
Micro-cómo empezar a trabajar con CVS

Ismael Olea

\$Id: micro-como-empezar-con-cvs.xml,v 1.3 2003/12/04 14:08:24 olea Exp \$

Tabla de contenidos

Introducción	1
terminología cvs	1
Invocar a cvs	2
Configuración	2
La autenticación	3
Uso	4
Modo de uso	4
Bajar por primera vez el módulo	4
Actualizar cambios	5
Publicar nuestras modificaciones	6
Resolución de conflictos	6
Añadir ficheros al módulo	7
Eliminar ficheros del módulo cvs	7
Cómo configurar un servidor y cómo subir nuevos módulos al repositorio.	7
Bibliografía	8
interfaces gráficos	8
Otros recursos cvs interesantes	8
Otros recursos cvs más avanzados	8
Licencia	9

Introducción

CVS [<http://www.cvshome.org/>] es algo que al principio es difícil de explicar con sencillez pero que al final resulta indispensable para la vida. Así, como lo oye.

CVS es un sistema de mantenimiento de código fuente extraordinariamente útil para grupos de desarrolladores que trabajan cooperativamente usando alguna clase de red.

para ser más concreto, CVS permite a un grupo de desarrolladores trabajar y modificar concurrentemente ficheros organizados en proyectos. Esto significa que dos o más personas pueden modificar un mismo fichero sin que se pierdan los trabajos de ninguna. Además, sus funciones más usadas son muy sencillas de usar.

Además CVS guarda las versiones antiguas de los ficheros. Esto permite recuperar en cualquier momento versiones anteriores a la actual.

Dado que trabaja con ficheros ASCII es igual de útil para trabajar con código fuente de programas o con toda clase de documentos siempre que su formato sea completamente de texto, como pueden ser ficheros sgml/html/xml.

Este documento es una referencia inmediata para trabajar con cvs. A poco que le saques partido necesitarás consultar otro más extenso. Sin embargo el 80% o más de las acciones que desarrollan los usuarios de cvs están documentadas en este artículo.

Con CVS puede trabajarse de forma local (repositorio y copias de trabajo en el mismo sistema) o remota (el repositorio está en un sistema servidor y la copia local en otro que es cliente del primero).

En este artículo sólo prestaré atención al modo de trabajo remoto, que es el que más impresiona a las visitas.

terminología cvs

para que no te pierdas, un brevísimo vocabulario:

repositorio

jerarquía de directorios alojada en el servidor que contiene diferentes módulos a disposición de los usuarios.

módulo

árbol de directorios que forma parte del repositorio; cuenta con un nombre identificador gracias al cual podremos bajárnoslo de forma selectiva.

Invocar a cvs

cvs es un programa que se invoca desde intérpretes de órdenes. según cómo sea su configuración (y estoy pensando en las diferentes formas de autenticación) lo podrás usar en procesos por lotes sin ningún problema.

Un aspecto que debes tener en cuenta (sobre todo si este es el primer documento que lees sobre cvs) es que cvs tiene parámetros para cada una de sus órdenes. para conocerlas tienes dos métodos: invocar cvs como «cvs help» o mirar la ayuda.

Configuración

Puedes usar varios ficheros de configuración que cvs reconocerá y usará.

Yo, que soy el mejor ejemplo que tengo a mano, uso los siguientes:

~/cvsignore

que contiene los sufijos de los ficheros que no nos interesa que cvs controle:

```
*.tex *.aux *.dvi *.ps *.log
```

~/cvsrsc

que contiene parámetros que cvs usará cada vez que se invoque:

```
cvs -z 3  
update -Pd  
diff -uw
```

La autenticación

Al trabajar en remoto con cvs pueden elegirse varias alternativas de autenticación (es decir, de demostrar al servidor que somos quienes decimos que somos).

Las que más he visto usar son vía pserver y vía ssh.

Deberás elegir alguna de estas técnicas en función de sus necesidades, las ganas que tengas de complicarte la vida y, sobre todo, del grado de neurosis obsesiva por la seguridad que padezcas.

ssh

Como uno va por la vida predicando el software libre no puedo más que recomendar el uso de OpenSSH que puede encontrarse en <http://www.openssh.org>. Si no me equivoco, esta herramienta es compatible con la norma ssh 1.x.

para que cvs use este modo de autenticarse se pueden usar estas variables de entorno:

```
export CVSROOT=":ext:USUARIO@www.vivalavirgen.com:/home/cvs"  
export CVS_RSH=/usr/bin/ssh
```

donde

USUARIO

es nuestro nombre de usuario;

www.vivalavirgen.com

es el servidor que aloja al repositorio;

/home/cvs

es el directorio del servidor en el que está el repositorio;

/usr/bin/ssh

es la ruta completa al amigo ssh.

