
XOOPS CVS Access Guide

Robert Kraske

XOOPS CVS Access Guide

Robert Kraske

Published 2005-02-14
Copyright © 2004,2005

This document describes working with CVS. The intended audience is the user or developer who has had no prior experience with CVS or with how CVS is used by XOOPS developers.

Released under the Creative Commons license. See Creative Commons [<http://creativecommons.org/>]

Table of Contents

Introduction	vii
1. Browsing the CVS online	1
2. CVS clients	3
2.1. TortoiseCVS	3
2.2. WinCVS	3
2.3. WinMerge (for code comparison)	4
2.3.1. TortoiseCVS and WinMerge	5
2.3.2. WinCVS and WinMerge	6
3. Anonymous CVS Access	9
3.1. Directory structure of XOOPS	9
3.2. Checking out a module	10
3.2.1. Some variations on checking out	12
4. XOOPS Module Developers	13
4.1. SourceForge.net account	13
4.2. Account on dev.xoops.org	13
4.3. Authorization to use XOOPS CVS Repository	14
4.3.1. Start a new project	14
4.3.2. Apply For CVS	17
4.3.3. Help an existing project	17
5. SourceForge.net	19
5.1. Setting up SSH	19
5.1.1. SSH key generation	19
5.1.2. Uploading SSH key	22
5.1.3. SSH authentication agent	23
5.2. Checking out existing module	25
5.2.1. Some variations on checking out	28
5.3. Adding new module	28
6. CVS Terminology	31
7. CVS resources	33
Index	35



List of Figures

1.1. Project CVS link screenshot	1
1.2. Accessing online CVS screenshot	2
2.1. TortoiseCVS screenshot with mascot	3
2.2. WinCVS screenshot	4
2.3. WinMerge screenshot	5
2.4. Open TortoiseCVS preferences	5
2.5. Set up WinMerge as the external diff and merge program	6
2.6. Open TortoiseCVS preferences	6
2.7. Set up WinMerge as the external diff program	7
3.1. The directory structure on SourceForge.net	10
3.2. Check out menu	11
3.3. Check out dialog	11
3.4. Finished check with success	12
4.1. The registration form on SourceForge.net.	13
4.2. You need to register on dev.xoops.org first	14
4.3. Start a new project on dev.xoops.org	15
4.4. Overview of the Start A New Project process	15
4.5. Services and Requirements	16
4.6. Terms of Service Agreement	17
4.7. Project Information	17
4.8. Apply For CVS	17
5.1. Start puttygen.exe	19
5.2. Adjust puttygen to the correct settings	20
5.3. Apply a passphrase	21
5.4. Save the private key	22
5.5. Go to the page where you can submit you public key	23
5.6. Start the SSH agent pageant	24
5.7. The pageant icon in the system tray	24
5.8. To add your private key to the SSH agent	24
5.9. Select the previously stored private key.	25
5.10. Type in the passphrase.	25
5.11. Check out menu	26
5.12. Check out menu	27
5.13. Finished check with success	28
5.14. The Make New Module... menu entry	29
5.15. The Make New Module dialog	30



Introduction

CVS stands for Concurrent Versions System. It is a system which has two main purposes:

- 1. to provide a history of changes made to files, including what was changed, when it was changed, and who made the change and
- 2. to coordinate changes made to files in multiple developer situations, so that two developers can work on the same file safely, knowing that their changes will be merged, not overwritten.



Chapter 1. Browsing the CVS online

You don't need to be a developer to see the latest changes to files of a XOOPS project. If the developer uses the dev.xoops.org site to host their XOOPS module project then you can easily browse from the project's online CVS link to see which files have been changed recently.



Warning

Remember, you are looking at files that are being developed. This means that there is no guarantee that the files are in a working state. If you intend to use the module on a live site then always use the officially released files, not files from CVS.

To browse the online CVS follow these simple steps:

1. Visit the project's page at <http://dev.xoops.org>.
2. Find the link to 'CVS' at the top of the project's page, as seen in Figure 1.1, "Project CVS link screenshot" [?]



Figure 1.1. Project CVS link screenshot

3. Click on the 'Browse CVS Repository' link on the right side of the page, as seen in Figure 1.2, "Accessing online CVS screenshot" [2]

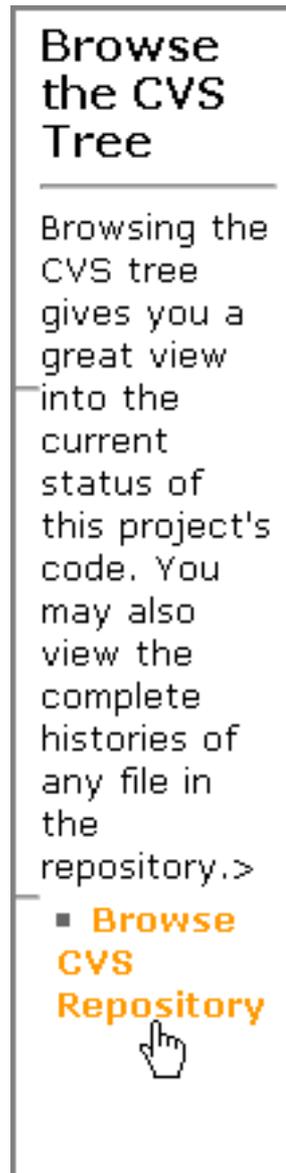


Figure 1.2. Accessing online CVS screenshot

4. That's it!

Looking at single files in the online CVS is interesting, but what if you just want to test the latest version of the module? How can you get all the files at once?

That is where CVS clients enter the picture. The following chapters give an overview of the most popular CVS clients, and how to use them to access a XOOPS project CVS repository.

Chapter 2. CVS clients

To work with CVS (in our case, with the **XOOPS** repository) we would recommend to install TortoiseCVS and maybe WinCVS. These two programs are GUI front-ends, that makes working with CVS much easier than using the CVS command line tools (That's something for the experts).

These two programs are for Windows only. People who are interested in GUI front-ends for other operating systems should take a look at the CvsGui homepage [<http://www.wincvs.org/>].

2.1. TortoiseCVS

TortoiseCVS lets you work with files under CVS version control directly from Windows Explorer. That means, it is perfectly usable for the day to day work. You can download TortoiseCVS from the TortoiseCVS homepage [<http://www.tortoise cvs.org/>].

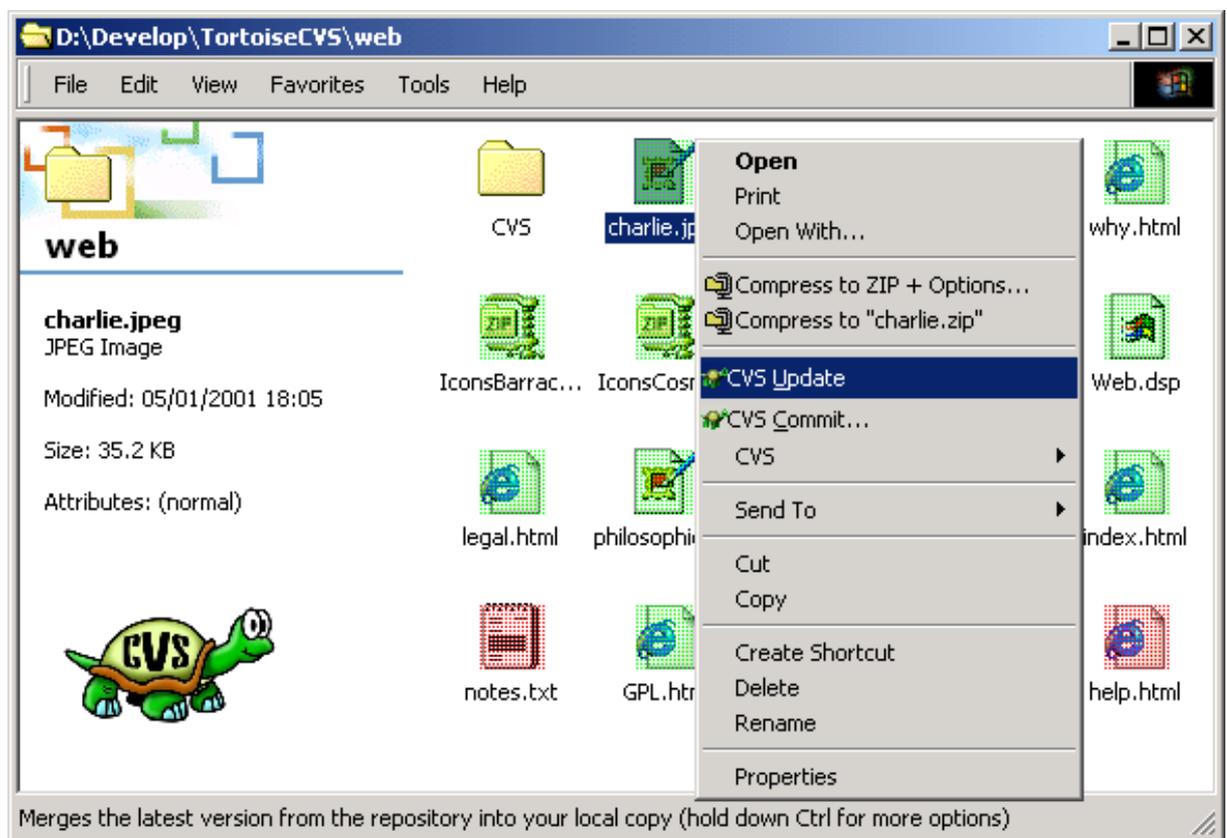


Figure 2.1. TortoiseCVS screenshot with mascot

2.2. WinCVS

For a more comprehensive working environment with CVS you can use WinCVS, a more complex GUI front-end. You can download it from SourceForge.net [http://sourceforge.net/project/showfiles.php?group_id=10072]. The latest stable release is WinCVS 1.3.17.

