Sourceforge CVS Access

Introduction

This document is intended to help you connect to the CVS server on a Windows system.

Sourceforge uses SSH2 encryption to secure access to its servers. The following will help you get set up pretty easily.

Tools

First thing we need is the right tools. Download the following items.

Putty – the SSH tools – (version as of writing is 0.55)
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

- PuTTY
- Pageant
- PuTTYgen
- Note: all the tools can be downloaded in one zip file

TortoiseCVS – the CVS client – (version as of writing is 1.6.1)
http://tortoisecvs.sourceforge.net/download.shtml

- TortoiseCVS

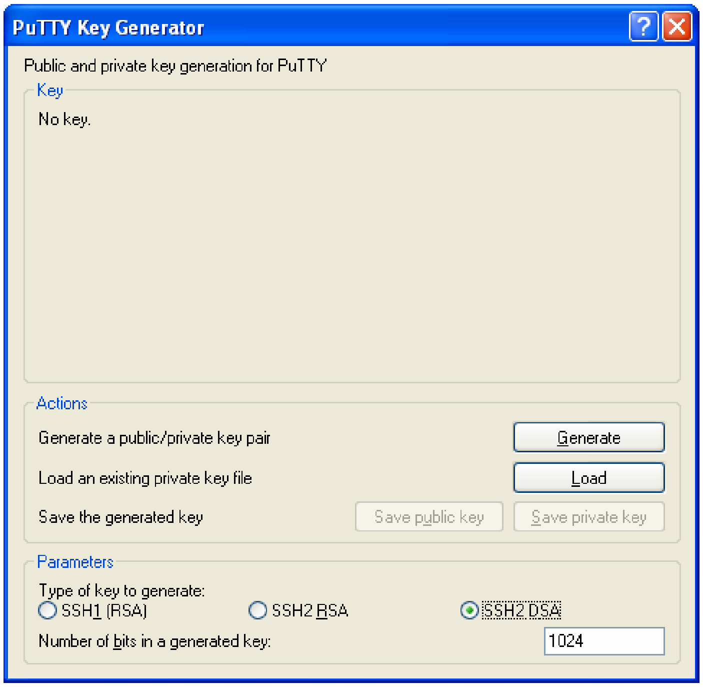
See addendum at end of document on the following updated software versions:

- TortoiseCVS 1.8.1

Generate Public / Private Keys

The keys are important for unique authentication and so you don't have to enter a password every time you access the servers.

Run the PuTTYgen program.



Public and private key generation for PuTTY

No key.

Generate a public/private key pair

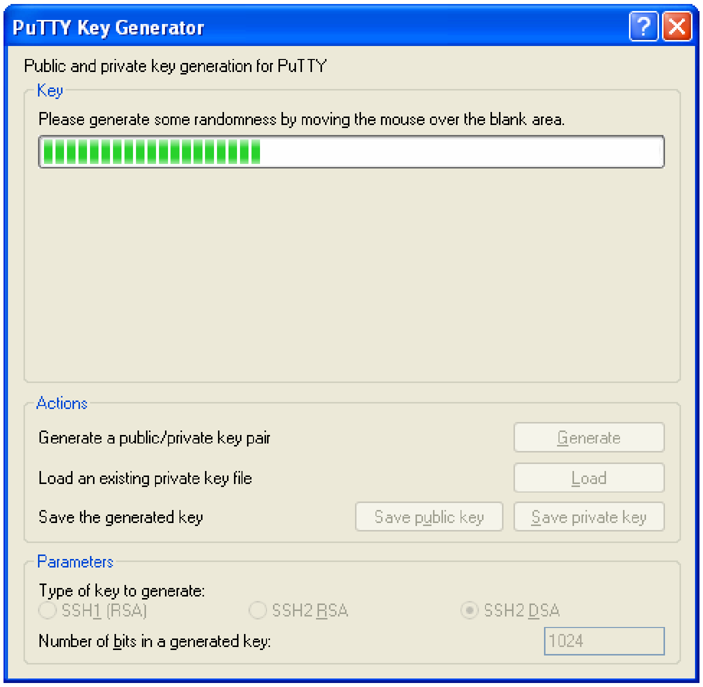
Load an existing private key file

Save the generated key

Type of key to generate:
○ SSH1 (RSA) ○ SSH2 RSA ● SSH2 DSA

Number of bits in a generated key: 1024

1. Be sure to select the SSH2 DSA option at the bottom



Public and private key generation for PuTTY

Please generate some randomness by moving the mouse over the blank area.

Generate a public/private key pair

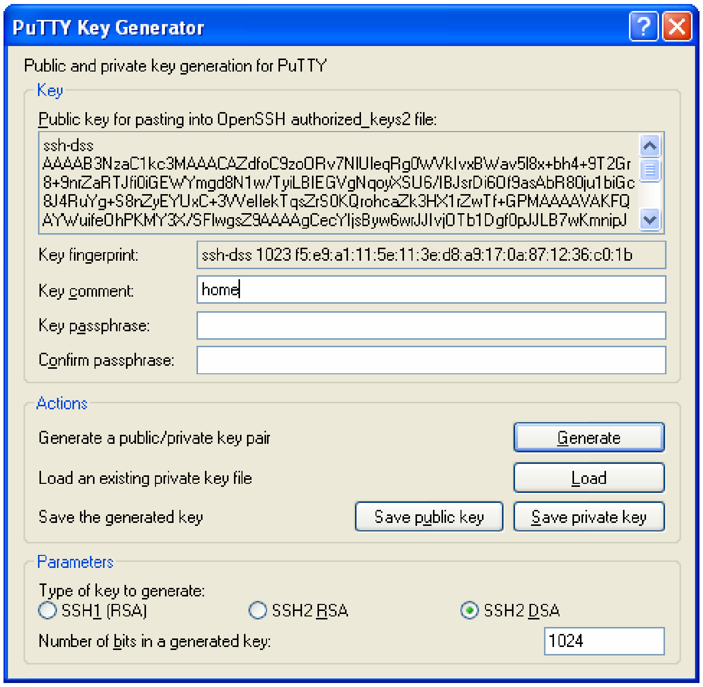
Load an existing private key file

Save the generated key

Type of key to generate:
○ SSH1 (RSA) ○ SSH2 RSA ● SSH2 DSA

Number of bits in a generated key: 1024

2. Click the generate button and move the mouse in the grey area until the key is full generated



Public key for pasting into OpenSSH authorized_keys2 file:

```
ssh-dss
AAAAB3NzaC1kc3MAAACAZdfoC9zoORv7NIUleqRg0WVkIvxBWav5l8x+bh4+9T2Gr
8+9nrZaRTJfi0iGEWYmgd8N1w/TyiLBIEGVgNqoyXSU6/IBJsrDi6Of9asAbR80ju1biGc
8J4RuYg+S8nZyEYUxC+3VVellekTqsZrS0KQrohcaZk3HX1rZwTf+GPMAAAAVAKFQ
AYWuifeOhPKMY3X/SFlwgsZ9AAAAgCecYIjsByw6wrJJIvjOTb1Dgf0pJJLB7wKmnipJ
```

