

Subversion FAQ

Leandro Lucarella <llucare@fi.uba.ar>

\$Id: faq-subversion.lyx 37 2003-08-06 02:37:18Z llucare \$

Este documento pretende explicar brevemente el uso *básico* del subversion <http://subversion.tigris.org/> (o svn). Como nació en el LUGFI <http://www.fi.uba.ar/lug/>, la mayoría de los ejemplos y explicaciones están orientadas al uso de su repositorio. De todas maneras creo que el documento es bastante general como para servirle a otra gente también.

También aclaro que esto no pretende ser una traducción del Manual de Subversion <http://svnbook.red-bean.com/>. Cualquiera de los temas aquí tratados pueden ampliados en dicho manual, que recomiendo leer (al menos por arriba) muy enfáticamente.

Índice

1. Uso básico.	1
1.1. Ya tengo una cuenta de svn, ¿y ahora?	1
1.1.1. ¿Y si uso Debian <i>stable</i> ? En la página web del subversion solo está el código fuente y binarios RPM ¿qué bajo?	1
1.2. Ya tengo mi cuenta de svn y mi svn andando ¿y ahora?	1
1.3. ¿Cómo modifico un archivo?	2
1.4. ¿Qué es un <i>commit</i> ?	2
1.5. ¿Qué es un <i>update</i> ?	2
1.6. ¿Cómo sé que cosas cambiaron/van a cambiar antes de hacer un <i>commit/update</i> ?	2
1.7. Me dice algo como que el repositorio está lockeado, ¿Qué pasa?	2
1.8. ¿Cómo agrego/nuevo/borro un archivo/directorio?	3
1.9. ¡Ups! Rompí algo, ¿cómo hago para volver atrás?	3
1.10. ¿Qué son esas cosas raras <i>\$id</i> , <i>\$date</i> , etc?	3
1.11. ¿Por qué a mí no me expande los <i>keywords</i> ?	3
1.12. ¿Hay una forma más fácil de poner los <i>keywords</i> ?	4

1. Uso básico.

1.1. Ya tengo una cuenta de svn, ¿y ahora?

Ahora asumo que ya tenés el svn andando también. Si no es el caso y usas Debian (*unstable*), hacé un:

```
# apt-get install subversion
```

Listo, ya tenés el subversion. Si usas otra distro, buscá si está disponible un paquete (seguramente sí) y si no vas a tener que compilarlo vos. En la página del subversion podés encontrar paquetes precompilados <http://subversion.tigris.org/servlets/ProjectDocumentList> para las distros más populares.

1.1.1. ¿Y si uso Debian stable? En la página web del subversion solo está el código fuente y binarios RPM ¿qué bajo?

Yo diría que pruebes con el código fuente. O probá de apuntar tu `/etc/apt/sources.list` al `unstable` y fijate si no te tiene que actualizar media distribución para bajar el paquete. Es un paquete que (creo que) no tiene muchas dependencias, así que con un poco de suerte...

1.2. Ya tengo mi cuenta de svn y mi svn andando ¿y ahora?

Ahora podés hacer un *checkout* del repositorio, solamente escribí esto:

```
$ svn co http://svn.llucax.hn.org/svn/lugfi/tronco lugfi-www
```

(Reemplaza el `http://svn.llucax.hn.org/svn/lugfi/tronco` por el repositorio que quieras bajar y el `lugfi` del final por el directorio en donde querés que baje todo)

Esto te va a crear un subdirectorio `lugfi-www` después de preguntarte tu usuario y password. En ese subdirectorio está todos los archivos de sitio.

1.3. ¿Cómo modifico un archivo?

Entrá al subdirectorio `lugfi-www` y editalo con tu editor favorito! Hacé de cuenta que estás trabajando en un subdirectorio cualquiera. Modificá todos los archivos que quieras, el repositorio no se va a modificar hasta que hagas un *commit*.

1.4. ¿Qué es un commit?

Con el comando `commit` del `svn` se *suben* los cambios al repositorio. Es recomendado hacer un `update` antes:

```
.../lugfi-www/$ svn up
[...]
At revision 30.
.../lugfi-www/$ svn ci [archivos]
A archivo
M archivo
[...]
```

El `svn ci` (commit) te va a abrir tu editor favorito (o se va a quejar si no puede hacerlo) para que escribas un mensaje comentando qué es lo que cambiaste. ¡Por favor, tratá de ser lo más descriptivo posible!

1.5. ¿Qué es un update?