If you want to use the macro capabilities of WinCVS you also need to download Python. The Python version you need depends on the version of WinCVS you want to use. At the time of writing, the WinCVS 1.3.21.2 works with Python 2.3.4. You can download Python from the Python download page [<http://www.python.org/download/>]

After you have installed WinCVS you need to set up the CVS_RSH environment variable to the plink.exe of

PuTTY, so you can work with SourceForge.net. E.g., if you have PuTTY installed into C:\program files\PuTTY you have to set CVS_RSH to C:\program files\PuTTY\plink.exe

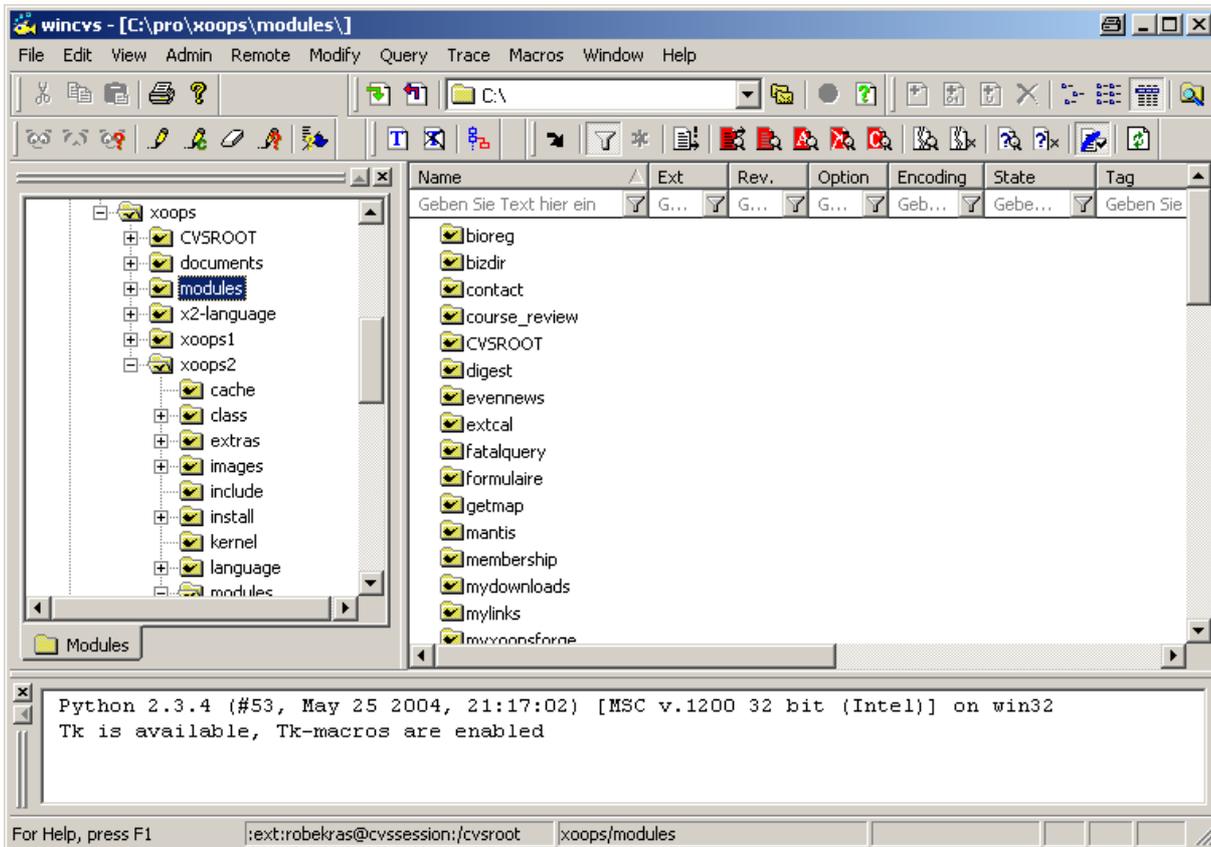


Figure 2.2. WinCVS screenshot

2.3. WinMerge (for code comparison)

Another program which is recommended if you work with CVS is a tool for showing the differences between the versions of a file.

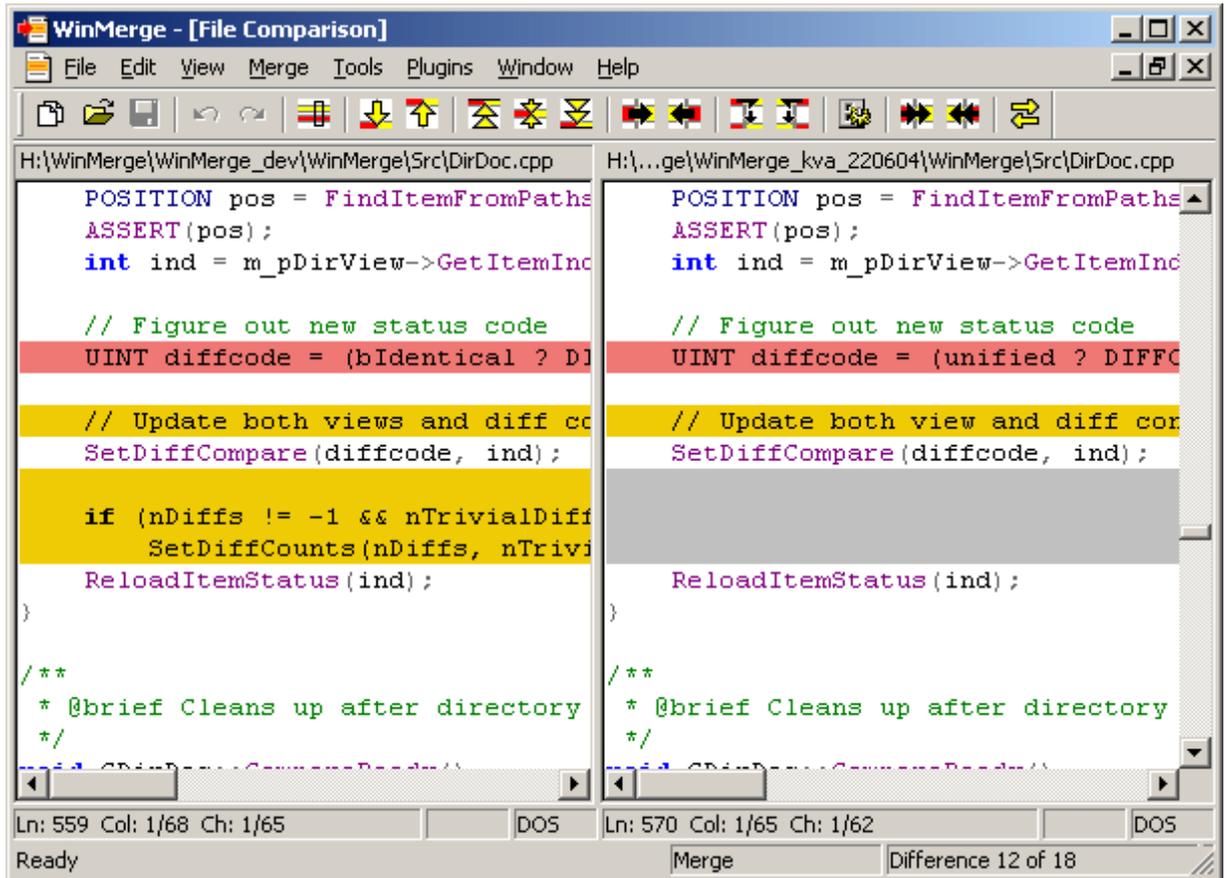


Figure 2.3. WinMerge screenshot

You can set up TortoiseCVS and WinCVS so they use WinMerge as an external diff program.

2.3.1. TortoiseCVS and WinMerge

Right click anywhere in an explorer window and go to the CVS entry. This opens the TortoiseCVS menu, where you can point to the Preferences... entry. Within the preferences dialog click on the [Tools] tab and make the appropriate settings as shown in Figure 2.5, "Set up WinMerge as the external diff and merge program" [6].

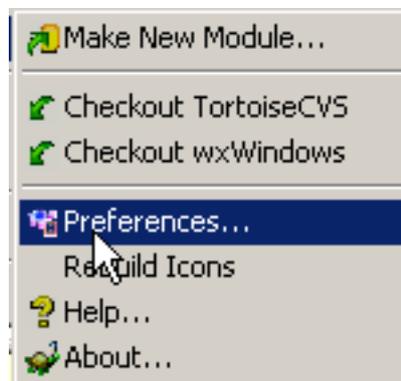


Figure 2.4. Open TortoiseCVS preferences

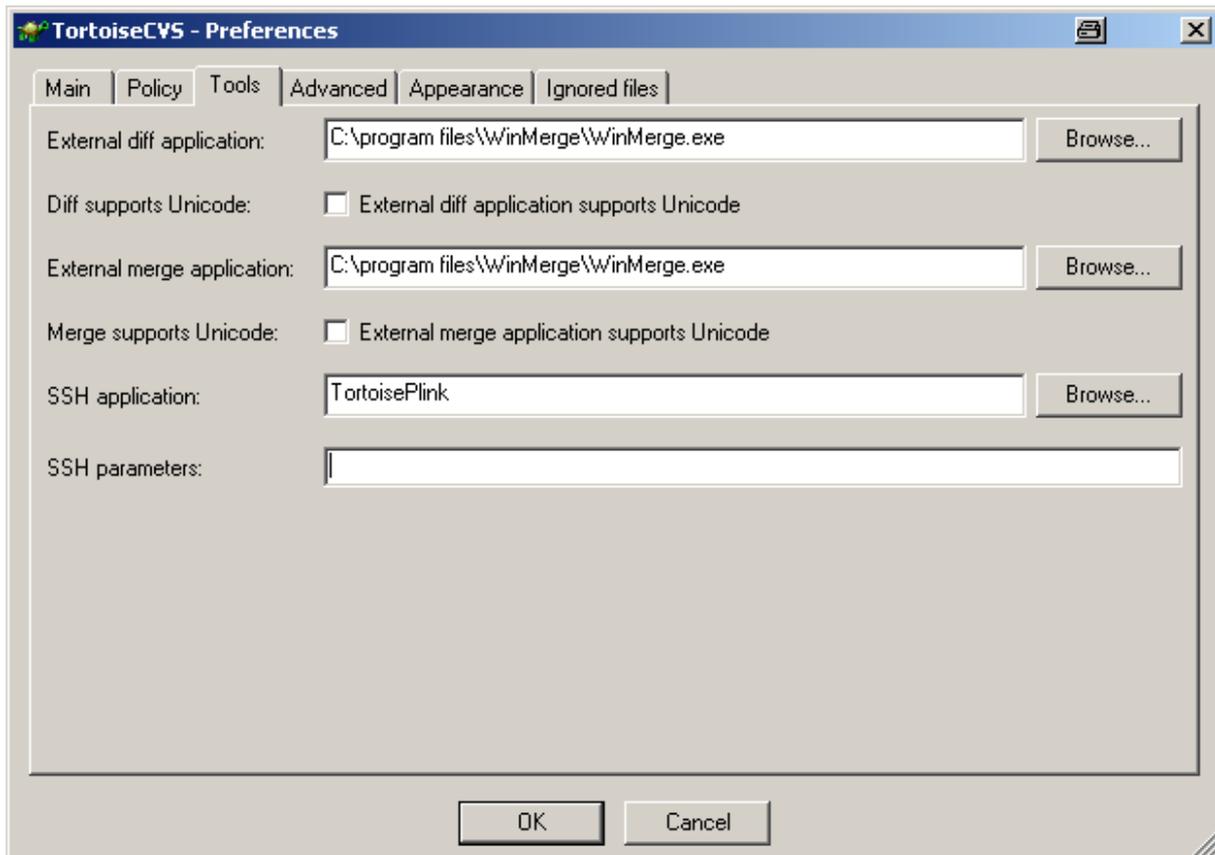


Figure 2.5. Set up WinMerge as the external diff and merge program

2.3.2. WinCVS and WinMerge

Open WinCVS, go to the Admin menu and open the Preferences dialog. Click on the [WinCvs] tab and make the appropriate settings as shown in Figure 2.7, “Set up WinMerge as the external diff program” [7].