Key fingerprint: ssh-dss 1023 f5:e9:a1:11:5e:11:3e:d8:a9:17:0a:87:12:36:c0:1b

Key comment: home

Key passphrase:

Confirm passphrase:

Generate a public/private key pair

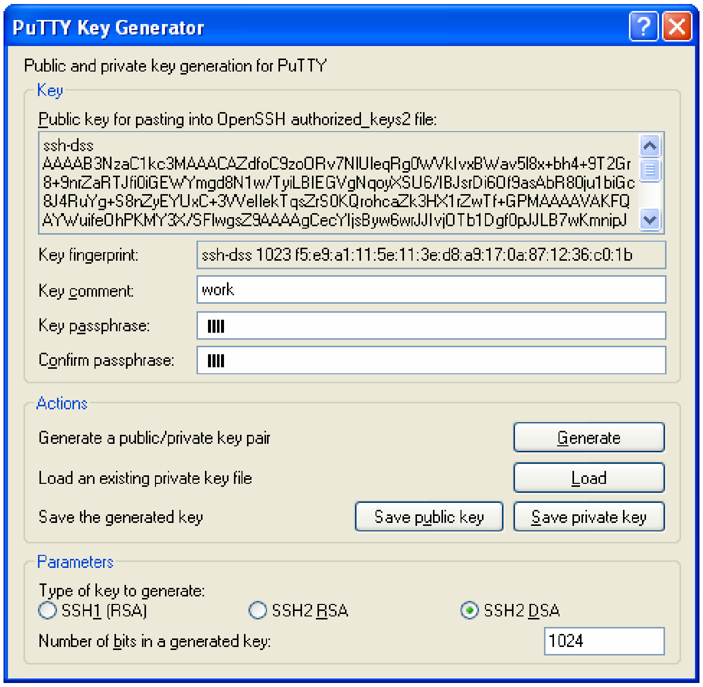
Load an existing private key file

Save the generated key

Type of key to generate:
○ SSH1 (RSA) ○ SSH2 RSA ● SSH2 DSA

Number of bits in a generated key: 1024

3. Enter a comment you can use to identify the key. You can have multiple keys for different computers. I have one for work and one for home. That way if the work key is compromised I can replace it without changing my home key. (make sure the key generated is SSH2 DSA or it won’t work. I’ve been caught on this before).



Public key for pasting into OpenSSH authorized_keys2 file:

```
ssh-dss
AAAAB3NzaC1kc3MAAACAZdfoC9zoORv7NIUleqRg0WVkIvxBWav5l8x+bh4+9T2Gr
8+9nrZaRTJfi0iGEWYmgd8N1w/TyiLBIEGVgNqoyXSU6/IBJsrDi6Of9asAbR80ju1biGc
8J4RuYg+S8nZyEYUxC+3VVellekTqsZrS0KQrohcaZk3HX1rZwTf+GPMAAAAVAKFQ
AYWuifeOhPKMY3X/SFlwgsZ9AAAAgCecYIjsByw6wrJJIvjOTb1Dgf0pJJLB7wKmnipJ
```

Key fingerprint: ssh-dss 1023 f5:e9:a1:11:5e:11:3e:d8:a9:17:0a:87:12:36:c0:1b

Key comment: work

Key passphrase: ||||

Confirm passphrase: ||||

Generate a public/private key pair

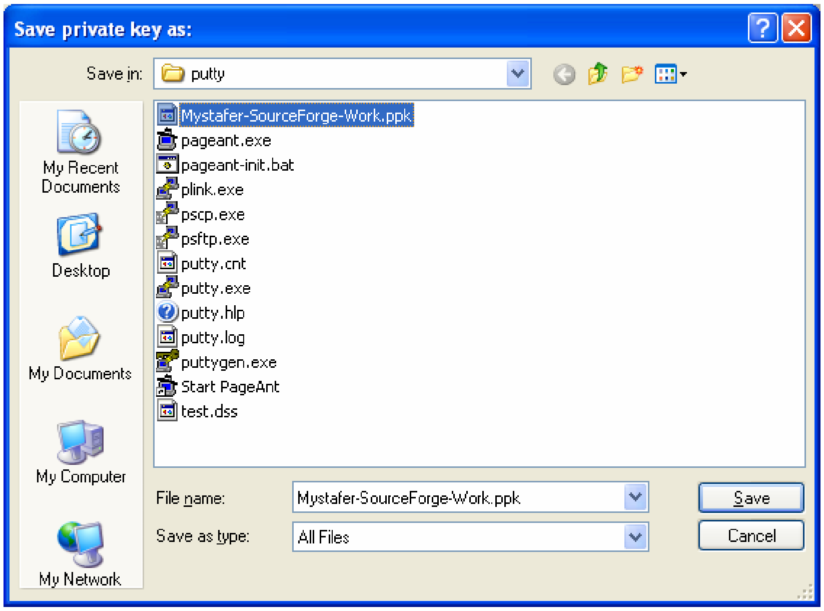
Load an existing private key file

Save the generated key

Type of key to generate:
○ SSH1 (RSA) ○ SSH2 RSA ● SSH2 DSA

Number of bits in a generated key: 1024

4. Optionally enter a passphrase for the key. The passphrase is used to keep anyone else from being able to use your key without a password. This would be good if you share your machine with others. However, it sucks to have to enter the password everytime you reload the key (say on reboot, etc)



Save private key as:

Save in: putty

File name: Mystafer-SourceForge-Work.ppk

Save as type: All Files

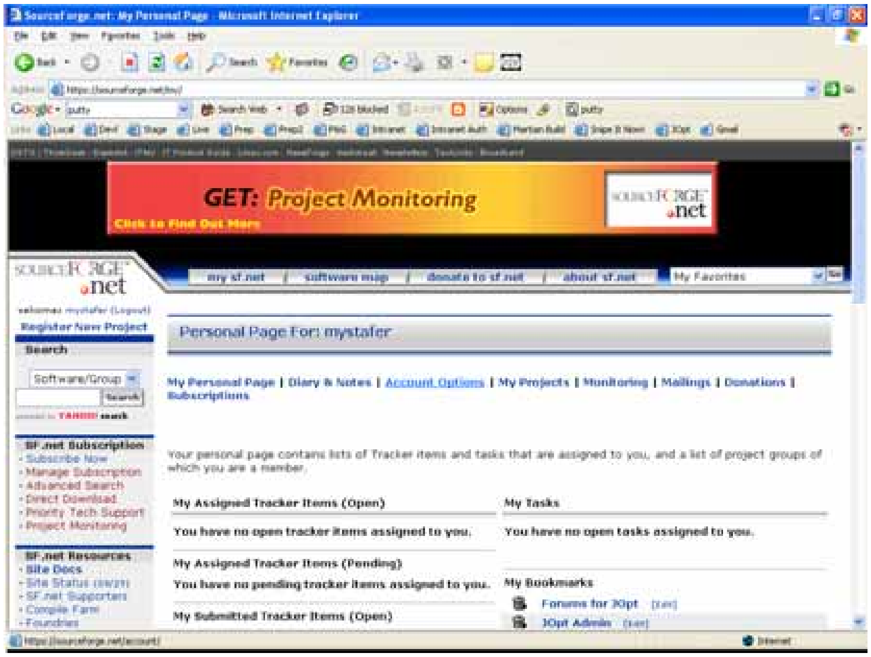
5. Click the ‘Save private key’ button. Most keys end with a .ppk extension. Don’t worry about saving the public key since you can load the private key into PuTTYgen to retrieve it later if needed.

6. Copy the public key text in the large list box area to the clipboard for use in the next step. The string should start with ssh-dss and end with whatever the comment was in your comment box. This is the data that will be given to the sourceforge site.

Configuring Sourceforge Account

Now that you have a public / private key pair, you need to give load the public key into your Sourceforge account. Once sf has your public key, the sf servers will be able to authenticate you based on your private keys.

Go to sourceforge and log in to your account.
https://sourceforge.net/account



SourceForge.net: My Personal Page - Microsoft Internet Explorer

https://sourceforge.net/my/

welcome: mystafer (Logout)

Register New Project

Search

Software/Group

SF.net Subscription
- Subscribe Now
- Manage Subscription
- Advanced Search
- Direct Download
- Priority Tech Support
- Project Monitoring

SF.net Resources
- Site Docs
- Site Status (09/29)
- SF.net Supporters
- Compile Farm
- Foundries

Personal Page For: mystafer

My Personal Page | Diary & Notes | Account Options | My Projects | Monitoring | Mailings | Donations | Subscriptions

Your personal page contains lists of Tracker items and tasks that are assigned to you, and a list of project groups of which you are a member.

My Assigned Tracker Items (Open)

You have no open tracker items assigned to you.

My Assigned Tracker Items (Pending)

You have no pending tracker items assigned to you.

My Submitted Tracker Items (Open)

My Tasks

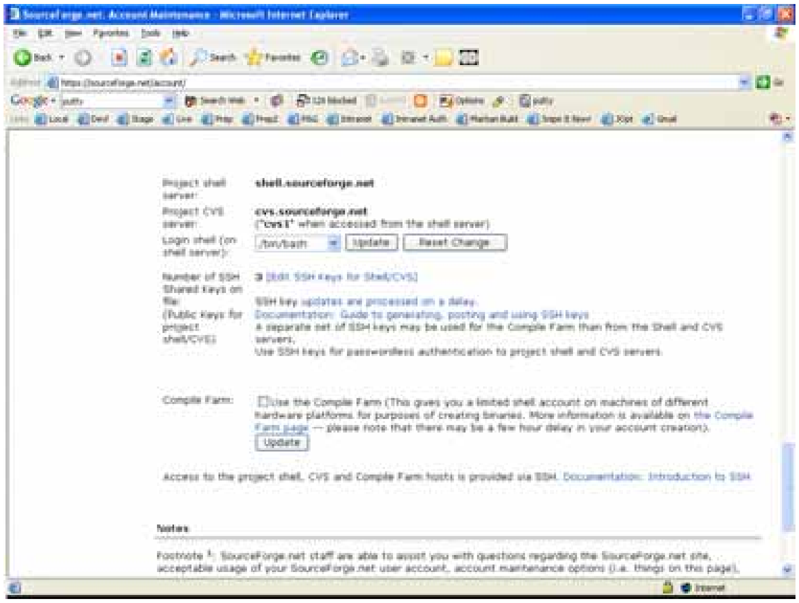
You have no open tasks assigned to you.

My Bookmarks

- Forums for JOpt [Edit]
- JOpt Admin [Edit]

https://sourceforge.net/account/

Click the account options link in the center of the screen



SourceForge.net: Account Maintenance - Microsoft Internet Explorer

https://sourceforge.net/account/

Project shell server:	**shell.sourceforge.net**
Project CVS server:	**cvs.sourceforge.net** ("**cvs1**" when accessed from the shell server)
Login shell (on shell server):	/bin/bash [Update] [Reset Change]
Number of SSH Shared Keys on file: (Public Keys for project shell/CVS):	3 [Edit SSH Keys for Shell/CVS] SSH key updates are processed on a delay. Documentation: Guide to generating, posting and using SSH keys A separate set of SSH keys may be used for the Compile Farm than from the Shell and CVS servers. Use SSH keys for passwordless authentication to project shell and CVS servers.
Compile Farm:	□ Use the Compile Farm (This gives you a limited shell account on machines of different hardware platforms for purposes of creating binaries. More information is available on the Compile Farm page -- please note that there may be a few hour delay in your account creation). [Update]

