

Using Glast CVS on Windows

The [official glast instructions](#) for using CVS on windows describe how to use WinCVS. These are my own unofficial instructions, they differ in that:

- They use TortoiseCVS instead of WinCVS. TortoiseCVS integrates directly into windows explorer and provides more functionality that WinCVS
- They do not require that you use an insecure ssh key with no pass phrase

Installing TortoiseCVS

Download TortoiseCVS from <http://www.tortoisecvs.org/download.shtml>. These instructions were written using version 1.8.11. Install the downloaded .exe in the normal way.

TortoiseCVS integrates CVS capabilities directly into the windows shell (aka explorer). If you already have some code checked out from CVS just open the folder with the checked out code in explorer and you will see that Tortoise CVS has added new icons to all the files to show their status and has added new items to the popup context menus for all files and folders. You can now perform all the normal CVS commands directly from windows explorer.

If you want to checkout a new module from CVS navigate to the folder you want to check the code out in, and select "CVS checkout" from the explorers File menu. Fill the form in using the values below as a guide, then click OK to check-out the new module.

Previous CVSROOTs	Module
:pserver:tonyj@london.jaws.com:/home/cvs/root	SystemTests
:ext:tonyj@centaurusa.slac.stanford.edu:/nfs/slac/g/glast/ground/cvs	users/tonyj/Pipeline
:ext:tonyj@ext:centaurusa.slac.stanford.edu:/nfs/slac/g/glast/ground/cvs	users/tonyj/Pipeline
:pserver:tonyj@cvs.dev.java.net:/cvs	jspreadsheet
:pserver:tonyj@cvs.dev.java.net:/cvs	jnn

CVSROOT:

Protocol:

Protocol parameters:

Server:

Port:

Repository folder:

User name:

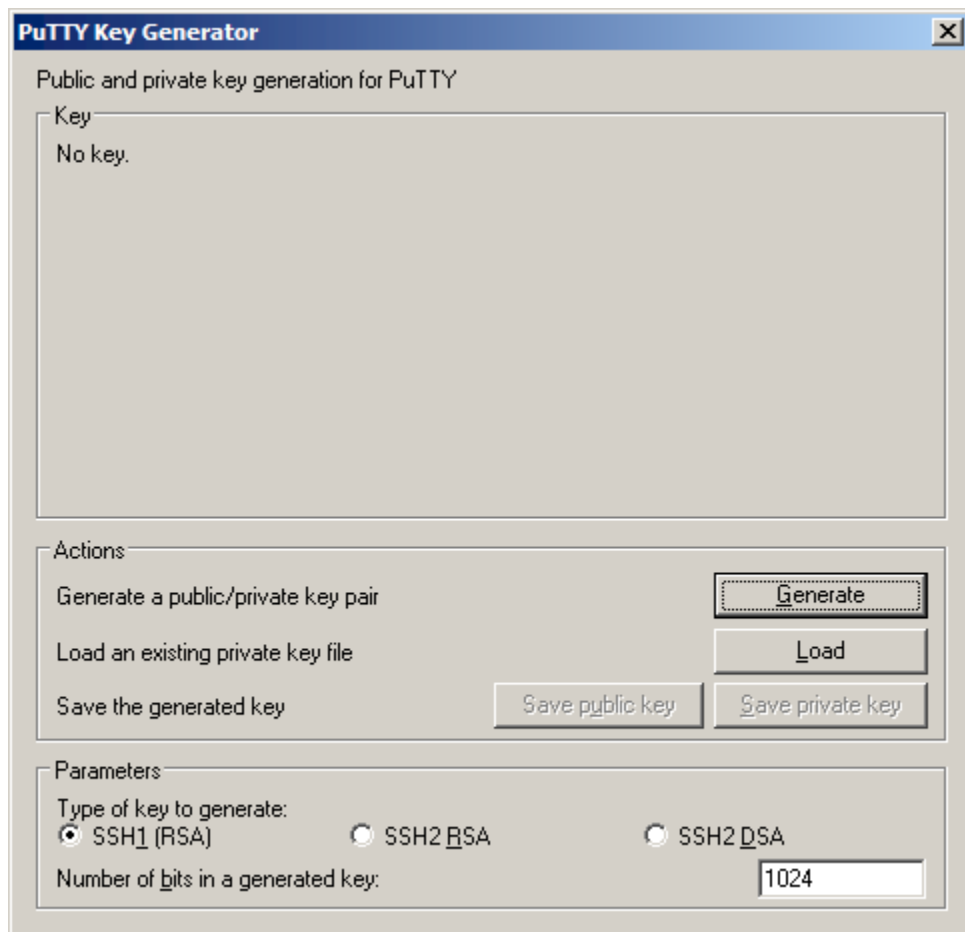
Module:

You will have noticed if you have followed the instructions above the Tortoise will prompt you for your password each time you perform a CVS operation. This is mildly annoying, so in the next section I provide instructions for fixing this.

Setting up ssh so that you don't have to give you password each time

The official Glast instructions describe how to create an ssh private/public key pair with no password. This is generally considered to be a "bad thing(TM)" since it allows anyone who manages to hack into your machine to gain access to SLAC. Hackers love this kind of thing, SLAC security does not. The same thing can be done using TortoiseCVS, but in these instruction I describe how to do a little more work in order to be a good security-conscious network citizen.

Create a new public/private key pair. In the directory where you installed TortoiseCVS (probably C:\Program Files\TortoiseCVS) you will find a file called puttygen.exe. Double-clicking this will produce a dialog like this:



The image shows the 'PuTTY Key Generator' dialog box. It has a title bar with the text 'PuTTY Key Generator' and a close button. The main area is titled 'Public and private key generation for PuTTY'. It contains a large text area labeled 'Key' with the text 'No key.' inside. Below this is a section labeled 'Actions' with three buttons: 'Generate', 'Load', and 'Save the generated key'. The 'Save the generated key' button is disabled. Below the 'Actions' section is a section labeled 'Parameters' with two radio buttons: 'SSH1 (RSA)' (selected) and 'SSH2 RSA'. There is also a 'SSH2 DSA' option which is disabled. At the bottom, there is a text box labeled 'Number of bits in a generated key:' with the value '1024' entered.

PuTTY Key Generator

Public and private key generation for PuTTY

Key

No key.

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

Generate

Load

Save public key

Save private key

Parameters

Type of key to generate:

☒ SSH1 (RSA) ☐ SSH2 RSA ☐ SSH2 DSA

Number of bits in a generated key: 1024

To generate a new key-pair select "SSH2 RSA" at the bottom and then click "Generate" and wiggle your mouse around in the dialog to generate some randomness. After a while your key-pair will be generated and the dialog will look like this:

PuTTY Key Generator

Public and private key generation for PuTTY

Key

Public key for pasting into OpenSSH authorized_keys2 file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAIEAur8G036bIF8Mw85ieus7RwTD78yeh330rlMBhi
upNullw/8UGwTzBwE7JFTIss1uW/seF1/CGH4LZolBRx2b56Dr3h7u02LPuchORM+w
clSxcUMwB+6qqEkvRA3wmQ+iKKJwGdQUvGklK6Ppsh6D1H/1YFhCV0qGcmDdLD
K/zDqCU= rsa-key-20050129
```

Key fingerprint: ssh-rsa 1024 e0:04:93:46:e7:6a:23:e4:b3:12:ce:13:20:38:64:f0

Key comment: rsa-key-20050129

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair Generate

Load an existing private key file Load

Save the generated key Save public key Save private key

Parameters

Type of key to generate:

☐ SSH1 (RSA) ☒ SSH2 RSA ☐ SSH2 DSA

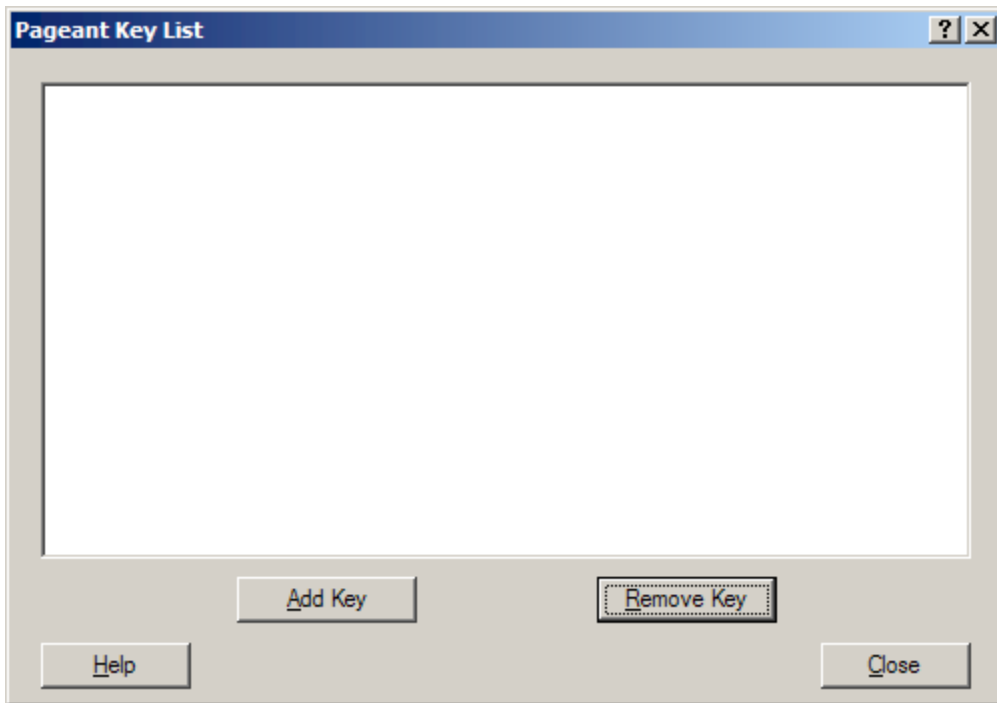
Number of bits in a generated key: 1024

Enter a passphrase for the private key (twice) and then click "save private key". Save the file in a local directory (preferably one only you have access to) with the name identity.ppk (actually you can call it anything you like, but the filetype of ppk is a good idea). Don't forget the passphrase you entered, you will need it again in a minute.

Next log in to SLAC unix (e.g. noric.slac.stanford.edu), and cd to ".ssh/public". Open the file authorized_keys2 using your favorite editor. Now select and copy the string labelled "Public key for pasting into OpenSSH authorized_keys2 file:" from the puttygen dialog, and surprise-surprise, paste it into your authorized_keys2 file. Save the file and double-check that the filemode is "-rw-r--". If not fix it with:

```
chmod 644 authorized_keys2
```

OK, we are nearly done. Finally download pageant.exe from [here](#) and save it in the same directory as puttygen.exe. Double-click on pageant.exe and nothing will happen except a small icon showing a computer with a hat on will appear in the windows taskbar. Double-click on this icon, and a dialog will appear like this:



Click on "Add key" and select the identity.ppk file that you saved earlier. You will have to give the passphrase you entered earlier (you did remember it, right?). Now you can close the dialog. Pageant is an agent which stores your private key in memory, and provides it to other programs as they need it.

Now go back to TortoiseCVS and try performing some CVS operation. By magic you should no longer have to give your password, since Tortoise will get your private key from pageant and use it to authorize you to SLAC unix using your public key. If it doesn't work go back and check the instruction carefully to see what you did wrong.

A few additional notes:

- Each time you log in to windows you will have to restart pageant and give your passphrase. If like me you stay logged in for weeks, this is no big deal. You can slightly simplify things by creating a shortcut to start pageant and load in your identity.ppk automatically. To do this create a shortcut with a target of: "C:\Program Files\TortoiseCVS\pageant.exe" "c:\Documents and Settings\Tony Johnson\My Documents\identity.ppk" This shortcut will then start pageant and load your identity (it will still prompt for a password).
- pageant and puttygen are part of PuTTY, a terminal emulator with support for SSH. If you choose to use PuTTY too it can share the same ssh keys as TortoiseCVS. PuTTY can be found at <http://www.chiark.greenend.org.uk/~sgtatham/putty/>.