

Subversion: Referencia Rápida

Sistema de control de versiones

Un sistema de control de versiones es un software que administra el acceso a un conjunto de ficheros, y mantiene un historial de cambios realizados. El control de versiones es útil para guardar cual quier documento que cambie con frecuencia, como una novela, o el código fuente de un programa.

¿Qué es Subversion?

- Subversion es un software destinado a facilitar el trabajo en equipo sobre un conjunto de ficheros.
- Cada usuario puede participar desde un ordenador conectado a Internet o una red local.
- Los usuarios se conectan al servidor para intercambiar modificaciones.
- El servidor recuerda todos los cambios realizados sobre un fichero y permite recuperar versiones anteriores.
- El acceso al servidor esta protegido mediante contraseñas.
- Es posible realizar copias de seguridad de toda la información.

¿Cómo funciona?

- Subversion se compone de un programa *servidor* y otro *cliente*.
- El servidor contiene una copia maestra de la información a compartir.
- Los usuarios usan el cliente para descargar la información existente en el servidor.
- Cuando un usuario realiza un cambio, lo envía al servidor para que otros usuarios puedan descargarlo.
- El servidor guarda los ficheros dentro de una base de datos (no son visibles en el sistema de ficheros).

Lista de comandos

Bajar un proyecto: checkout

```
# Nos situamos en el directorio donde queramos
# descargar el proyecto
cd directorio
# Lo descargamos
svn checkout https://servidoresvn/ruta/proyecto .
```

Actualizar la copia local: update

```
# Actualizamos
svn update
# Actualizamos sólo el directorio actual
svn -N update
# Actualizamos sólo un fichero
svn update <fichero>
# Actualizamos en la revisión que indicamos con n
# donde n es el número (1,2,3,...) de la revisión
svn update -r n
```

Resolver conflictos

Comprobar el estado de la copia local: status

```
svn status
L   ./abc.c           # Fichero bloqueado
M   ./bar.c           # Modificado localmente
?   ./foo.o           # Fichero ignorado
!   ./foo.c           # Fichero borrado por el
                          # usuario u otro programa
D   ./stuff/fish.c    # Preparado para borrado
A   ./stuff/loot/blou.h # Preparado para añadir
C   ./stuff/loot/lump.c # Conflictos con un update
G   ./foo.c           # Modificado localmente e
                          # incorporados cambios
                          # remotos
```

Subir cambios: commit

```
# Sube los cambios y abre un editor en el que se escriben
# los cambios efectuados en la revisión
svn commit
# Cita los cambios en la misma orden
svn commit -m "Bla bla bla..."
# Envía los cambios de un fichero y de un directorio
svn commit <fichero> <directorio>
# Envía los cambios de todos los ficheros .c
svn commit *.c
```

Mover un fichero o directorio: move

```
# Movemos el fichero origen al fichero destino
svn move <fichero origen> <fichero destino>
```

Eliminar un fichero o un directorio: rm, remove, del, delete

```
# Eliminamos el fichero
svn rm <fichero>
```

Añadir un fichero o un directorio: add

```
# Añadimos un directorio (sin incluir subdirectorios)
svn add <directorio>
# Añadimos un directorio (incluyendo subdirectorios)
svn add -R <directorio>
# Añadimos un fichero
svn add <fichero>
```

Deshacer cambios: revert

```
# Deshacer los cambios locales (recursivamente con -R)
svn revert
```

Ver logs: log

```
# Para ver el log de todas las revisiones
svn log
# Ídem con información adicional
svn log --verbose
# Log de revisión n
svn log -r <n>
# Log de un fichero
svn log <fichero>
```

Crear un Changelog

```
# Creo un fichero de nombre Changelog con todos los mensajes
# desde la revisión 1 (-r1) hasta la última revisión (HEAD)
svn log -r1:HEAD >> ChangeLog
# Hago add porque es un fichero nuevo
svn add ChangeLog
# subo el fichero para añadirlo a la distribución
svn commit ChangeLog
```

Crear una release en tar.gz: export

```
svn export https://servidoresvn/ruta/proyecto
# Esto creará un directorio llamado proyecto
# con nuestro proyecto exportado
tar czvf proyecto.tgz proyecto/
```

Obtener ayuda: help

```
# Ayuda general
svn help
# Ayuda con algún comando
svn help <comando>
```

Un día de trabajo con Subversion

Este es un ejemplo de trabajo con Subversion usando la línea de comando:

```
# Descargo (checkout) el proyecto del servidor usando
# la URL que me dió el administrador del repositorio
svn checkout https://servidor/ruta/proyecto
# Ahora tengo una copia del proyecto en mi ordenador y puedo
# trabajar en ella
# Si quiero modificar ficheros no tengo más que editarlos
# Si decido añadir ficheros o directorios hago esto:
svn add fichero1 fichero2 directorio3
# Si quisiera eliminar, haría esto otro:
svn delete ficheroA directorioB
# En cualquier momento puedo revisar los cambios que he hecho
# en mi copia local
svn status
# Al terminar, actualizo mi copia local con los cambios
# subidos por otros usuarios
svn update
# Si hay conflictos los resuelvo
%TODO
# Envío mis cambios al servidor informando de lo que he hecho
svn commit m "he modificado tal y cual, añadido tal"
```

2005 Fernando Blat, Jose Luis Hidalgo
(fblat,jhidalgo}@dsic.upv.es)
\$Revision: 0.1 \$, \$Date: 08/01/2005\$.
Basado en el tutorial de Alejandro Ramírez (jano@1x4x9.info), que se puede encontrar en <http://www.1x4x9.info>