Tenga en cuenta que, usando esta técnica, tendrá que autenticarse (es decir, suministrar su contraseña) cada vez que ejecute cvs.

pserver

Esta técnica es más de «andar por casa». Se monta rápidamente y no necesita de programas añadidos.

```
export CVSROOT=":pserver:USUARIO@www.vivalavirgen.com:/home/cvs"
```

donde

USUARIO

es nuestro nombre de usuario;

`www.vivalavirgen.com`

es el servidor que aloja al repositorio;

`/home/cvs`

es el directorio del servidor en el que está el repositorio.

Si usa esta técnica, antes de poder trabajar con cvs debe autenticarse con el servidor. Eso se hace con la orden login:

```
$ cvs login
```

CVS le pedirá la contraseña del usuario que haya configurado.

Si la contraseña es correcta cvs guardará la información que necesita en el fichero

```
~/ .cvspass
```

y no tendrá que volver a autenticarse.

Si por algún motivo exotérico quisiera «cerrar la sesión cvs» bastará con hacer:

```
$ cvs logout
```

Uso

Modo de uso

A continuación se propone una sencilla metodología de trabajo con cvs para evitar trabajos redundantes. Piénsese por ejemplo en la eliminación de erratas o errores en documentos o en código fuente.

Antes de cada sesión de trabajo es conveniente hacer «cvs update» para asegurarnos de que disponemos de las últimas modificaciones.

Justo al acabar cada sesión de trabajo es conveniente hacer «cvs ci» para que todas nuestras modificaciones se propaguen en el servidor.

Bajar por primera vez el módulo

para crear una copia de trabajo local del módulo cvs deseado debemos usar la orden co (o su equivalente checkout):

```
$ cd [padre-de-directorio-donde-se-alojará-el-módulo]  
$ cvs co [nombre-del-módulo]
```

Esto creará una jerarquía de directorios donde se almacenará el módulo.

Este paso sólo hay que hacerlo una vez por cada módulo.

A partir de este momento no es necesario configurar las variables de entorno porque cvs sabe a qué repositorio pertenece el módulo con sólo examinar los subdirectorios CVS. No debes modificar nunca esos subdirectorios o volverás loco al pobre cvs.

Actualizar cambios

Cuando queramos actualizar la copia local del módulo con los cambios que hayan podido hacer otros usuarios y que ya han sido enviados al repositorio deberemos hacer:

```
$ cd [directorio-del-módulo]
$ cvs update
```

Observaréis que cvs informa acerca de qué está haciendo con cada fichero y para eso utiliza un código de un letra. Este es el significado:

U

Se ha bajado un fichero completamente nuevo.

P

Se ha bajado una modificación del fichero y ahora éste está actualizado.

A

El fichero lo ha añadido usted pero no lo ha subido todavía al repositorio.

R

El fichero ha sido borrado (pero podría recuperarse del histórico que se almacena en el repositorio).

C

Hay un conflicto: otra persona ha modificado antes las mismas partes del fichero y no le queda más remedio que revisarlo manualmente antes de poder subir los cambios de ese fichero.

M

Usted ha modificado el fichero pero aún no lo ha subido al repositorio.

?

CVS se ha encontrado un fichero que no tiene registrado y simplemente avisa y lo deja en paz.

Añadir ficheros al módulo

No olvides que cvs controlará sólo los ficheros que se le hayan indicado expresamente. Cualquier otro fichero en el directorio cvs será ignorado.

Si quieres incluir un nuevo fichero o directorio al módulo cvs hay que seguir los siguientes pasos:

```
$ cd [directorio-del-módulo]
$ cvs add [fichero]
```

pero si el fichero es binario hay que tener la precaución de hacer:

```
$ cd [directorio-del-módulo]
$ cvs add -kb [fichero]
```

¿Por qué?, se preguntará el lector más intrépido. Resulta que CVS usa varias variables (en realidad son de RCS, que funciona por debajo de CVS). Si el fichero es binario es posible que se de una combinación de bytes que coincidan con alguna de estas variables. Si así fuera, RCS/CVS modificaría el contenido y lo corrompería. También se debe a que el sistema de cálculo de diferencias que usan estos sistemas no está diseñado para trabajar con información binaria. Si se obra equivocadamente es probable que corrompamos los datos.