El comando `update` se encarga de poner al día nuestra copia local del repositorio. Si alguien hizo algún cambio después de que lo hayamos bajado, este comando nos lo actualiza de una forma *barata*, ya que sólo baja los cambios (no todo el repositorio de nuevo).

```
$ svn up [<archivo>]
```

`<archivo>` es opcional (como en prácticamente todos los comandos de `svn`), si no se especifica, se toma el directorio actual.

1.6. ¿Cómo sé que cosas cambiaron/van a cambiar antes de hacer un commit/update?

Para tener un pantallazo, el comando `status` es tu mejor amigo:

```
$ svn st
```

Si querés ver qué cosa cambió en cada archivo, el comando `diff` es lo que necesitas:

```
$ svn diff <archivo>
```

1.7. Me dice algo como que el repositorio está lockeado, ¿Qué pasa?

Es probable que hayas cortado un comando del `svn` con `Ctrl-C` o algo así. Probá de hacer un:

```
$ svn cleanup
```

y si eso no lo soluciona preguntá en `lugfi-www`.

1.8. ¿Cómo agrego/muevo/borro un archivo/directorio?

Para agregar archivos o directorios (de forma recursiva por defecto) está el comando `add`:

```
$ svn add miarchivo midirectorio
A miarchivo
A midirectorio
```

Para mover (renombrar) un archivo o directorio es casi igual que lo que hacés frecuentemente, sólo que hay que agregar antes el `svn`:

```
$ svn mv miarchivo milindoarchivo
D miarchivo
A+ milindoarchivo
$ svn rm otroarchivo
D otroarchivo
```

Todos estos cambios (y cualquier cambio que hagas) no se subirá al repositorio hasta que no hagas un `commit`.

1.9. ¡Ups! Rompí algo, ¿cómo hago para volver atrás?

Si todavía no hiciste un `commit`, es fácil, simplemente escribí:

```
$ svn revert [<archivo>]
```

Si ya hiciste el `commit`, tendrías que hacer un `merge` con la versión anterior pero eso escapa a los objetivos de este documento.

1.10. ¿Qué son esas cosas raras \$id\$, \$date\$, etc?

Son *keywords*, es decir, palabras clave que el `svn` puede traducir a propiedades del documento. En realidad van con mayúscula la primera letra pero lo puse a propósito mal para que en el título no lo expanda. A continuación las escribo bien así son expandidas y ven el significado de cada una:

```
$Id: faq-subversion.lyx 37 2003-08-06 02:37:18Z llucare $
$Date: 2003-08-05 23:37:18 -0300 (Tue, 05 Aug 2003) $
$Rev: 37 $
$URL: http://svn.llucax.hn.org:81/svn/lugfi/tronco/proyectos/faq_website.lyx $
$Author: llucare $
```

1.11. ¿Por qué a mí no me expande los *keywords*?

El `svn` por defecto no expande nada para evitar problemas indeseables, tengan en cuenta que en especial al trabajar con PHP (o Perl) se usa mucho el signo `$` y si usamos una variable `$Id` y luego otra es probable que lo expanda. Es por esto que recomiendo altamente **no** utilizar nombres de variable en PHP que empiecen con Mayúsculas (para evitar problemas con futuros *keywords* también). En fin, en caso de querer que expanda los *keywords* hay que decirlo explícitamente, seteando la propiedad especial (los archivos además del contenido pueden tener propiedades arbitrarias) llamada `svn:keywords`:

```
$ svn ps svn:keywords "Id Date Author" <archivo>
```

Donde "Id Date Author" puede ser en realidad cualquier combinación de *keywords* que queramos expandir. También podemos hacer un:

```
$ svn pe svn:keywords <archivo>
```

Para que abra nuestro editor favorito y escribir en él los *keywords* a expandir.

Es importante que no expandan keywords en archivos binarios, porque pueden tener la mala suerte de que les parta el archivo al medio al expandir algo no deseado.

1.12. ¿Hay una forma más fácil de poner los *keywords*?

No que conozca, pero para simplificar un poco las cosas (y considerando que siempre expando los mismos *keywords*) me hice dos *micro-scripts*:

```
svnk:  
#!/bin/sh  
svn ps svn:keywords "Id Date Author Rev URL" "$@"
```

que setea los *keywords* y

```
svnaddkw:  
#!/bin/sh  
svn add "$@"  
svn ps svn:keywords "Id Date Author Rev URL" "$@"
```

que agrega los archivos y setea los *keywords*.