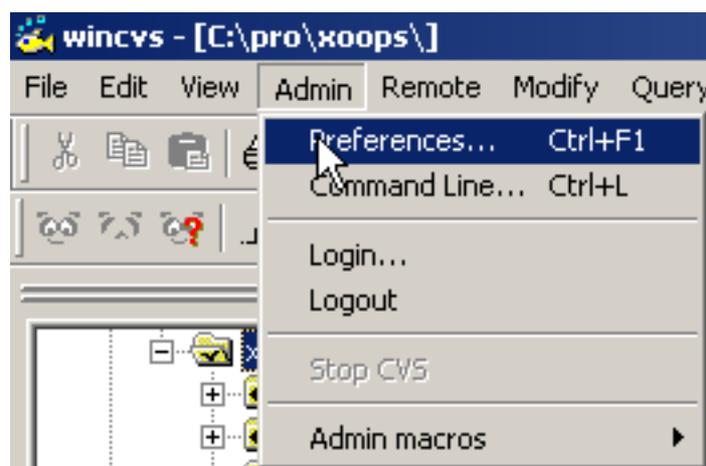


Figure 2.6. Open TortoiseCVS preferences

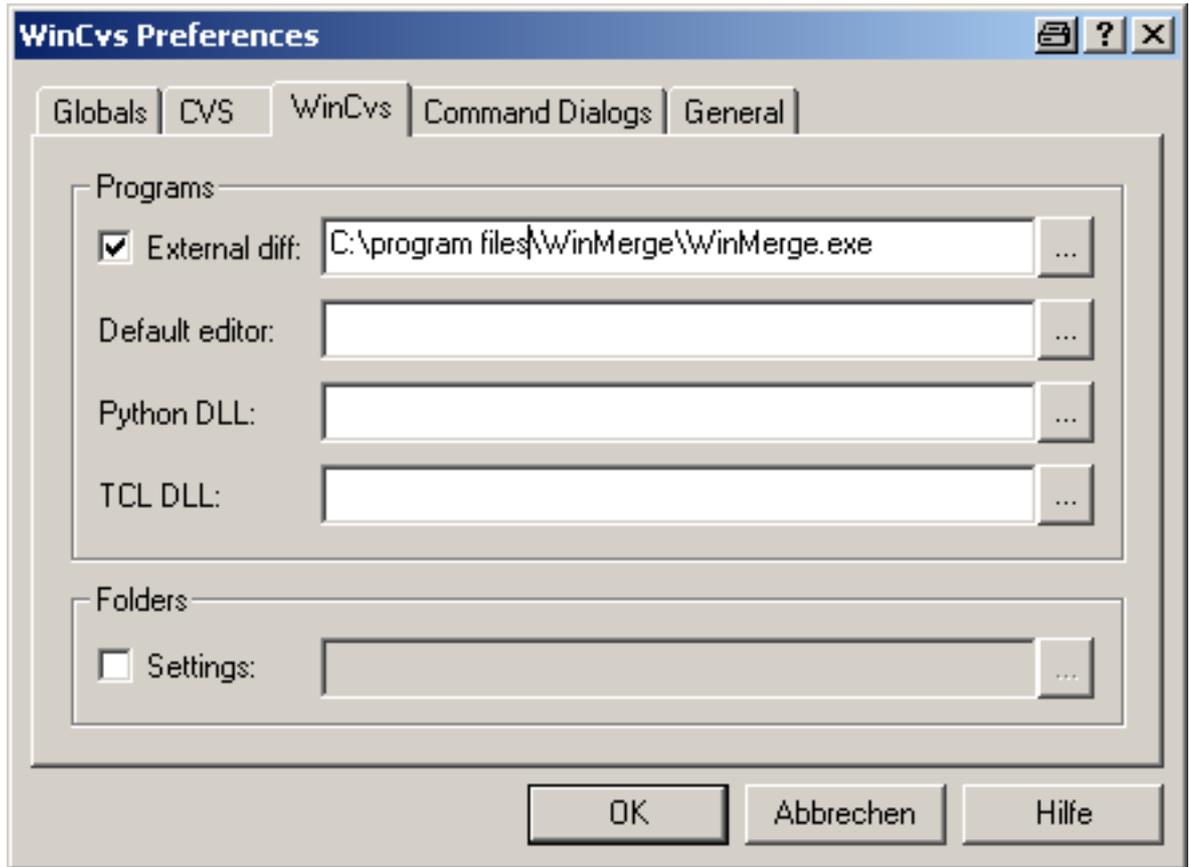


Figure 2.7. Set up WinMerge as the external diff program



Chapter 3. Anonymous CVS Access

This section of the guide shows how to access the CVS repository of a project using anonymous access and the TortoiseCVS client. If you are a developer of the XOOPS project then you can skip this section and go straight to the Developer CVS Access section.

Before we look at setting up the TortoiseCVS client, we should understand how the directories are set up at the SourceForge.net CVS repository. XOOPS is only one of many open source projects hosted at that site. This explains why the root folder of the CVS is not xoops.

3.1. Directory structure of XOOPS

The XOOPS project has the following top level CVS modules:

- **documents** [<http://cvs.sourceforge.net/viewcvs.py/xoops/documents/>]: All the XOOPS documentation sourcefiles (xml files, images, build file etc.)
- **modules** [<http://cvs.sourceforge.net/viewcvs.py/xoops/modules/>]: All the xoops modules are kept within this CVS module.



Important

All module names are lower case. All XOOPS modules are to be set up within this modules directory. Any misplaced files will be removed (we will try to warn you first).

- **x2-language** [<http://cvs.sourceforge.net/viewcvs.py/xoops/x2-language/>]: XOOPS 2.x translations
- **xoops1** [<http://cvs.sourceforge.net/viewcvs.py/xoops/xoops1/>]: XOOPS 1.x files
- **xoops2** [<http://cvs.sourceforge.net/viewcvs.py/xoops/xoops2/>]: XOOPS 2.x files

The directory structure on SourceForge.net looks like the following image shows.

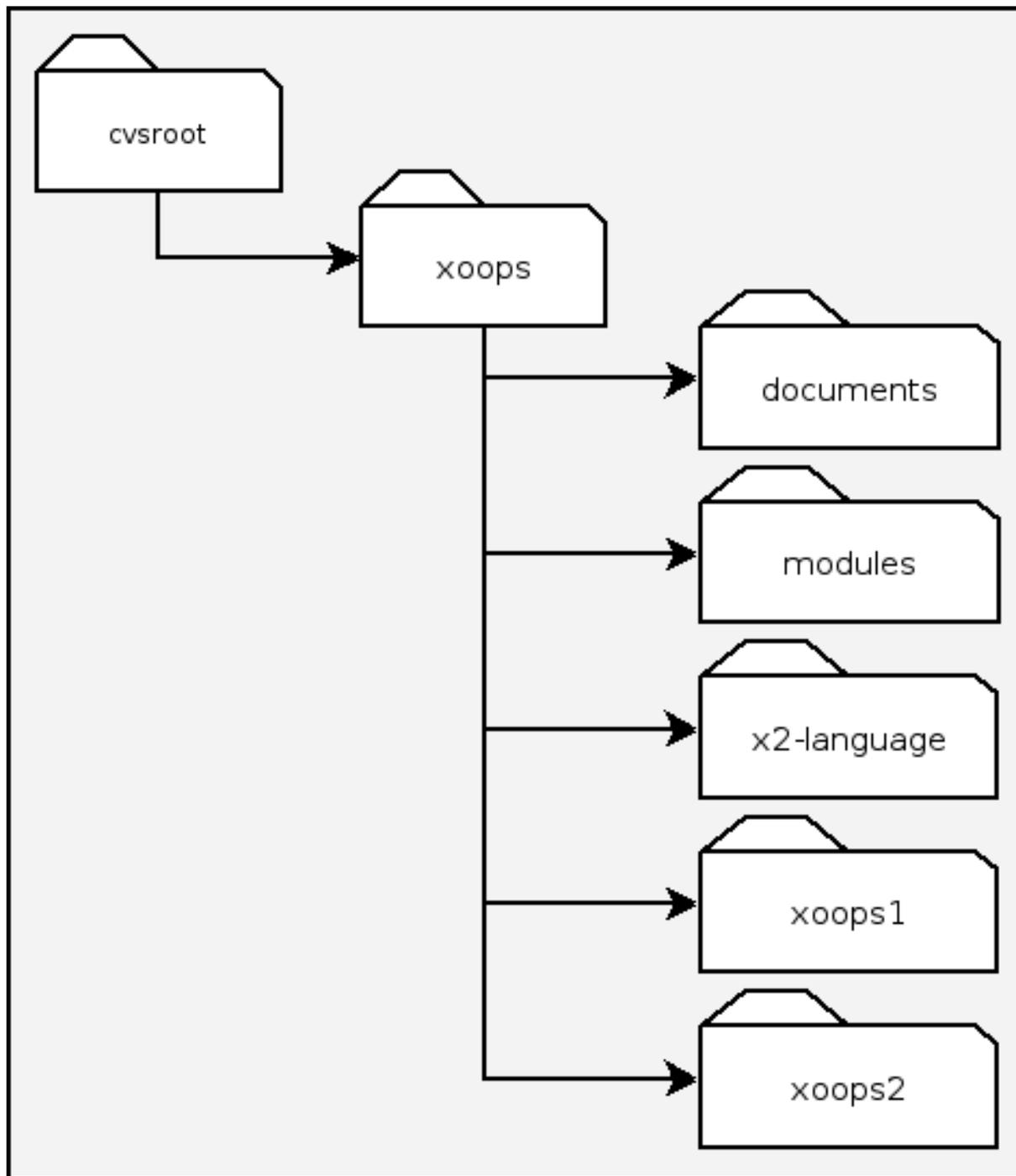


Figure 3.1. The directory structure on SourceForge.net

3.2. Checking out a module

When you want to acquire all the files of a project for the first time it is called 'check out'. To check out the files of a project do the following:

- Within the explorer window open the directory where you want to have the module files, and right click into it. This opens the context menu. Click on [CVS Checkout...] (see Figure 3.2, "Check out menu" [11]).

