

# Subversion で巡る歴史散歩

by Tasmanian

## 1. Subversion を使おう

---

プログラミングでもなんでも、パソコンで数ヶ月に渡る作業をこなしていると、途中でバックアップと最新版の区別がつかなくなることがよくあります。かといってバックアップをとらないと、ちょっと冒険的な変更を試してみたくなったときなどに困ることになります。

そんなあなたのお悩みを解決してくれるのが、「バージョン管理システム」という類のシロモノです。有名なものに CVS や git などがありますが、ここではその CVS の後継版といえ、ここ数年で世界標準となりつつあるシステム “Subversion” (以下 “svn”) についてご紹介します。

なお、“Subversion” はサブなバージョンではなく、「(政府) 転覆」という意味だそうです。恐ろしい陰謀だまったく。

## 2. インストールと環境整備

---

svn は割とよく知られているプログラムですから、GNU/Linux の著名なディストリビューションをご使用の方は yum や apt-get など容易にダウンロードとインストールが可能かと思いますが、Windows (9x 系には対応せず) でもそう難しい作業ではありません。

svn の公式サイト ( <http://subversion.tigris.org/> ) から [Downloads] を選択、[Windows NT, 2000, XP and 2003] のセクションから [Win32 packages built against Apache 2.0] にジャンプして、svn-1.x.x-setup.exe (ファイル名はバージョンによって変わります) をダウンロード、実行してください。この方法で見つからなければ、Google などで svn setup exe と検索すれば出てくるはずです。

実行してしまえば、あとは普通の製品アプリケーションと何ら変わりはありません。英語ですが。以下の解説は、ひとまず Windows 環境を前提として解説します。

## 3. とりあえずなんかしてみる

---

Subversion を使ってみるといっても、さしあたり管理するモノが思い浮かばないという方が多いかと思います。ここでは、ネット上に公開されているプログラムを svn で入手してみましょう。

ある程度開発が進んだプログラムは、アーカイブ・圧縮されて配布されますが、開発の最先

端に行くバージョンは、バージョン管理システムによってのみ提供される、というケースがあります。svn を使って、時代の先取りをしてしまいましょう。

ここでは、Wikipedia などの中・大規模な Wiki<sup>1</sup>によく利用されるソフト、”MediaWiki”<sup>2</sup> をダウンロードしてみます。コマンドプロンプト (cmd.exe) を開いて、

```
> cd /D C:¥
> svn checkout http://svn.wikimedia.org/svnroot/mediawiki/trunk/phase3
```

とすれば、C:¥phase3¥ に最新版がダウンロードされているはずです。インストーラは英語でしたが、svn コマンドはしっかり日本語化されています。

ここでは詳しくは解説しませんが、MediaWiki は PHP で書かれているので、比較的容易に実行環境を構築することができます。

#### 4. 編集と確認

分かりやすい例として、先ほどダウンロードした MediaWiki の、表示メッセージを変更してみます。language/messages 内の、MessagesJa.php を適当なテキストエディタで開いて、編集してみます。ここでは、「日曜日」をひらがなの「にちようび」に変更してみよう<sup>3</sup>。

保存して、先ほどのコマンドプロンプトに戻ります。

```
> cd ./language/messages
> svn diff MessagesJa.php
```

と実行してみます。Windows では残念ながら文字化けしますが、文字化けを直すと実は

```
Index: MessagesJa.php
=====
--- MessagesJa.php      (リビジョン 26861)
+++ MessagesJa.php      (作業コピー)
@@ -94,7 +94,7 @@
     'skinpreview' => '(プレビュー)',

# Dates
```

<sup>1</sup> 不特定多数のユーザーによって編集することができる Web サイトのシステム。

<sup>2</sup> 公式サイト: <http://www.mediawiki.org/>

<sup>3</sup> たまごやきはまあるくふんわり

## Subversion で巡る歴史散歩

```
- 'sunday'      => '日曜日',
+ 'sunday'      => 'にちようび',
' monday'      => '月曜日',
' tuesday'     => '火曜日',
' wednesday'   => '水曜日',
```

のように先ほど変更した箇所(差分)がプラスとマイナスの記号で表されています。あなたが MediaWiki の開発者であれば、この変更を世に知らしめるための操作ができるのですが、ここでは自分の PC の中の変更留着めておきます。次節では、自分が管理者なファイルを管理してみます。

## 5. リポジトリをつくる

---

では実際に、自作のファイルを `svn` で管理してみましょう。 `svn` では、「リポジトリ」と呼ばれるファイルの集まりを管理の対象としますので、まずこの「リポジトリ」を作成します。

```
svnadmin create C:¥svnrepo
```