También quiero señalar que si bien se pueden gestionar ficheros binarios, no se hará control de versiones de los mismos. Sólo se guardará la última versión.

Tras el «add» hay que hacer un «cvs ci» para actualizar el servidor con los nuevos ficheros.

Eliminar ficheros del módulo cvs

para eliminar un fichero del módulo cvs hay que hacer lo siguiente una vez borrado el fichero:

```
$ cd [directorio-del-módulo]
$ cvs remove [fichero]
```

En cambio, si queremos borrar físicamente los ficheros a la vez que los eliminamos del módulo deberemos usar:

```
$ cd [directorio-del-módulo]
$ cvs remove -f [fichero]
```

Cómo configurar un servidor y cómo subir nuevos módulos al repositorio.

Si este documento es tu referencia favorita sobre cvs no merece la pena que te explique cómo se hacen estas cosas porque de todas formas no te ibas a enterar. Espabila un poco porque es bien fácil una vez que uno se ha mirado los manuales.

Si alguien con mejor corazón que yo tiene tiempo, que amplíe esta sección para beneficio de todos.

Bibliografía

Por un lado hay un completísimo fichero info dedicado a cvs. Si usa GNU/Linux es muy probable que ya lo tenga instalado en su sistema.

Por otro lado en <http://cvsbook.red-bean.com/> está disponible otro libro documentando cvs.

interfaces gráficos

Espero que si llegas a este punto es por curiosidad o por la inercia de la lectura porque los verdaderos tecnófilos tienen que saber exactamente qué es lo que hacen en cada momento sin que ninguna clase de software intente facilitarles la vida encapsulando complejidades. Sobre todo cuando ese software ofrece interfaces gráficos.

En cualquier caso y para que nadie pueda dudar de mi profesionalidad voy a enumerar unos cuantos interfaces gráficos cvs en mayor o menor estado de desarrollo:

- pharmacy [<http://pharmacy.sourceforge.net/index.html>]
- gcvs [<http://freshmeat.net/projects/gcvs/?highlight=gcvs>]
- tkcvs [<http://www.twobarleycorns.net/tkcvs.html>], para unix y windows y que además saca un bonito grafo con el historial de desarrollo.
- Tortoise [<http://www.tortoisecvs.org/>], para windows, integrado en el gestor de ficheros.
- wincvs [<http://www.wincvs.org>], para windows, como bien sugiere el nombre. He oído hablar mal de él, pero como no lo he probado no puedo asegurar nada.
- cervisia [<http://freshmeat.net/projects/cervisia/?highlight=cvs>], para KDE.
- lincvs [<http://freshmeat.net/projects/lincvs/>]

Bueno, ya me aburrí. para encontrar los enlaces que faltan, la mayoría aparecerán en <http://freshmeat.net> y con el tiempo tal vez aparezcan nuevas herramientas.

Otros recursos cvs interesantes

- La página principal que creo que ahora está en <http://www.cvshome.org/>.
- cvsq, una utilidad para trabajar sin conexión permanente a la red: <http://freshmeat.net/news/2000/07/02/962595201.html>.
- Enlaces [<http://www.loria.fr/~molli/cvs-index.html>] a documentación y otros recursos.
- cvs2cl.pl, que es una herramienta para crear ficheros *Changelog* al estilo GNU y que puede encontrarse en <http://www.red-bean.com/cvs2cl/>

Otros recursos cvs más avanzados

- cvsadmin [<http://freshmeat.net/projects/cvsadmin/?highlight=cvsadmin>] es una herramienta para administrar las cuentas de un repositorio.
- cvs-nserver [<http://alexm.here.ru/cvs-nserver/>] es una reescritura y extensión de cvs para mejorar sus capacidades en red.
- cvsauth [<http://freshmeat.net/projects/cvsauth/?highlight=cvsauth>] sirve para autenticar usuarios sin ejecutar en el servidor cvs como root.

Licencia

Este documento está bajo los auspicios de la licencia de documentación libre de GNU que puede encontrarse en <http://www.gnu.org>