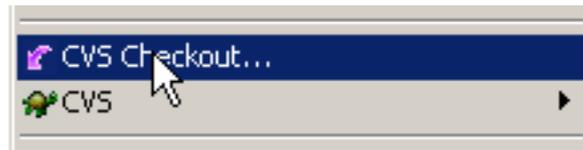


Figure 3.2. Check out menu

- This open the “Checkout Module” dialog. Make the appropriate settings:
 - [Protocol:] Select *Password Server (:pserver:)*
 - [Server:] type in *cvs.sourceforge.net*
 - [Repository folder:] type in */cvsroot/xoops*
 - [User name:] type in *anonymous*
 - [Module:] type in the name of the module you want to have on your local drive, but don't forget the preceding *modules/*. E.g. type *modules/theModuleYouWant*.

Figure 3.3, “Check out dialog” [11] shows you a screenshot of how it would look like.

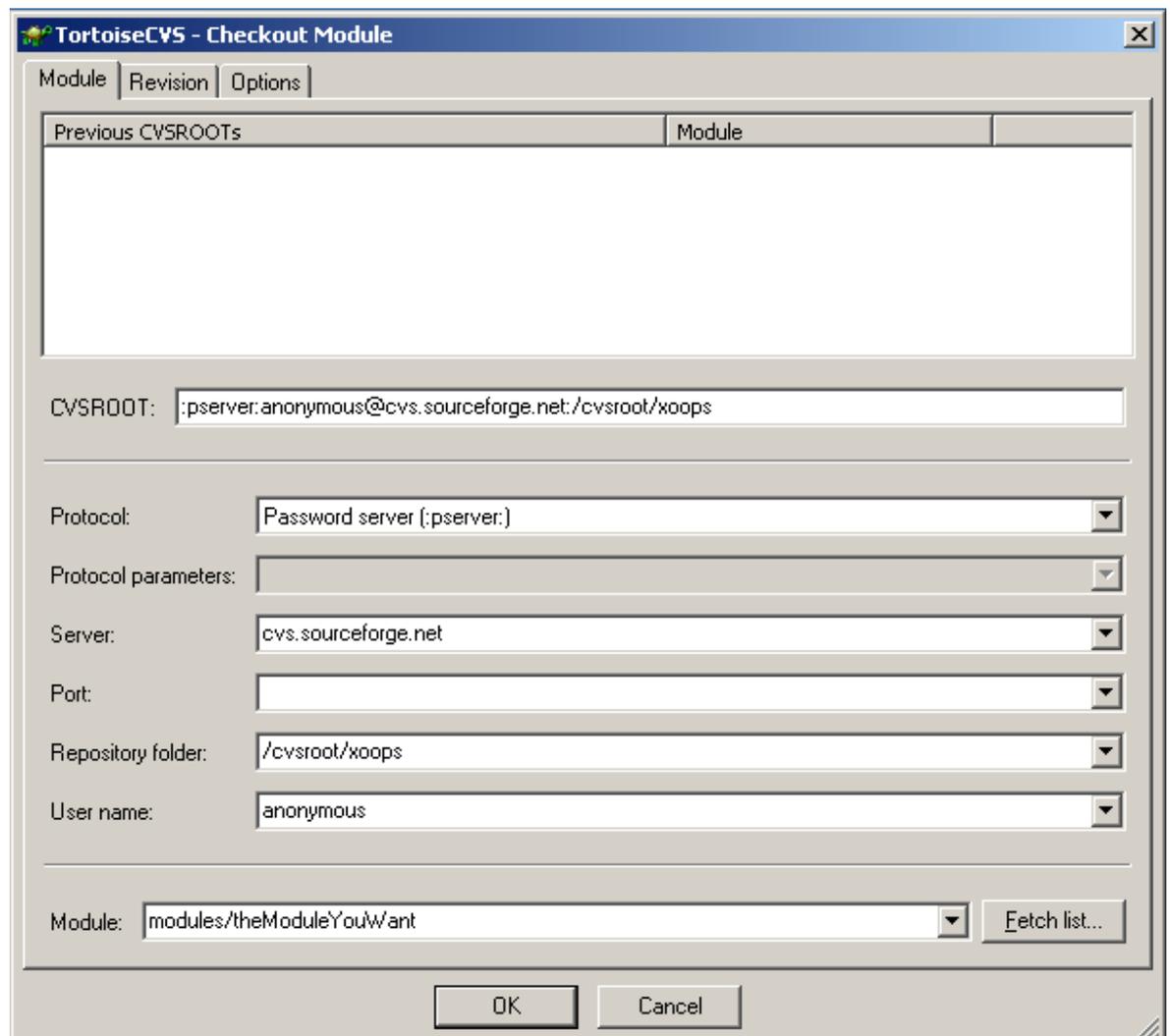


Figure 3.3. Check out dialog

Click on [OK] if everything is prepared.

- In the case everything works well, you will see the “Finished checkout” window with a success message. Figure 3.4, “Finished check with success” [12]) shows the success message after checking out my xdocman module.

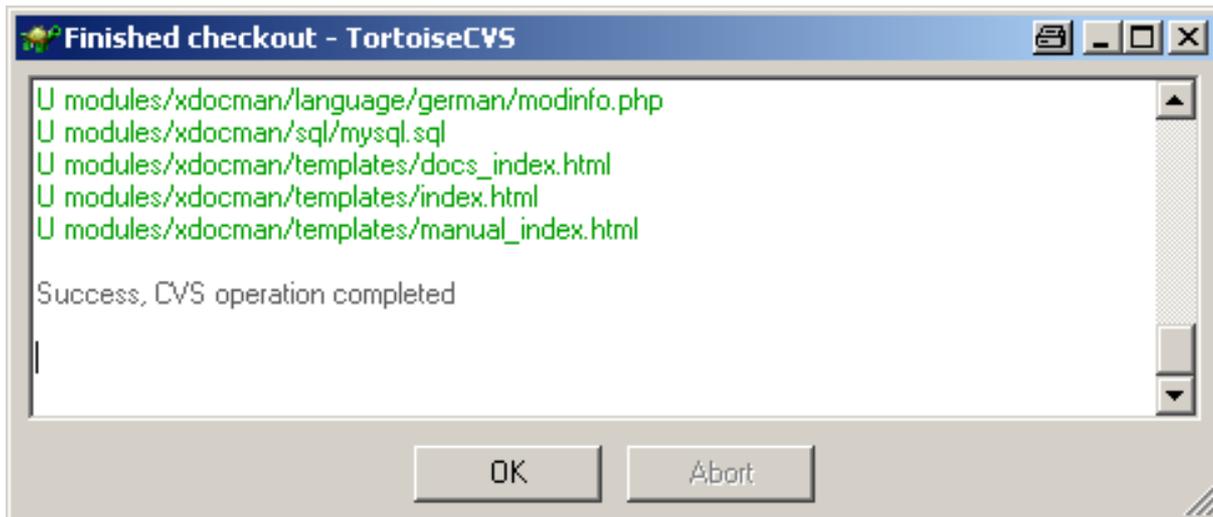


Figure 3.4. Finished check with success

3.2.1. Some variations on checking out

- *Checking out an empty existing module* By default, TortoiseCVS prunes empty directories after the checkout. If a module is empty, then it'll be deleted immediately after it is checked out. Go to CVS -> Preferences -> Policy and uncheck the [Prune empty folders checkbox] if you want to change this behaviour.
- *Checking out on a branch* On CVS-submenu click on [Revision Graph...] right click on a version and click on [Tag...] or [Branch...], then click on [Update list...] and select the tag from the list.
- *Checking out XOOPS core or other modules* You can use these same settings to check out any number of modules or the XOOPS core files. Change the Module setting to xoops2 for the core or modules to checkout all the modules. Checking out all the modules at once could take a very long time though, as there are a great number of XOOPS modules!

Chapter 4. XOOPS Module Developers

XOOPS module developers are able to upload their module files to a CVS server, the same SourceForge.net server that the XOOPS core uses.

In order to make use of this service there are a few things you need to get started:

- A SourceForge account (see Section 4.1, “SourceForge.net account” [13])
- An account at the XOOPS development site (see Section 4.2, “Account on dev.xoops.org” [13])
- Authorization to use CVS (see Section 4.3, “Authorization to use XOOPS CVS Repository” [14])
- A CVS client (see Section 4.3.3, “Help an existing project” [17])

This section of the guide will provide the appropriate information to guide you through the necessary processes.

4.1. SourceForge.net account

Go to the SourceForge User Registration Page [http://sourceforge.net/account/newuser_emailverify.php] if you don't already have a SourceForge.net account. See Figure 4.1, “The registration form on SourceForge.net.” [13]

You are now starting the process to register an account on SourceForge.net. Accounts are provided free-of-charge. Before registering an account, you should review our [Terms of Use Agreement](#) and [Privacy Policy](#).

At this time, we will request two pieces of information from you:

First, you will be asked to pick a password for use with SourceForge.net. This should be a unique password consisting of at least six letters and numbers.

Second, you will need to tell us your email address. This should be an existing email address that you check regularly. We will be sending email to this address as part of the registration process.

Please fill in all provided fields. If we find a problem with your responses, we will let you know and give you a chance to correct your answers.

Password: Enter a unique password containing only letters and numbers. Minimum length is 6 characters. Password will not be shown on screen. Do not use a password already used for another account.

