SourceForge guide

Tom Huckstep

tomhuckstep@bigfoot.com

This guide is intended to get newbieDoc up to speed on the development services provided by SourceForge (http://www.sourceforge.net/).

1. Introduction

1.1. What is SourceForge?

From the SourceForge FAQ (http://sourceforge.net/docman/display_doc.php?docid=756&group_id=1#what-main): “SourceForge is a free hosting service for Open Source developers which offers, among other things, a CVS repository, mailing lists, bug tracking, message forums, task management software, web site hosting, permanent file archival, full backups, and total web-based administration.”

1.2. Why use SourceForge?

All of the resources listed above allow any group to easily establish a development environment through which work can be done quickly, easily and safely.

For example, we at the newbieDoc project use SourceForge because it allows us to take advantage of web space, mailing lists, and other features listed in the section *What is SourceForge?*. The CVS repository is particularly useful as it allows easy peer review of the files it contains.

2. Signing up

To use SourceForge for development, you need to register as a site user.

Head on over to http://www.sourceforge.net/ and follow the link, ‘New User Via SSL (http://sourceforge.net/account/register.php)’. There, you will be presented with the SourceForge New Account Registration page. You will need to fill in the following details:

- Login Name -- Set it to anything you like. This will also be your SourceForge email address.

- Password (min. 6 chars) -- Your SourceForge password is used for nearly everything you might want to do at the site. Choose it carefully.
- Full/Real Name--This one is required.
- Language Choice -- If your preferred language is anything but English, change it here. If needed, this can be changed later.
- Timezone -- By default, this is set to GMT (Greenwich Mean Time). While you are not required to set this correctly, it is helpful as it can tell people when they might expect a response to their email.
- Email Address -- Your email address must be correct or you will never be able to use your account at SourceForge. This address will not be shown on SourceForge's web site, but you will receive a mail forwarding account at <login@users.sourceforge.net> that will forward to this address.

After you have completed that you will receive an email which will send you to the account verification page. This is where you will login with your new username and password.

To join the newbieDoc developers list you must contact one of the project administrators. To find out who these are, have a look at http://sourceforge.net/projects/newbiedoc/ (http://sourceforge.net/projects/newbiedoc/). Follow one or more of the links in the developer info box to find contact details. Send an e-mail to any or all of the admins explaining that you wish to become a developer, and be sure to *tell them your SourceForge user name*.

3. Web services

SourceForge provides a web-based interface to most of the functions it provides. All project administration is done through the web, as well as bug tracking, technical support, task management and more. The entry point to all this is the project's home page. For example, the home page for newbieDoc is at http://sourceforge.net/projects/newbiedoc/.

4. SourceForge Documentation

SourceForge provides good documentation on the services it provides. You can get it at http://sourceforge.net/docman/?group-id=1 (http://sourceforge.net/docman/?group_id=1), or by following the 'Site Docs' link.

5. ssh services

In the interests of security, SourceForge does all its communication with logged-in users over secure encrypted channels. This is the case even for the web-based services. There are two important utilities that cannot be accessed via a web interface, but instead by using **ssh**, the secure shell. These utilities are the shell server and CVS.

5.1. Installing ssh

Note: Debian-specific

```
# apt-get install ssh
```

There are two versions of ssh available in Debian. As of the Woody release (Debian 3.0), ssh is included in the main distribution but ssh2 is only available under the non-free section. For our purposes, normal ssh is perfectly fine.

5.2. The shell server

The shell server provides command line access to the SourceForge system.

Login via

```
$ ssh username@shell1.sourceforge.net
```

5.3. CVS

CVS is used to organize and store collections of files in a central repository, so that many developers can work on them at once. More than one developer can work on the same file at a time. If one change conflicts with somebody else's change, CVS will warn the second user and attempt to solve the problem.

5.3.1. Installing CVS

Note: Debian-specific

```
apt-get install cvs
```

5.3.2. Using CVS for newbieDoc

Before using CVS with SourceForge you need to make sure the environment variable CVS_RSH is set to 'ssh'. For bash you can set it like this:

```
$ export CVS_RSH=ssh
```

You can add this to your `~/.bashrc` to have the variable set it every shell.

5.3.2.1. Check out a working copy

First of all you must check out a working copy of the repository. This contains all the newbieDoc source files. Go to the directory in which you wish your local copy to be kept, e.g. `~/cvs`.

```
$ mkdir ~/cvs
$ cd ~/cvs
```

To check out a working copy, issue the command

```
$ cvs -d:ext:username@cvs.newbiedoc.sourceforge.net:/cvsroot/newbiedoc checkout newbiedoc
```

Be sure to make the obvious substitution for ‘username’.

5.3.2.2. Make and commit your own changes

If you make changes to a file already in the repository you will need to ‘commit’ them so that the changes can be seen by everyone in the project and become the official copy of the file.

```
$ cd ~/cvs
$ cvs commit -m 'comments for this commit' filename
```

Alternatively, you can omit the -m argument ($ **cvs** commit *filename*). Your preferred editor, as specified in the EDITOR environment variable, will appear for you to note your changes. The commit will then continue once the file is saved.

Your comments can be a short note, summarizing your changes.

Another useful option for committing files is ’-r’, which forces a particular version to be assigned to the file, regardless of what CVS might automatically want to assign it. For instance, if the current version was 2.5, CVS would want to name the next version 2.6. Of course, if you think there are enough changes to warrant a completely new version designation, you can use the following to force it:

```
$ cvs commit -r3.0 filename
```

5.3.2.3. Add a new file

```
$ cd ~/cvs
$ emacs my_new_file.sgml
$ cvs add my_new_file.sgml
$ cvs commit
```

Of course, you may substitute your favorite editor for emacs.