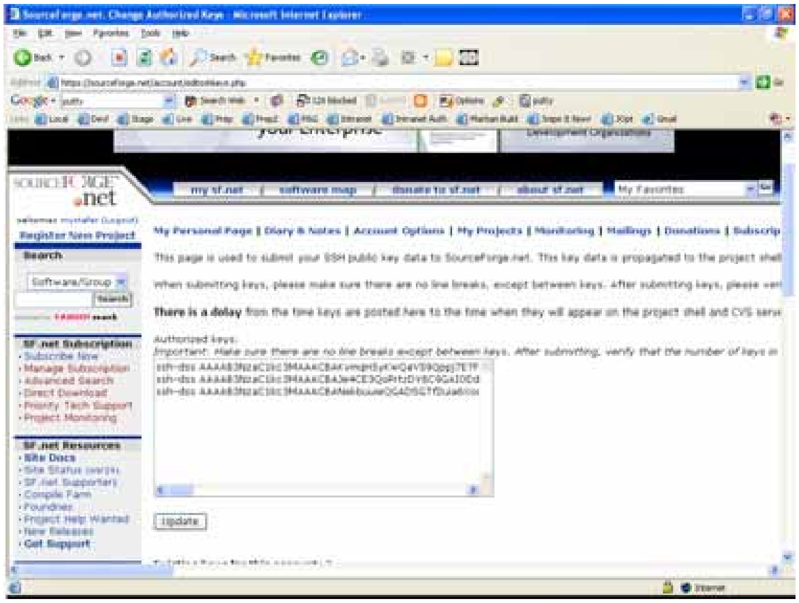
Access to the project shell, CVS and Compile Farm hosts is provided via SSH. Documentation: Introduction to SSH

Notes

Footnote [1]: SourceForge.net staff are able to assist you with questions regarding the SourceForge.net site, acceptable usage of your SourceForge.net user account, account maintenance options (i.e. things on this page),

Scroll down in your account and click the link that says ‘[Edit SSH Keys for Shell/CVS]’

The number next to this link will probably be zero.



This page is used to submit your SSH public key data to SourceForge.net. This key data is propagated to the project shel

When submitting keys, please make sure there are no line breaks, except between keys. After submitting keys, please ver

There is a delay from the time keys are posted here to the time when they will appear on the project shell and CVS serv

Authorized keys:
Important: Make sure there are no line breaks except between keys. After submitting, verify that the number of keys in

```
ssh-dss AAAAB3NzaC1kc3MAAACBAKvmqH5yKwQaV59Qpgj7E7F
ssh-dss AAAAB3NzaC1kc3MAAACBAJe4CE3QoPrtzDY8C9GxI0Dd
ssh-dss AAAAB3NzaC1kc3MAAACBANebuuwQGADSG7fDuJa6ccc
```

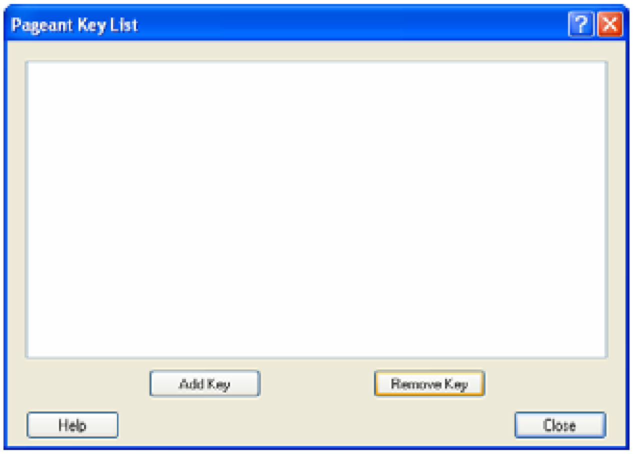
Paste the public key string from the PuTTYgen program into the text box on it’s own line. Then click the update button.

It may take up to 10 minutes before sourceforge will update the shell / cvs servers with the keys so you will need to be a little patient in the following steps if it doesn’t work right away.

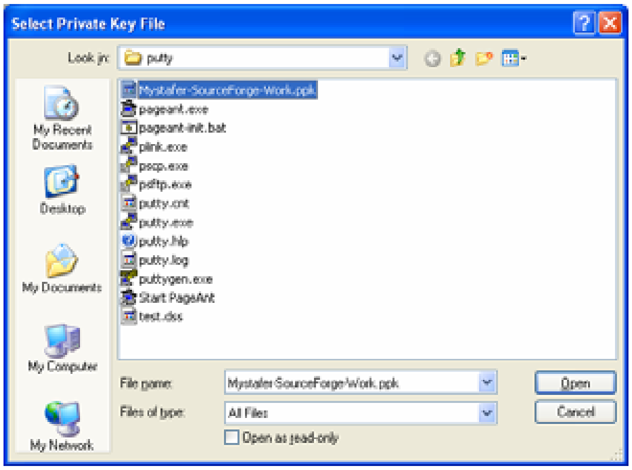
Loading Private Keys for Authentication

The private keys need to be loaded using the Pageant application. This program will need to be running whenever you attempt to access Sourceforge via CVS or you will be prompted with a password over and over.

When pageant is started, it minimizes to the system tray. You will need to locate it and double click it.



It will not have any private keys loaded. Click the ‘Add Key’ button to load the private key you generated earlier.



Look in: putty

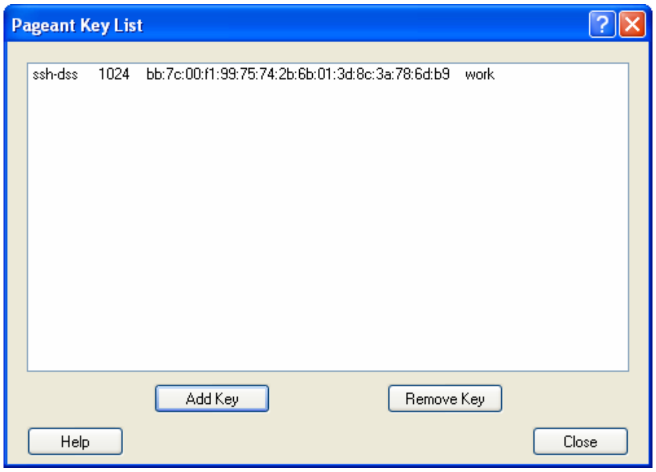
```
Mystafer-SourceForge-Work.ppk
pageant.exe
pageant-init.bat
plink.exe
pscp.exe
psftp.exe
putty.cnt
putty.exe
putty.hlp
putty.log
puttygen.exe
Start PageAnt
test.dxs
```

File name: Mystafer-SourceForge-Work.ppk

Files of type: All Files

□ Open as read-only

If you entered a passphrase when you saved your key in the PuTTYgen program, you will be prompted to enter it at this time.



```
ssh-dss    1024    bb:7c:00:f1:99:75:74:2b:6b:01:3d:8c:3a:78:6d:b9    work
```

Once the key is loaded it should appear similar to this screen. You can click the X button and Pageant should minimize to your system tray.