Enter the password once:

As confirmation, enter the same password again:

Email Address: Enter your private email address. SourceForge.net will send an email containing a verification URL to this address. This must be an address that you already have, which you regularly check for new email. After account registration, we will use this address to contact you about account problems, or if you lose your password.

Enter your email address:

Figure 4.1. The registration form on SourceForge.net.

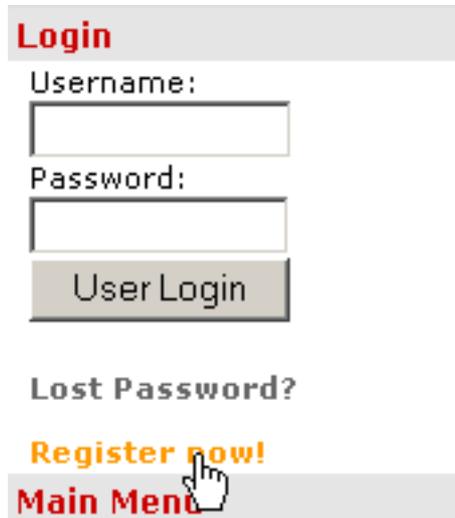
4.2. Account on dev.xoops.org

An account on dev.xoops.org? Very easy, I think you already know how, but for the sake of completeness...

Just click on [**Register now!**] (see Figure 4.2, “You need to register on dev.xoops.org first” [14]), fill in the requested data into the “User Registration” form and click on [**Submit**] , or just go directly to the User Registra-

tion form [<http://dev.xoops.org/register.php>].

Soon you should get an email with a link which is necessary to make up the registration process. After that, you have an account on dev.xoops.org, but you are not yet allowed to work on the **XOOPS** CVS repository. This need some more steps.



Login

Username:

Password:

User Login

Lost Password?

Register now!

Main Menu

Figure 4.2. You need to register on dev.xoops.org first

4.3. Authorization to use XOOPS CVS Repository

There are two possibilities to get CVS access to SourceForge.net via **XOOPS**. First you have your own project, or second there is a project where you want to help. If you want to establish you own project, please follow the information within this and the following section, else please read Section 4.3.3, “Help an existing project” [17].

4.3.1. Start a new project

- Login into dev.xoops.org and click on link “Start New Project” (see Figure 4.3, “Start a new project on dev.xoops.org” [15]).
- You will be guided through three info pages:
 - The “Start A New Project” process info page (see Figure 4.4, “Overview of the Start A New Project process” [15])
 - The “Services and Requirements” info page (see Figure 4.5, “Services and Requirements” [16])
 - The “Terms of Service Agreement” info page (see Figure 4.6, “Terms of Service Agreement” [17])
- The last page is the “Project Information” form page (see Figure 4.7, “Project Information” [17]), where you fill in the required information about your project.



Figure 4.3. Start a new project on dev.xoops.org

Start A New Project

The XOOPS Development Forge would like to extend an invitation to XOOPS Module Developers to host your **Open Source** development project(s) for no price and with no catch.

The Process

Registering a XOOPS Module Development project with the XOOPS Development Forge is an easy process, but we do require a reasonable amount of information in order to automate things as much as possible and present your project(s) adequately.

Registering project consists of the three steps:

- Submitting a request
- Approval of the request
- Setting up your project's account

Currently, you are going to proceed with the first step of the process. It should take about 10 minutes. After that, please allow several days for our review of the request. If it complies with our requirements for hosting (see the next step), your project will be approved, and you will receive e-mail with directions for future steps.

During signup, we will present you with some legal documents regarding your account with us. Please do not ignore these; they are very important to you and your project.

Continue

Figure 4.4. Overview of the “Start A New Project” process

Step 1 - Services and Requirements

The XOOPS Development Forge offers a full suite of services for your hosted module development project(s). If you haven't already, please be sure to browse the most recent revision of these services.

Use of Project Account

The space provided to you on this server is given for the expressed purpose of XOOPS Module development. For more information, please read the XOOPS Development Forge Terms of Service ("Terms of Service").

Creative Freedom

It is our intent to allow you creative freedom on your project. For our legal protection and yours, there are limits. Details about these restrictions are described in the Terms of Service.

Advertisements

You may not place any revenue-generating advertisements on the XOOPS Development Forge website.

Open Source/Rights to Code

You will be presented with a choice of Open Source approved licenses for your project. You will still own the code, but these licenses also allow us to make your code available to the general public. Although you may choose to stop hosting your project with us at any time, the nature of these licenses will allow us to continue to make your code available. Details about content ownership are described in the Terms of Service.

If you wish to use another license that is not currently approved by the Open Source Initiative, let us know and we will review these requests on a case-by-case basis.

It is our intent to provide a permanent home for all versions of your code. We do reserve the right, however, to terminate your project if there is due cause, in accordance with the Terms of Service.

Back

Continue

Figure 4.5. Services and Requirements

Step 2 - Terms of Service Agreement

1. ACCEPTANCE OF TERMS

A person's use of XOOPS Development Forge and any of its Services (as defined below) at any time is subject to XOOPS Development Forge's then-current Terms of Service ("TOS"), which may be updated from time to time as set forth below. Each user can review the then-current TOS at any time at:

<http://dev.xoops.org/modules/xfmod/tos/tos.php>. In addition, each user's use of particular Services may be subject to specific guidelines or rules ("Service-specific Rules") posted from time to time and incorporated by this reference into the TOS. A user's use of XOOPS Development Forge and/or its Services constitutes full acceptance of and agreement to the TOS; if a user does not accept our TOS, he or she is not granted rights to use XOOPS Development Forge or any of its Services, as defined herein, and should refrain from accessing XOOPS Development Forge and its Services.

Figure 4.6. Terms of Service Agreement

Step 3 - Project Information

You want to host your XOOPS Module Development project here on the XOOPS Development Forge. Doing this gives you the advantage of being close to where the action is, being able to talk to other XOOPS Module developers, see good examples, and you get to show off your own prowess.

To apply for a new development project, we will need some information about the project. Please fill in this form and we will contact you as soon as possible.

1. Project Full Name

Please name your project. The "Full Name" should be descriptive of your project, and must be from 3 to 60 characters in length.

Project Full Name

2. Project Purpose and Summarization

Figure 4.7. Project Information

4.3.2. Apply For CVS

In addition to the "Start New Project" procedure you have to fill out the "Apply For CVS" which is started by the the appropriate link in the dev.xoops.org "Main Menu" section (see Figure 4.8, "Apply For CVS" [17]).

Fill in the appropriate information, and after some days you will get an email about whether you are enabled for the SourceForge.net CVS access.

Figure 4.8. Apply For CVS

4.3.3. Help an existing project

If you want to contribute to an existing project, then the following steps are required:

- Ask the project admin if you could become a member of his project.
- The project admin will add you to the project via project -> admin -> **[Add User]** .
- You can now run “Apply For CVS” as it is described in Section 4.3.2, “Apply For CVS” [17].

Chapter 5. SourceForge.net

This section of the guide shows how to gain developer access the CVS repository of a project.

For a quick reminder of the directory structure of the CVS repository read Section 3.1, "Directory structure of XOOPS" [9]

Once you have developer authority organized, as outlined in Section 4.3, "Authorization to use XOOPS CVS Repository" [14] and you have a CVS client (see Chapter 2, *CVS clients* [3]) we can continue by installing the necessary tools for SSH.

For those of you who are interested in more detailed information about SSH and SourceForge.net please take a look at: Guide to Generation and Posting of SSH Keys [http://sourceforge.net/docman/display_doc.php?docid=761&group_id=1]. The following is a summary of that document.

5.1. Setting up SSH

To work with SourceForge.net (as a developer) we have to set up a special protocol, called SSH (Secure Shell).

For Windows we can use PuTTY [<http://www.chiark.greenend.org.uk/~sgtatham/putty/>] as a SSH client. You can download it from here [<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>]. There are links for downloading the programs separately, for different platforms, but there is also a link for a Windows-style installer, which I recommend for downloading.

After downloading and installing PuTTY we can go ahead with the generation of a SSH private and public key, which we will need for accessing the CVS repositories on SourceForge.net.

5.1.1. SSH key generation

- Execute puttygen.exe

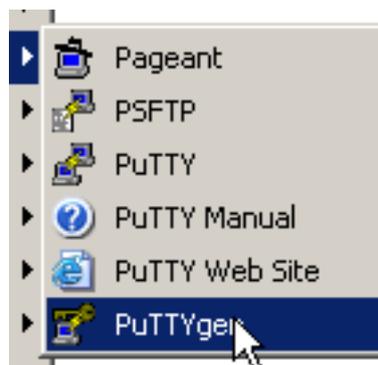


Figure 5.1. Start puttygen.exe

- Select the desired key type ("SSH2 DSA", within the "Parameters" section).

