff

- 変更点に対する unified diff 出力を確認

```
c:¥work¥myproject>svn diff
Index: Hoge.java
=====
--- Hoge.java      (リビジョン 0)
+++ Hoge.java      (リビジョン 0)
@@ -0,0 +1,4 @@
+public class Hoge {
+    public static void main(String[] args) {
+    }
+}
¥ No newline at end of file
Index: Foo.java
=====
--- Foo.java      (リビジョン 1)
+++ Foo.java      (作業コピー)
@@ -1,4 +1,5 @@
 public class Foo {
     public static void main(String[] args) {
+        System.out.println("Foo!");
     }
 }
¥ No newline at end of file
```

svn commit

- リポジトリに自分のファイルの新しいバージョンをコミット

```
commit [<パス>...]
```

```
c:¥work¥myproject>svn commit
```

```
-- この行以下は無視されます --
```

```
A      Hoge.java
M      Foo.java
```

```
ログメッセージの変更や指定がなされていません
中断 (a), 続行 (c), 編集 (e):
```

```
C
送信しています          Foo.java
追加しています          Hoge.java
ファイルのデータを送信しています ..
リビジョン 2 をコミットしました。
```

svn update

- ・リポジトリの「最新の」状態を自分の作業コピーに反映します。
- ・リポジトリの変更箇所と作業コピーの変更箇所に競合がある場合

```
c:¥work¥myproject>svn update
'Bar.java' で競合が見つかりました。
選択: 延期 (p), 全差分 (df), 編集 (e),
      衝突自分 (mc), 衝突他人 (tc),
      すべてのオプションを表示 (s): e
public class Bar {
    public static void main(String[] args) {
<<<<<<< .mine
        System.out.println("Bar!");
=====
        System.out.println("Bar2");
>>>>>>> .r3
    }
} 選択: 延期 (p), 全差分 (df), 編集 (e), 解決版 (r),
      衝突自分 (mc), 衝突他人 (tc),
```

Tips

subversion の導入

- ・ [subversion の導入](#)
- ・ [subversion ホスト名が解決できないエラー対応](#)
- ・ [Eclipse で Subversion プラグインのオーバーレイアイコンが消えたときの対応](#)
- ・ [subversion リポジトリの引越し](#)

Client

Windows

- ・ [subversion VisualSVN Windows 用サーバーの導入](#)

Eclipse

- ・ [subclipse Eclipse Subversion Plugin](#)

Visual Studio

- ・ [Visula Studio 用 Subversion プラグイン Ankhsvn](#)