5.3.2.4. Update your working copy

To get the changes that the other developers have added to the repository since you last checked, do

```
$ cd ~/cvs
$ cvs update
```

If a new directory has been added to the central repository since your last checkout you can do

```
$ cvs update -d
```

to get the new directory and its contents.

5.3.3. Learn more about CVS

The CVS book at http://cvsbook.red-bean.com/ has fairly comprehensive coverage.

6. Communication utilities

SourceForge provides the opportunity for both mailing lists and web-based forums. Go to http://lists.sourceforge.net/lists/listinfo/newbiedoc-discuss (http://lists.sourceforge.net/lists/listinfo/newbiedoc-discuss) to subscribe to the mailing list. You don't have to be a member of newbiedoc or sourceforge to subscribe to the mailing list. So please subscribe and pass on any ideas or thoughts you may have! The web based forums are located here: http://sourceforge.net/forum/?group_id=18089 (http://sourceforge.net/forum/?group_id=18089) in case you do not want to subscribe to the mailing list.

7. Advanced ssh

You can log into the SourceForge shell and CVS servers without using your password, but instead using a public/private key pair. This means you never have to send your password over the network -- encrypted or otherwise -- and you only have to type your password once when you log on to your local workstation.

7.1. A little background

Normally, when you log on to a UNIX system, the password you give -- after having been sent over the network if it is a remote system -- is compared with that in `/etc/passwd` or `/etc/shadow` and if it matches then you are allowed to log in.

There is another method based on public/private keys which ssh allows. In this situation your public key, stored on the system you wish to log on to, is used by **sshd** to encrypt a very large random number. This is sent across the network back to you. If your ssh on the local system can decrypt it (using your private key), and send it back to the remote system correctly, then you are allowed to log in.

7.2. Setting up your public/private key

First you need to generate you public and private key pair. You do this using **ssh-keygen**. You can just keep pressing **Enter** to select all of the default values:

```
$ ssh-keygen
Generating RSA keys:  ........................ooooo0.................ooooo0
Key generation complete.
Enter file in which to save the key (/home/tom/.ssh/identity):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/tom/.ssh/identity.
Your public key has been saved in /home/tom/.ssh/identity.pub.
The key fingerprint is:
1024 3d:50:a1:6d:96:2e:5f:18:2e:6f:8f:1f:32:25:c8:9d tom@henry
```

7.3. Setting up the server side

7.3.1. The shell server

Now you need to copy your public key to the SourceForge shell server, so it can authenticate you.

The (public) keys that you are allowed to log in with are stored in `~/.ssh/authorized_keys` on the server, one per line. If you don't have any keys there yet then it's possible to **scp** `localhost:~/.ssh/identity.pub` directly over to `server:~/.ssh/authorized_keys`.

```
$ scp ~/.ssh/identity.pub \
> myusername@shell1.sourceforge.net:/home/user/users/m/my/myusername/.ssh/authorized_keys
```

Where 'm' and 'my' are the first and first two letters of your username.

Now you can

```
$ ssh username@shell1.sourceforge.net
```

without having to type your password

7.3.2. The CVS server

Because you can't log directly into the CVS server you have to copy your authorized keys via a web page. Log into your account on sourceforge.net (http://sourceforge.net/) then head over to sourceforge.net/account/editsshkeys.php (http://sourceforge.net/account/editsshkeys.php). Copy the contents of your `~/.ssh/identity.pub` into the box. It may take a few hours after submission to sync the key over to the CVS server. When that's done you can

```
$ cvs [update] [commit] etc...
```

without having to type your password.

7.4. Protecting your private key

Although your private key is set to be only readable by you, the root user of your system can also see it. If you don't trust your admins that much, or just want some extra security then you can put a passphrase on your key.

```
$ ssh-keygen -p
Enter file in which the key is (/home/tom/.ssh/identity):
Key has comment 'tom@henry'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

> Your choice of passphrase is not as restrictive as your choice of UNIX password is. It can contain all sorts of characters, including spaces and can be very long.

7.4.1. ssh-add

Now every time ssh wants to use your private key it will ask you for your passphrase. This rather defeats one of the objects of using ssh, which was to avoid having to type your password. However, a program called **ssh-agent** can hold any and all of your private keys in memory. If ssh-agent is running, ssh will ask it rather than you for your password.

When starting X, Debian Potato systems run ssh-agent automatically. If you want this to work in a terminal shell or on a system that does not run ssh-agent automatically then check the man page. ssh-agent has a unusual syntax.

If you try

```
$ ssh-add ~/.ssh/identity
```

ssh-add will add your private key to ssh-agent's list. If your public key is encrypted then you will be asked for your passphrase. You can add ssh-add to your `~/.xsession` file, but if ssh-add is not attached to a console when it is run you will need one of the ssh-askpass* packages. These provide a way for ssh-add to get your passphrase in X.

```
$ apt-get install ssh-askpass
$ apt-get install ssh-askpass-ptk
$ apt-get install ssh-askpass-gnome
```

7.5. Concluding ssh

Using ssh with your own public and private keys allows secure communication with a remote machine, and you never have to send any password across a network. Using ssh-agent means you only have to type your passphrase once per session.

8. Further information

8.1. SourceForge

- SourceForge site documentation: http://sourceforge.net/docman/?group-id=1

8.2. CVS

- The CVS book: http://cvsbook.red-bean.com/