として、 `C:¥svnrepo` に `svn` の管理用ファイルを作ります。この中のファイルは、原則として `svn` からのみ操作します。

実際のファイル編集をするディレクトリ<sup>4</sup> を、ここでは `C:¥svnwork` とします。リポジトリの中に `trunk` というディレクトリを作って、それを管理することにしましょう。

```
> cd /D C:¥
> svn mkdir file://localhost/C:/svnrepo/trunk
> svn checkout file://localhost/C:/svnrepo/trunk svnwork
リビジョン 0 をチェックアウトしました。
```

この操作は、先ほど MediaWiki をダウンロードしたときと同じですね。ここで `C:¥svnwork` にできたファイル群(今回は空です)のことを「作業コピー」といい、ファイルの編集はここで行います。作業コピーに対して行った変更は「コミット」するまで `svn` の管理下に入りません。

`svnwork` 内に “`main.txt`” というファイルを作ってみました。ここで、作業コピーの状態を見るコマンド (`svn status`) を実行してみます。

```
> cd /D C:¥svnwork
> svn status
?      main.txt
```

---

<sup>4</sup> 俗に言うフォルダのこと。

はてなマークが表示されてしまいました。このファイルは、svn にとっては捕捉外の、いわばヤミファイルです。こいつを svn に認識させてやらねばなりません。

```
> svn add main.txt
> svn status
A      main.txt
```

これで、main.txt の状態が “A”dd (新規追加ファイル)になりました。では、先ほどの「コミット」を試みましょう。

```
> set SVN_EDITOR=notepad
> svn commit
```

一行目では、コミット時に必要となるテキストエディタを指定しています。メモ帳 (notepad.exe) で十分でしょう。二行目で、早速そのメモ帳が起動して、変更内容の概要を求めてきます。適当に入力して、保存ののち終了しましょう。コマンドプロンプトのほうに戻ると、

```
追加しています          main.txt
ファイルのデータを送信しています .
リビジョン 1 をコミットしました。
```

などと表示されているはずですが、この状態で “svn status” を実行しても、リポジトリと比較して特に変更はないため、なにも出力されないと思います。そこで、先ほどの main.txt を編集してみましょう。その後、“svn diff” で変更点を確認します。

```
> svn di
Index: main.txt
=====
--- main.txt      (リビジョン 1)
+++ main.txt      (作業コピー)
@@ -3,7 +3,7 @@
 3      恐怖大作戦
 4      時の中のアイドル
 5      小さな思い出
-6      初めてのデート
+6      総集編
 7      ねらわれた学園祭
 8      トラブル
 9      風…流れる
```

## Subversion で巡る歴史散歩

ここら辺の作業も MediaWiki のときと全く一緒ですね。この状態で “svn status” すると、main.txt の状態は “M” (変更あり) になっているかと思います。確認したら、これも「コミット」しておきましょう。

## 6. 過去を振り返る

---

折角 svn の管理下に置いたのですから、変更点などが後々確認できるようにならなければ仕方ありません。ここでは、過去のバージョンを確認・活用する便利なコマンドをいくつか紹介します。

```
svn log main.txt
```

これで、main.txt がどのように変更されてきたのか、という履歴がわかります。過去のバージョンを参照するには、“svn cat” を使います。リダイレクトしてファイルに保存してみます。

```
svn cat -r 1 main.txt >main_old.txt
```

-r スイッチで、どのバージョンを取り出すか指定しています。これで、main\_old.txt に先ほど main.txt を作成したときの内容が保存されました。

また、バージョン間の差分を確認することもできます。

```
svn diff -r 1:2 main.txt
```

作業コピーとの差分をとるには、-r 1 とだけすれば OK です。

## 7. ファイルを退避させる

---

冒頭で提起した問題を解決するためのものですが、svn にはファイルを退避させ、いわば「パラレル・ワールド」を作る機能があります。リポジトリの中に branch/danger というディレクトリを作って、今後の作業はそこで行うことにしましょう。こうすることで、先ほどまで操作していた/trunk の中身には影響を与えずに済みます。

```
svn mkdir file://localhost/C:/svnrepo/branch
```

```
svn copy file://localhost/C:/svnrepo/trunk file://localhost/C:/svnrepo/  
branch/danger
```

こうすることで、現在の trunk と全く同じ内容がコピーされました。このままでは作業コピーは trunk のものですから、これを変更します。

```
svn switch file://localhost/C:/svnrepo/branch/danger .
```

これで完了です。あとは先ほどと同じように編集できます。

## 8. そして。

---

ここまで基本的な操作をチュートリアル的に解説してきましたが、svn の機能はこれだけに留まらず、他にも様々なことが可能です。リファレンスが <http://subversion.bluegate.org/> にありますので、これを参考にとしてみるとよさげです。

また、Web サーバーと連携してインターネット上にリポジトリを公開する機能など、外部アプリケーションと連携して他人と共同開発をすることもできます。

なにやら Windows Vista にはバージョン管理機能がついているらしいですが、本格的なアプリケーション開発を行うのであれば、やはり Subversion などを導入するのが最善でしょう。たぶん。