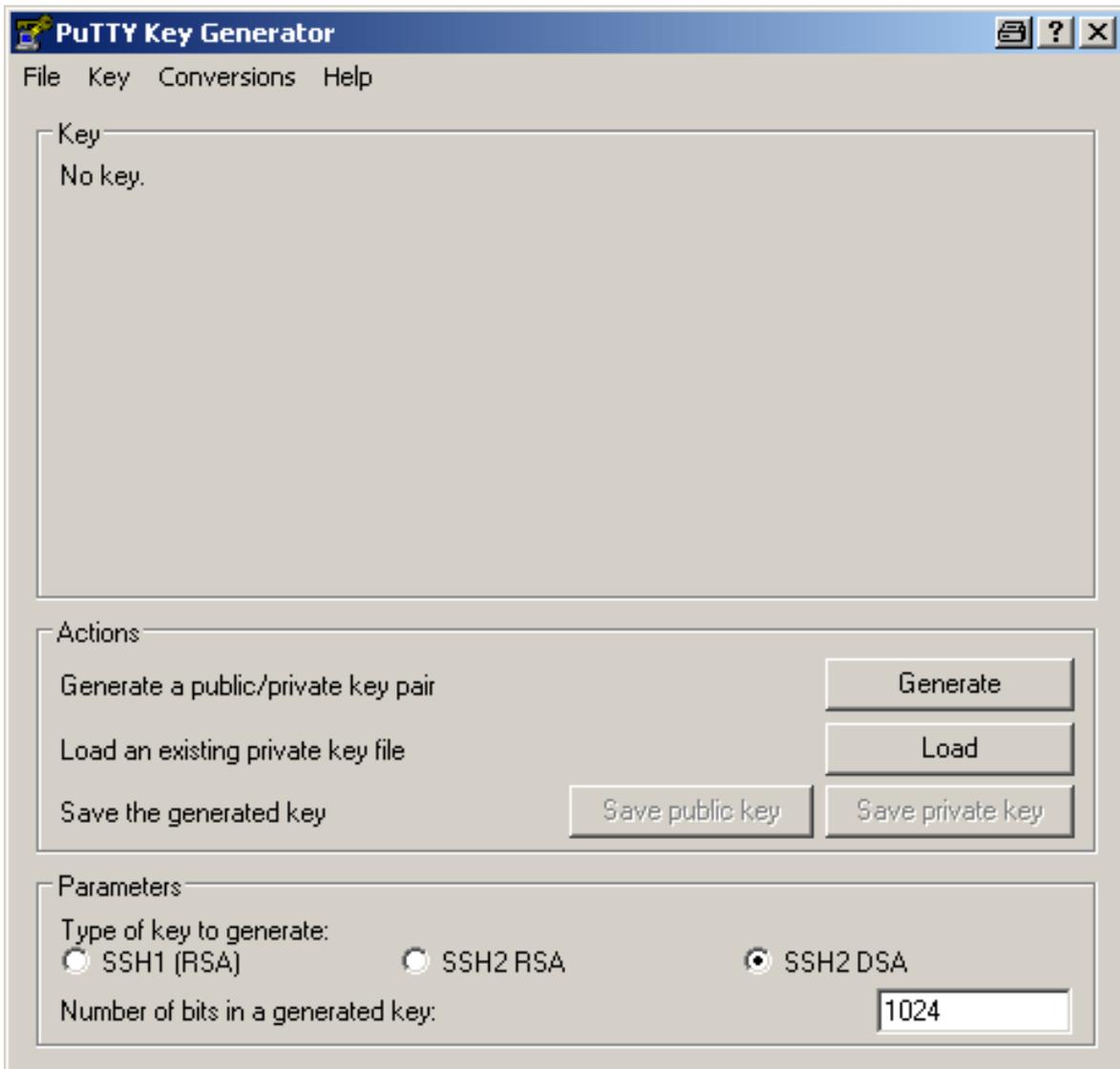


Figure 5.2. Adjust puttygen to the correct settings

- Click on the [Generate] button.
- Follow the on-screen instructions (“Please generate some randomness by moving the mouse over the blank area”). Key generation will be performed immediately afterward.
- Enter `username@shell.sf.net` (or `username@sourceforge.net`) in the [Key comment] field, replacing “username” with your SourceForge.net user name. This comment will help you to identify the purpose of this key.
- Enter the desired passphrase in the [Key passphrase] and [Confirm passphrase] fields.

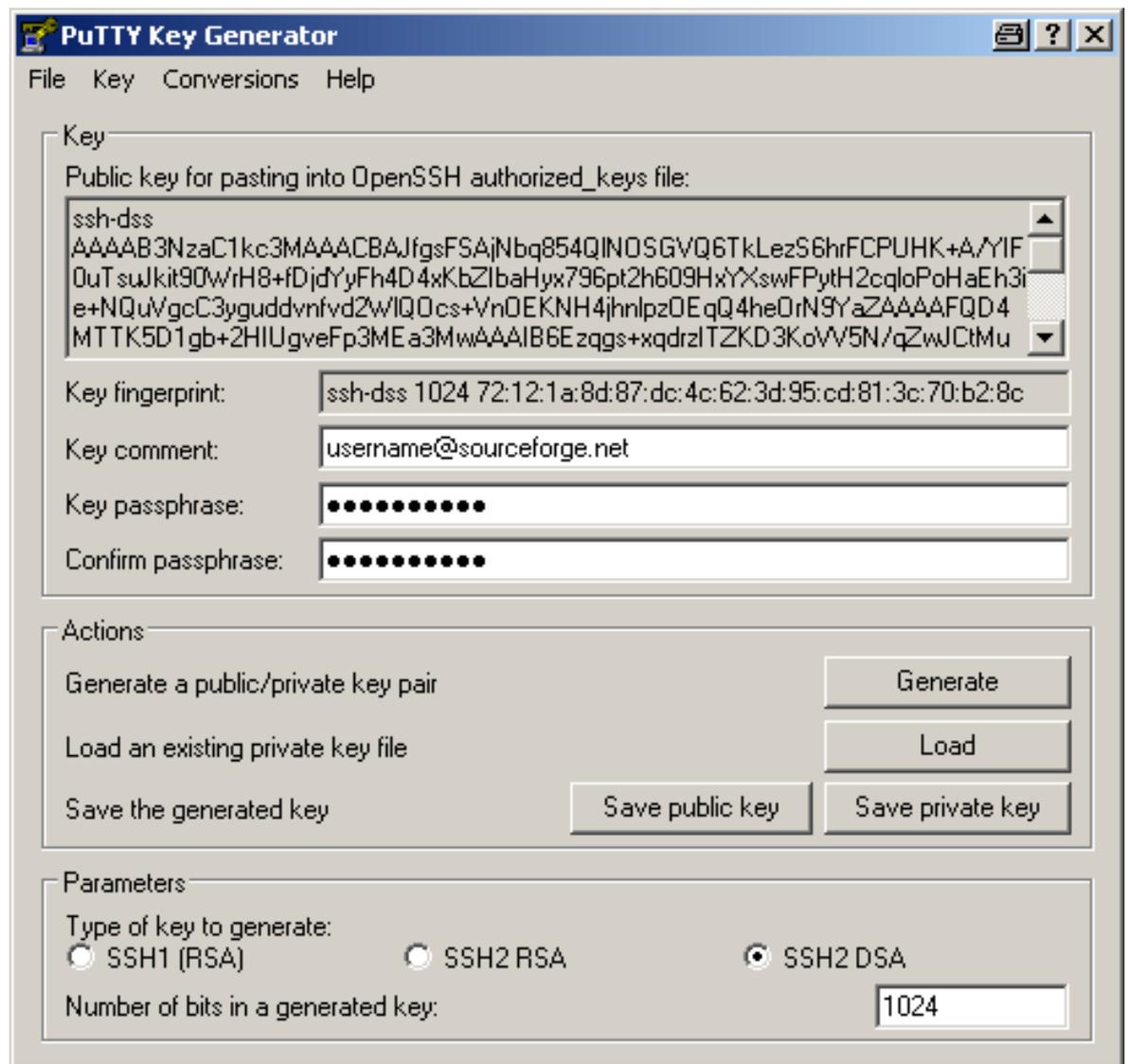


Figure 5.3. Apply a passphrase

- Click on the [Save private key] button; use the resulting dialog to save your private key data for future use. You may use a filename such as "SourceForge-Shell.ppk" or "SourceForge-CF.ppk". The .ppk extension is used for PuTTY Private Key files.

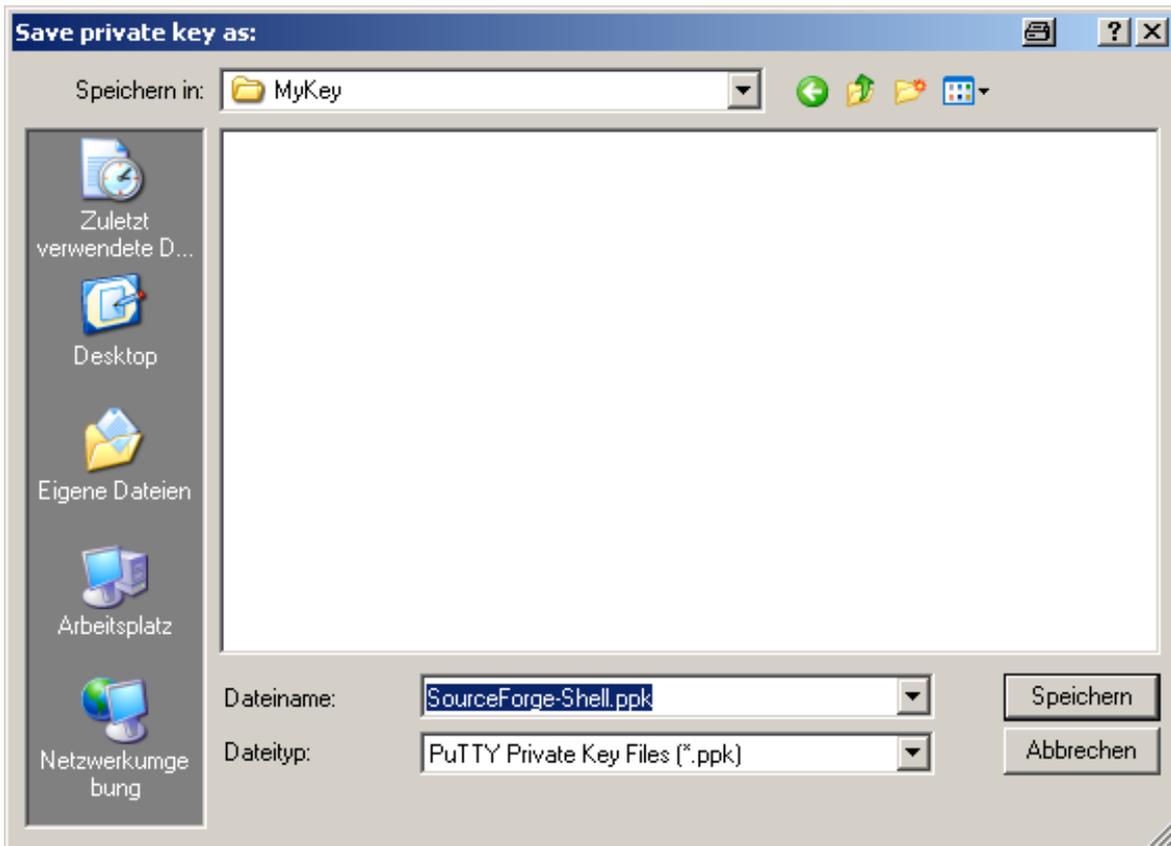


Figure 5.4. Save the private key

5.1.2. Uploading SSH key

- Go to your account page [<https://sourceforge.net/account/>] on SourceForge.net. Under the section “Host Access Information” you will find a link to [Edit SSH Keys for Shell/CVS] [<https://sourceforge.net/account/editsshkeys.php>]. Here you can copy your **public** key data from the [Public key for pasting into OpenSSH authorized_keys2 file] section of the PuTTY Key Generator and paste in to the provided form on the SourceForge.net site.