To automate the process of loading the key so that you don't need to perform these steps every time you reboot, a batch file can be created that can be added to your startup list. This will automatically load the private key file and keep pageant minimized to the system tray.

An example batch file that should work for Windows 2000 / XP systems is as follows:

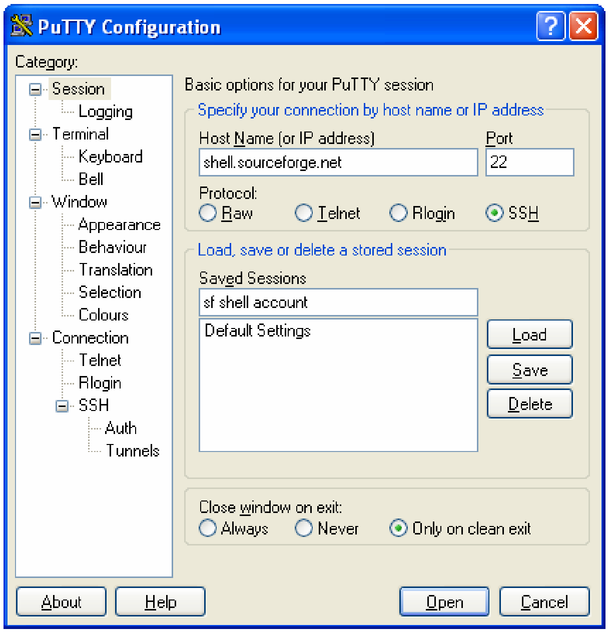
```
@echo off
@if "%OS%" == "Windows_NT"  setlocal

set PUTTYDIR=%~dp0%
start %PUTTYDIR%pageant.exe %PUTTYDIR%Mystafer-SourceForge-Work.ppk
```

Testing Authentication

All sourceforge users are given shell access to the shell.sourceforge.net server. If the SF servers have caught up with the posted public key change and everything is configured right, you should be able to connect to the server without being prompted for a password.

To test the connection we will use the putty.exe program.



Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address): shell.sourceforge.net

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

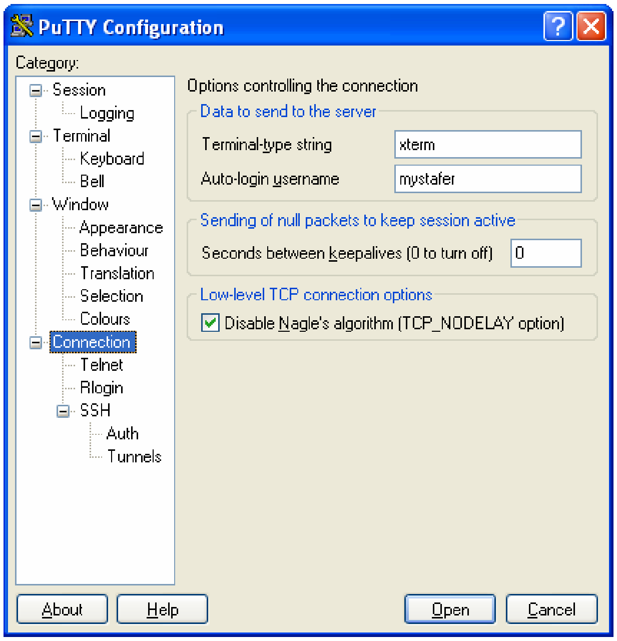
Load, save or delete a stored session

Saved Sessions: sf shell account

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

1. Enter shell.sourceforge.net in the Host Name box
2. Click the SSH button
3. Enter a name in the saved session box



Options controlling the connection

Data to send to the server

Terminal-type string: xterm

Auto-login username: mystafer

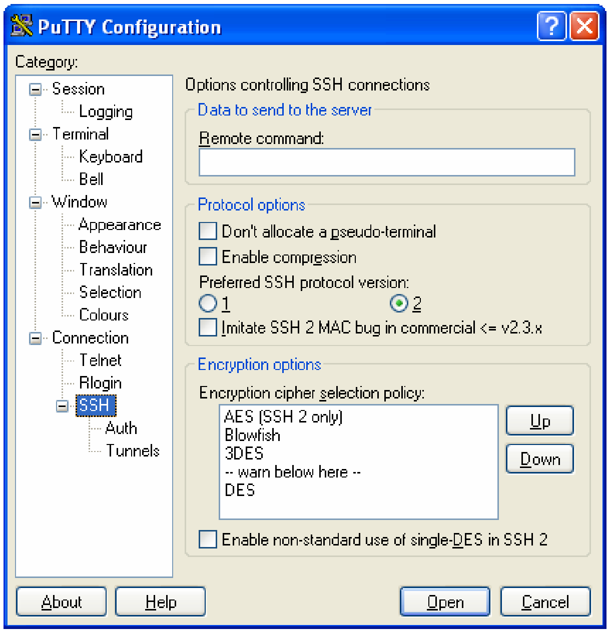
Sending of null packets to keep session active

Seconds between keepalives (0 to turn off): 0

Low-level TCP connection options

☑ Disable Nagle's algorithm (TCP_NODELAY option)

4. Click ‘Connection’ in the category tree
5. Enter your SF username in the ‘Auto-login username’ textbox



Options controlling SSH connections

Data to send to the server

Remote command:

Protocol options

□ Don't allocate a pseudo-terminal

□ Enable compression

Preferred SSH protocol version:

○ 1 ● 2

□ Imitate SSH 2 MAC bug in commercial <= v2.3.x

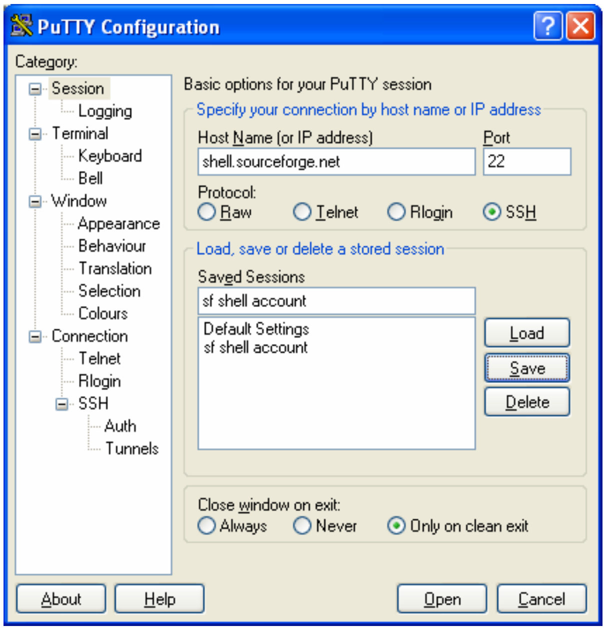
Encryption options

Encryption cipher selection policy:

AES (SSH 2 only)
Blowfish
3DES
-- warn below here --
DES

□ Enable non-standard use of single-DES in SSH 2

6. Click ‘SSH’ in the category tree
7. Select ‘2’ for the ‘Preferred SSH protocol version’



Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address): shell.sourceforge.net

Port: 22

Protocol:

○ Raw ○ Telnet ○ Rlogin ● SSH

Load, save or delete a stored session

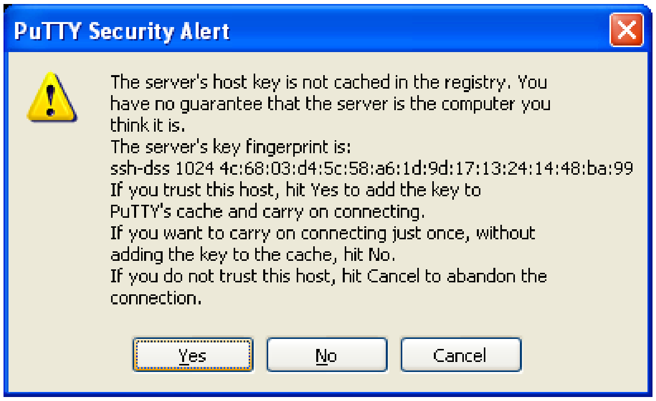
Saved Sessions: sf shell account

Default Settings
sf shell account

Close window on exit:

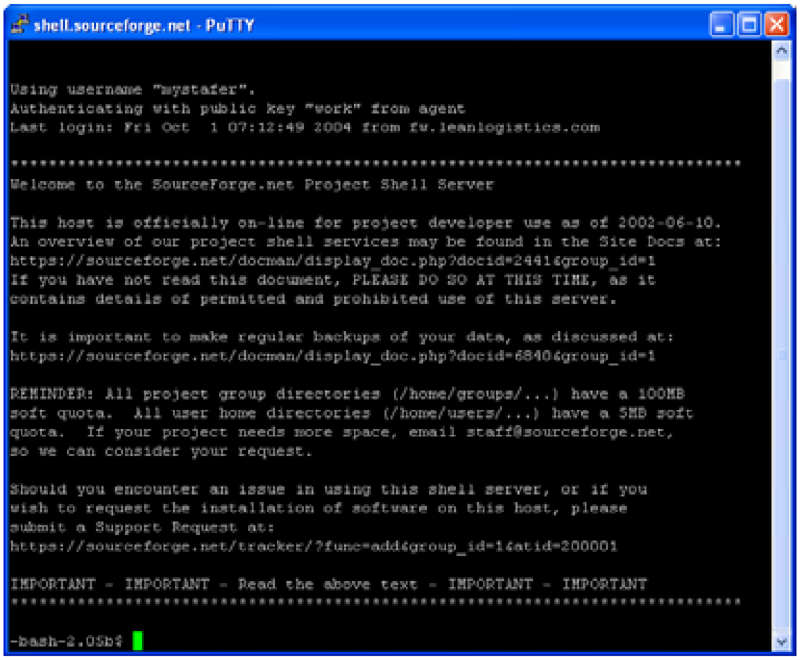
○ Always ○ Never ● Only on clean exit

8. Return the ‘Session’ in the category tree
9. Click the ‘Save’ button to save the session that was entered
10. Click the ‘Open’ button to connect to the sourceforge server



The server's host key is not cached in the registry. You
have no guarantee that the server is the computer you
think it is.
The server's key fingerprint is:
ssh-dss 1024 4c:68:03:d4:5c:58:a6:1d:9d:17:13:24:14:48:ba:99
If you trust this host, hit Yes to add the key to
PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without
adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the
connection.

11. When prompted about the server's key fingerprint, click the Yes button to trust the connection.



```
Using username "mystafer".
Authenticating with public key "work" from agent
Last login: Fri Oct  1 07:12:49 2004 from fw.leanlogistics.com

*****************************************************************************
Welcome to the SourceForge.net Project Shell Server

This host is officially on-line for project developer use as of 2002-06-10.
An overview of our project shell services may be found in the Site Docs at:
https://sourceforge.net/docman/display_doc.php?docid=2441&group_id=1
If you have not read this document, PLEASE DO SO AT THIS TIME, as it
contains details of permitted and prohibited use of this server.

It is important to make regular backups of your data, as discussed at:
https://sourceforge.net/docman/display_doc.php?docid=6840&group_id=1

REMINDER: All project group directories (/home/groups/...) have a 100MB
soft quota.  All user home directories (/home/users/...) have a 5MB soft
quota.  If your project needs more space, email staff@sourceforge.net,
so we can consider your request.

Should you encounter an issue in using this shell server, or if you
wish to request the installation of software on this host, please
submit a Support Request at:
https://sourceforge.net/tracker/?func=add&group_id=1&atid=200001

IMPORTANT - IMPORTANT - Read the above text - IMPORTANT - IMPORTANT
*****************************************************************************

-bash-2.05b$ 
```