Host Access Information

Members of registered SourceForge.net projects are provided access to a number of hosts. Access to the project shell and CVS servers is automatically provided to project developers. Access to the SourceForge.net Compile Farm is provided on an opt-in basis. Shell, CVS and Compile Farm services are only available to those users listed as developers on a project.

- [Documentation: Shell services](#)
- [Documentation: Developer CVS services](#)
- [Documentation: Guide to generating, posting and using SSH keys](#)
(allows authentication to shell and CVS servers without the use of passwords)
- [Documentation: Guide to the SourceForge.net Compile Farm](#)

Project shell server: **shell.sourceforge.net**

Project CVS server: **cvs.sourceforge.net**
("cvs1" when accessed from the shell server)

Login shell (on shell server):

Number of SSH Shared Keys on file: **1** [\[Edit SSH Keys for Shell/CVS\]](#)

(Public Keys for project 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).

Access to the project shell, CVS and Compile Farm hosts is provided via SSH. [Documentation: Introduction to SSH](#)

Figure 5.5. Go to the page where you can submit you public key

- Exit the PuTTY Key Generator



Note

There is an delay of about 10 minutes (and maybe up to 30 minutes) between the time SSH keys are uploaded on the SourceForge.net site until they are synchronized to the project shell and CVS servers, or SourceForge.net Compile Farm.

5.1.3. SSH authentication agent



Note

The following step has to be repeated everytime you restart your computer.

- Start the PuTTY SSH agent pageant.exe



Figure 5.6. Start the SSH agent pageant

- The small pageant icon appears in the Windows system tray



Figure 5.7. The pageant icon in the system tray

- Right click on it, and the following menu appears:



Figure 5.8. To add your private key to the SSH agent

- Click on [Add Key] and add the formerly stored private key

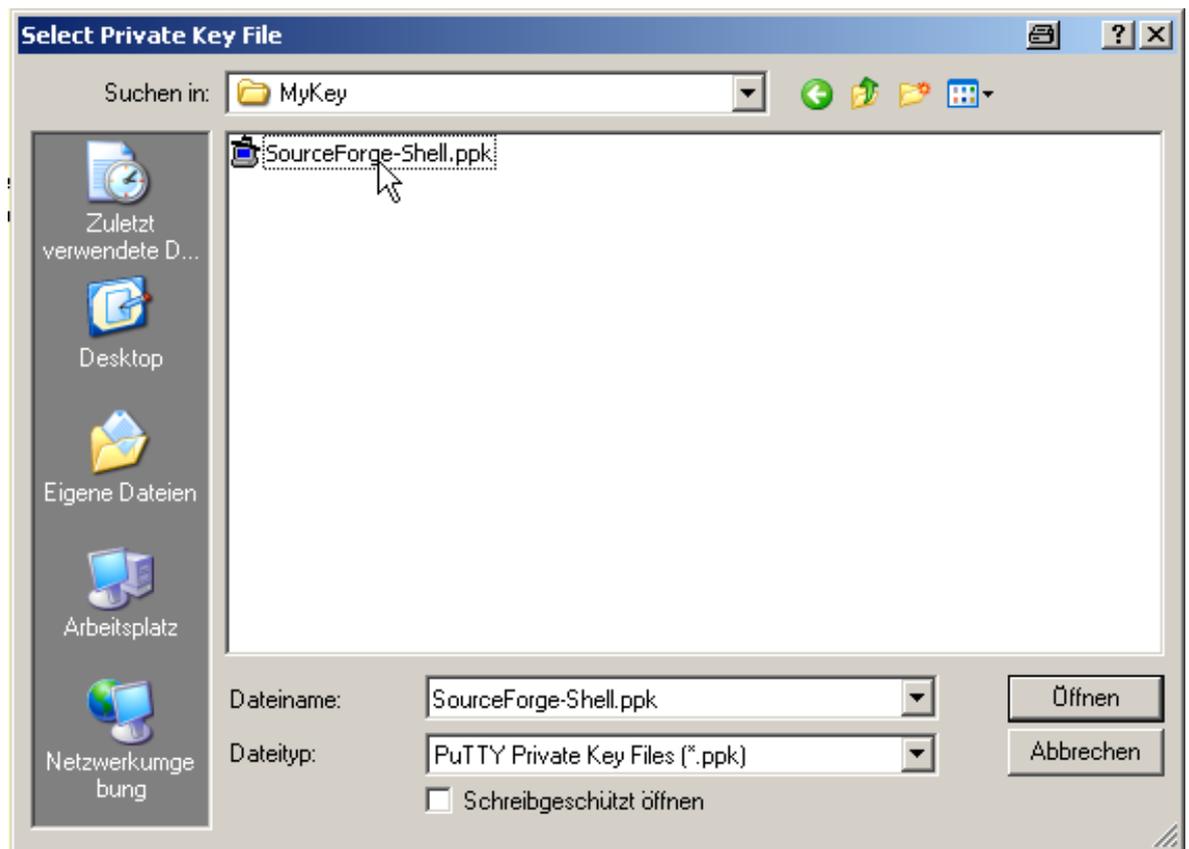


Figure 5.9. Select the previously stored private key.

- Type in the passphrase for that key and click on [Ok] .



Figure 5.10. Type in the passphrase.

5.2. Checking out existing module

In the case you want to work on an existing project, and thus you need to check out the appropriate module files for the first time do the following:

- Within the explorer window open the directory where you want to have the module files, and right click into it. This opens the context menu. Click on [CVS Checkout...] (see Figure 5.11, “Check out menu” [26]).

- This opens the “Checkout Module” dialog. Make the appropriate settings:
 - [Protocol:] Select *Secure Shell (:ext:)*
 - [Server:] type in *cvs.sf.net*
 - [Repository folder:] type in */cvsroot/xoops*
 - [User name:] type in the user name which you have on SourceForge.net
 - [Module:] type in the name of the module you want to have on your local drive, but don't forget the preceding *modules/*. E.g. type *modules/theModuleYouWant*.

Figure 5.11, “Check out menu” [26] shows you a screenshot of how it would look like.

Click on [OK] if everything is prepared.

- In the case everything works well, you will see the “Finished checkout” window with a success message. Figure 5.13, “Finished check with success” [28]) shows the success message after checking out my xdocman module.



Note

Don't forget to start *pageant* and add the appropriate private key for SourceForge.net.