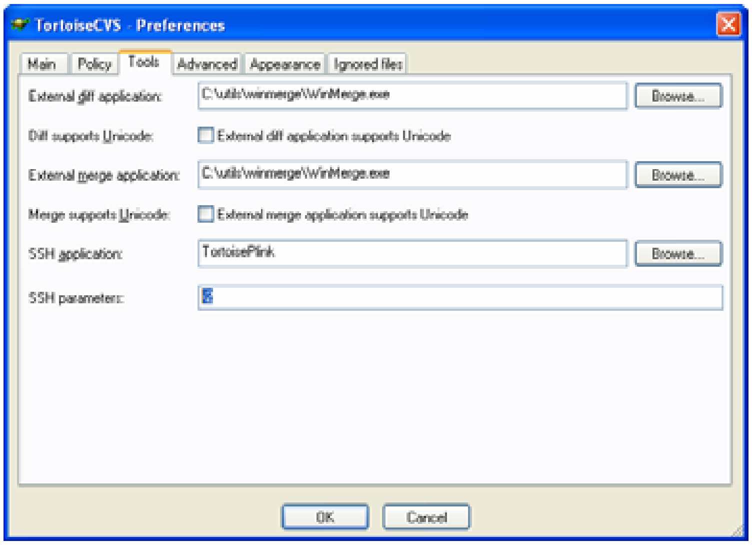
12. If all is well, you should see a screen similar to the following without being prompted for a password for access.

If you are prompted for a password and pageant is running with your private key, it may be possible that your public key has not yet been updated on the shell server. You can verify this by running a cat command on the .ssh/authorized_keys. If the key you generated earlier is in this file then the server has been updated, otherwise you may need to wait longer and try reconnecting for the test to be successful.

Connecting to JOpt CVS

First install the TortoiseCVS program that was downloaded. This will require a reboot of the system after install ☹

Once you have rebooted, you will need to launch the TortoiseCVS preferences application



External diff application: C:\utils\winmerge\WinMerge.exe

Diff supports Unicode: □ External diff application supports Unicode

External merge application: C:\utils\winmerge\WinMerge.exe

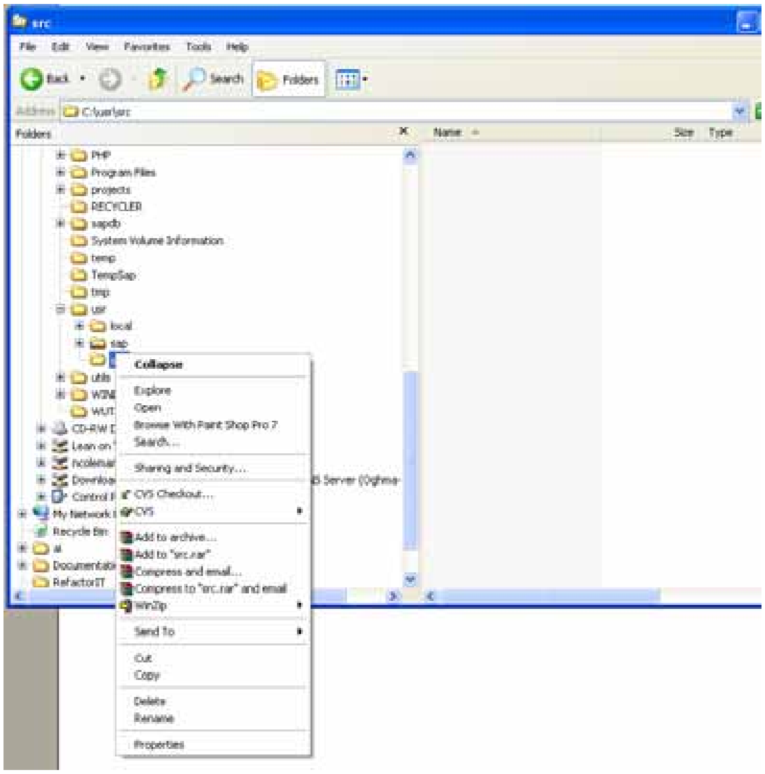
Merge supports Unicode: □ External merge application supports Unicode

SSH application: TortoisePlink

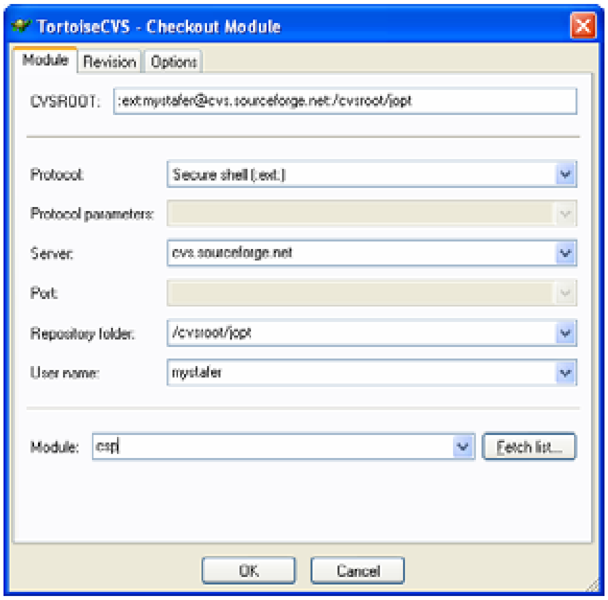
SSH parameters: -2

Enter '-2' in the SSH parameters text box to tell tortoise to use SSH2 protocol when using a secure shell for communication. Click ok to close the window.

You now need to open an explorer window.



Navigate to a directory you wish to contain the source code structure, right click on the directory and select the ‘CVS Checkout…’ command.



CVSROOT: :ext:mystafer@cvs.sourceforge.net:/cvsroot/jopt

Protocol: Secure shell (:ext:)

Protocol parameters:

Server: cvs.sourceforge.net

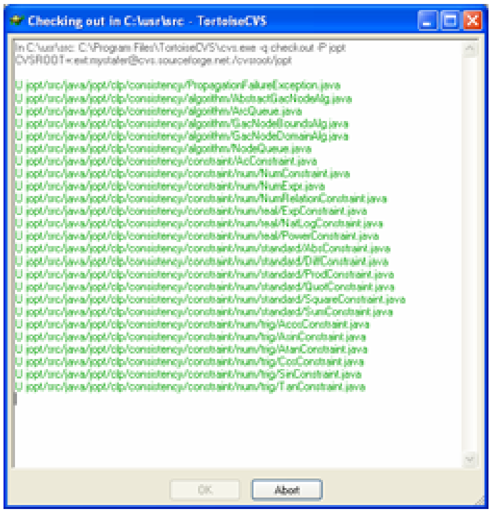
Port:

Repository folder: /cvsroot/jopt

User name: mystafer

Module: csp

- Select ‘Secure shell (:ext:)’ as the protocol
- Enter cvs.sourceforge.net as the server
- Enter /cvsroot/jopt for the repository folder
- Enter your SF username in the user name box
- Enter csp in the Module box
- Click the ok button



```
In C:\usr\src: C:\Program Files\TortoiseCVS\cvs.exe -q checkout -P jopt
CVSROOT=:ext:mystafer@cvs.sourceforge.net:/cvsroot/jopt

U jopt/src/java/jopt/clp/consistency/PropagationFailureException.java
U jopt/src/java/jopt/clp/consistency/algorithm/AbstractGacNodeAlg.java
U jopt/src/java/jopt/clp/consistency/algorithm/ArcQueue.java
U jopt/src/java/jopt/clp/consistency/algorithm/GacNodeBoundsAlg.java
U jopt/src/java/jopt/clp/consistency/algorithm/GacNodeDomainAlg.java
U jopt/src/java/jopt/clp/consistency/algorithm/NodeQueue.java
U jopt/src/java/jopt/clp/consistency/constraint/AcConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/NumConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/NumExpr.java
U jopt/src/java/jopt/clp/consistency/constraint/num/NumRelationConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/real/ExpConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/real/NatLogConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/real/PowerConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/standard/AbsConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/standard/DiffConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/standard/ProdConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/standard/QuotConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/standard/SquareConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/standard/SumConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/trig/AcosConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/trig/AsinConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/trig/AtanConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/trig/CosConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/trig/SinConstraint.java
U jopt/src/java/jopt/clp/consistency/constraint/num/trig/TanConstraint.java
```

If you see a screen similar to this and are not prompted for a password you have successfully connected to CVS.

CVS Basics

The following basic commands are used when working with CVS:

1. update – used to get the latest code, any changes will be merged with files that are updated locally
2. add – used to add files to the cvs repository that are not already in the version system
3. remove – used to delete file from the cvs repository that are not already in the version system
4. commit – used to write modifications, additions and removals from your local file system to the cvs repository

Please add meaningful comments to all items that are committed.

Note: you cannot rename or move files once added to CVS. You will need to remove existing files and add the new ones.

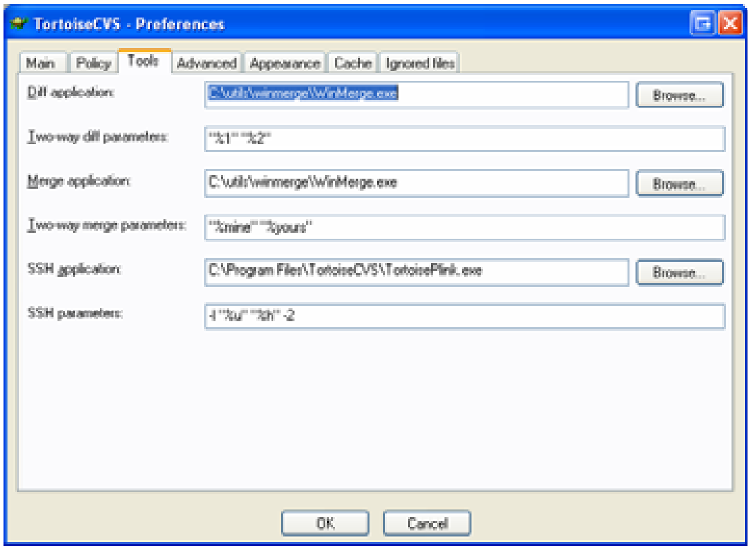
Rolling back changes to a previous version can be done by right clicking on a file and selecting 'History…' from the CVS submenu. You can right click on any previous version and save that over the current version before committing.

You can also click 'Weblog…' option from the CVS submenu for a file to see a list of changes and commits made for each file within your web browser.

CVS with Eclipse

1. From the main menu, select Window -> Preferences
2. In nav tree, select Team -> CVS -> Ext Connection Method
3. Select the ‘Use another connection method type to connect’ option
4. Select ‘extssh’ in the drop down
5. In the nav tree, select Team -> CVS -> SSH2 Connection Method
6. Eclipse cannot use putty keys so we will create a new one
    a. Click the key management tab
    b. Click ‘Generate DSA Key…’
    c. Type a comment in the box like ‘work – eclipse’
    d. Copy the generated public key text into your SF account page
    e. Save the private key using the ‘Save private key…’ button
    f. Click the general tab
    g. Add the private key file that was generated by clicking the ‘Add Private Key…’ button
    h. Close this dialog
7. From the main menu, select Window -> Show View -> Other…
8. Select CVS -> CVS Repositories and click ok
9. Right click in the CVS Repositories window and select New -> Repositories Location…
10. The following settings will need to be entered:
    a. Host: cvs.sourceforge.net
    b. Repository path: /cvsroot/jopt
    c. User: your sf username
    d. Password: your sf password
    e. Connection type: ssh (use default port)
    f. Check ‘Save password’ (this doesn’t use putty’s pageant)
11. Right click on Package Explorer and select New Project…
12. Select Java Project
13. Enter JOpt API as project name
14. Select ‘Create project at external location’ and navigate to the api directory located under directory where project was checked out
15. Click ‘Next’ button
16. You will need to add the commons-logging.jar, log4j.jar, junit.jar and resource directory to the build path
17. Click ‘Finish’ button
18. Right click on JOpt project in Packager, select Team -> Share Project…
19. A dialog should display indicating CVS directories already exist and you only need to click Finish
20. Repeat this process for SPI, Example and Test
21. SPI will depend on the API project in the project build path
22. Example and Test will depend on both the API and SPI projects
23. The workspace should now be connected to CVS and ready for development

TortoiseCVS 1.8.1



Diff application: C:\utils\winmerge\WinMerge.exe

Two-way diff parameters: "%1" "%2"

Merge application: C:\utils\winmerge\WinMerge.exe

Two-way merge parameters: "%mine" "%yours"

SSH application: C:\Program Files\TortoiseCVS\TortoisePlink.exe

SSH parameters: -l "%u" "%h" -2

The SSH parameters for version 1.8.1 are different than previous versions. You will need to add -2 to the existing parameters. The parameters should look like this `–l “%u” “%h” -2`