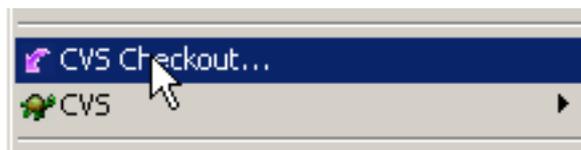


Figure 5.11. Check out menu

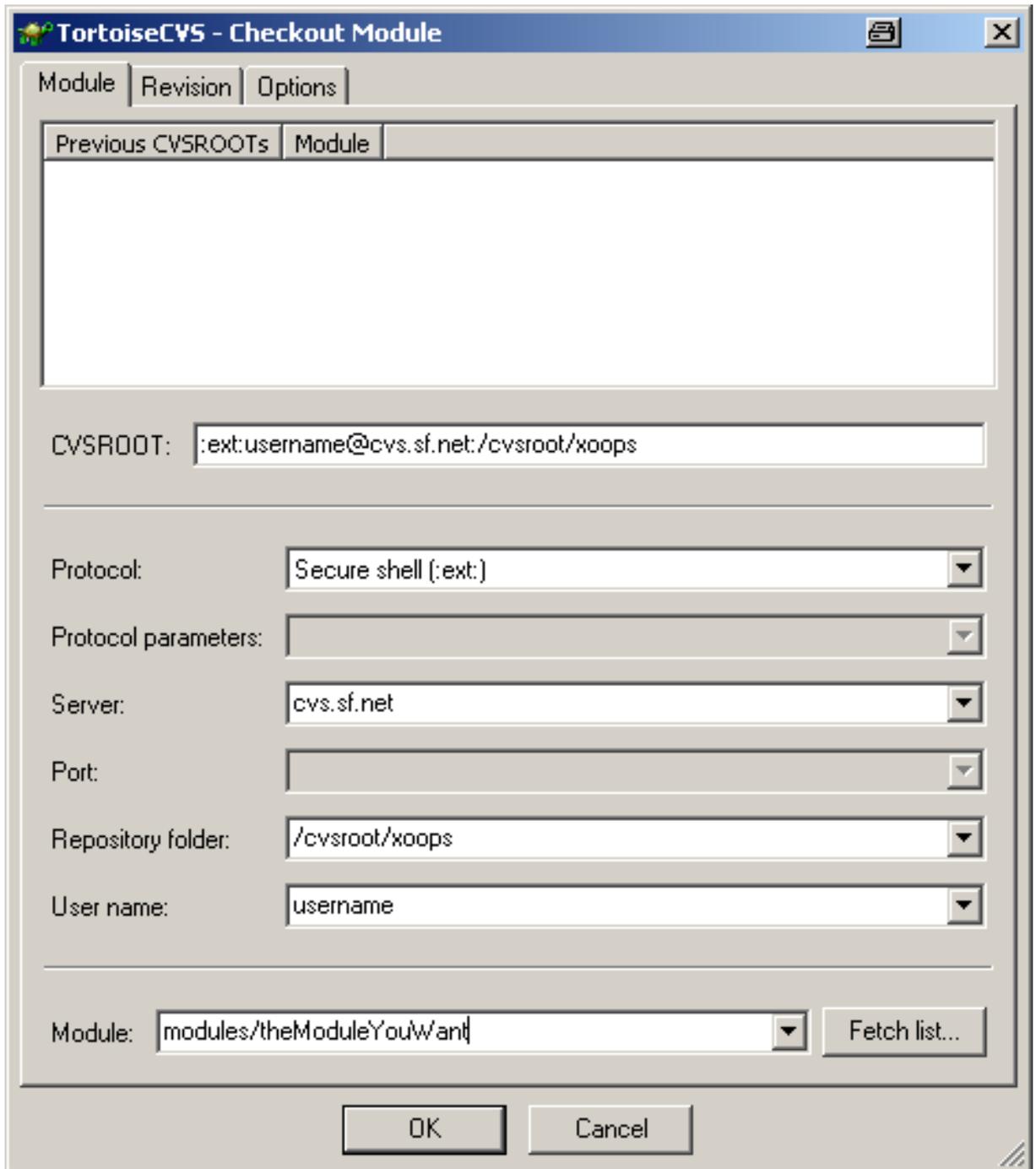


Figure 5.12. Check out menu

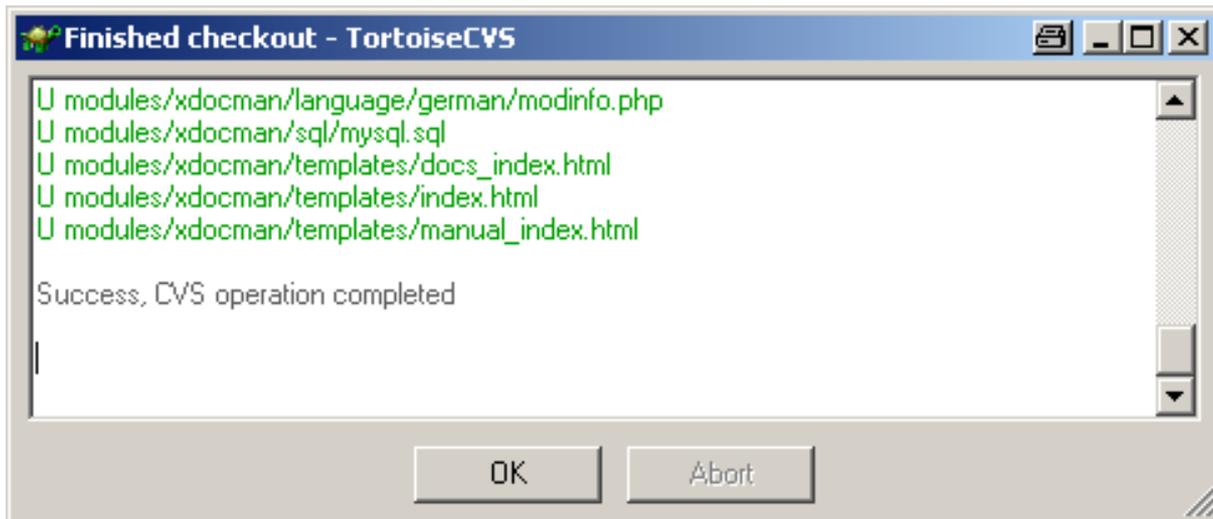


Figure 5.13. Finished check with success

5.2.1. Some variations on checking out

- *Checking out an empty existing module* By default, TortoiseCVS prunes empty directories after the checkout. If a module is empty, then it'll be deleted immediately after it is checked out. Go to CVS -> Preferences -> Policy and uncheck the [Prune empty folders checkbox] .
- *Checking out on a branch* On CVS-submenu click on [Revision Graph...] right click on a version and click on [Tag...] or [Branch...] , then click on [Update list...] and select the tag from the list.
- *Checking out XOOPS core or other modules* You can use these same settings to check out any number of modules or the XOOPS core files, but you will only be able to commit to the modules you have specifically been granted. Change the Module setting to xoops2 for the core or modules to checkout all the modules.

5.3. Adding new module

If you have started a new project as described in Chapter 4, *XOOPS Module Developers* [13] and now you want to bring your project files under version control follow these steps:

- Open an explorer window, go to the appropriate parent directory of your module folder and right click on the module folder icon. This will open the context menu.
- Point to the entry [CVS] . This will open the CVS-submenu, where you click on [Make New Module...] (see Figure 5.14, “The Make New Module... menu entry” [29]).
- This opens the “Make New Module” dialog. Make the appropriate settings:
 - [Protocol:] Select *Secure Shell (:ext:)*
 - [Server:] type in *cvs.sf.net*
 - [Repository folder:] type in */cvsroot/xoops*
 - [User name:] type in the user name which you have on SourceForge.net
 - [Module:] type in *modules/* just before your folder name. So this should look like the following: *modules/your_module_folder*.

Figure 5.11, “Check out menu” [26] shows you a screenshot of how it would look like.

Click on [OK] if everything is prepared.



Note

Don't forget, all module names are **lower case**.

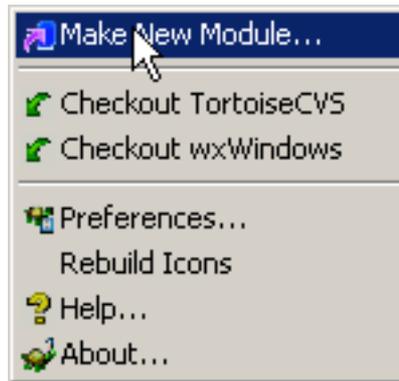


Figure 5.14. The “Make New Module...” menu entry

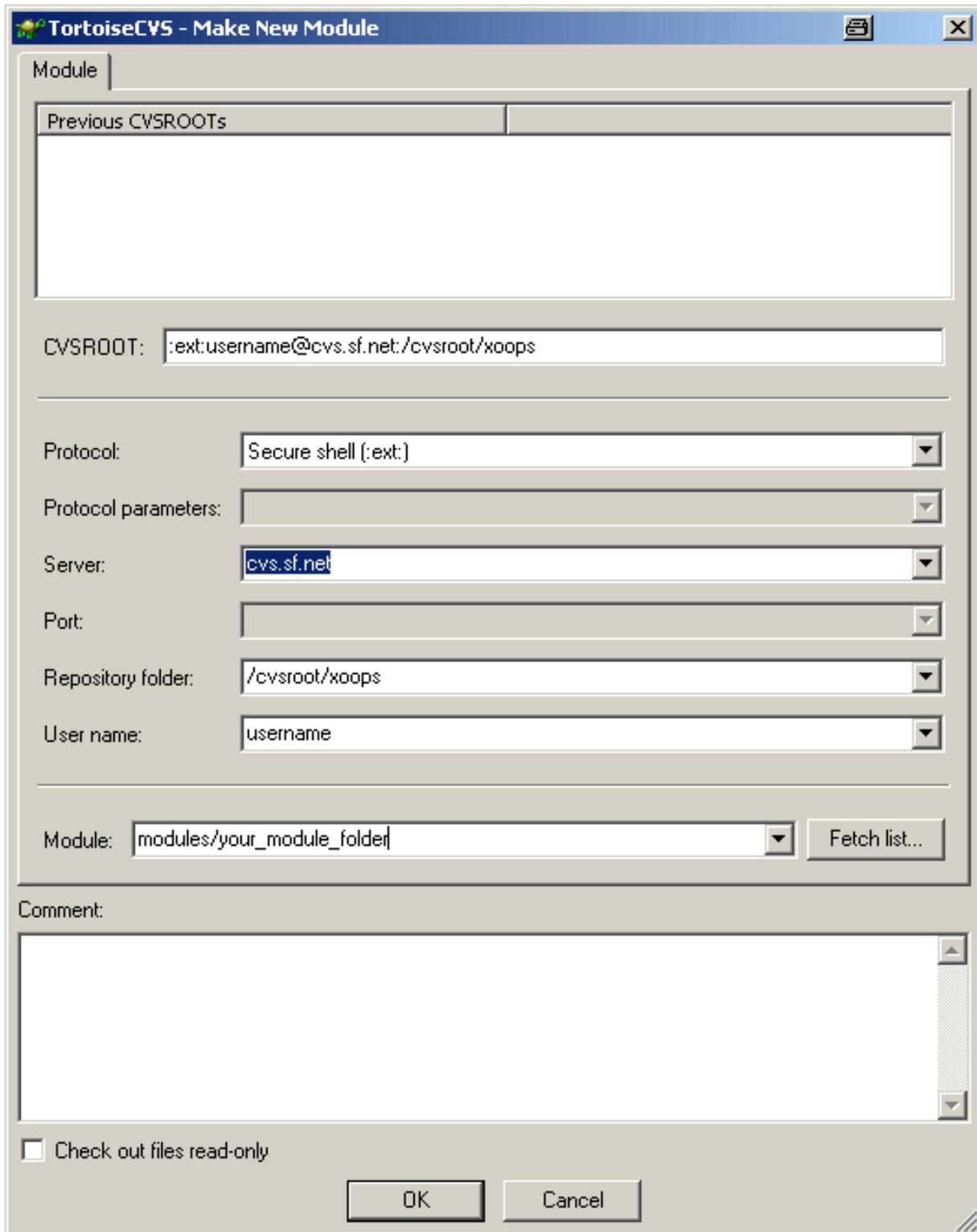


Figure 5.15. The “Make New Module” dialog

Chapter 6. CVS Terminology

CVS naturally has its own terminology. Some terms from CVS are mentioned here for reference. Consult the resources section for more information.

- “Avail file” The avail file exists within the XOOPS CVSROOT directory in the repository. It lists the modules that developers are allowed to commit to.
- “CVSROOT” In CVS terms, CVSROOT can mean a concatenation of the location information for the repository, or a special directory within the repository. A CVSROOT for Joe User (username “juser”) using the ext protocol to contact cvs.sf.net server with the repository root located at /cvsroot/xoops would look like this: “:ext:juser@cvs.sf.net:/cvsroot/xoops”.
- “Karma” Authorization to commit. The mechanism to grant or deny authorization to commit to modules within the **XOOPS** repository reports if you have enough Karma to perform the commit action.
- “Module” A CVS module is a directory within the repository. This is obviously not the same as a **XOOPS** module. Note that there is a modules module within the XOOPS repository. :)
- “Protocol” A method to transport the CVS commands and data between the client and server. “pserver” is widely available but insecure. “ext” allows you to use an external program to set up the communication channel, typically SSH or RSH. SourceForge.net CVS servers require pserver for all anonymous read-only access and SSH for developer access.



Chapter 7. CVS resources

The following resources can help you better understand CVS, how it works, and how to use it with SourceForge:

- The [Sourceforge CVS reference pages](http://sourceforge.net/docman/display_doc.php?docid=14033&group_id=1) [http://sourceforge.net/docman/display_doc.php?docid=14033&group_id=1] provide detailed instruction on how to use CVS with SourceForge.net repositories.
- A nice introduction to CVS and Open Source Development: [Open Source Development with CVS](http://cvsbook.red-bean.com) [http://cvsbook.red-bean.com]. Not all chapters are developer-related, chapter 2 gives a good overview.
- See [TortoiseCVS Support](http://www.tortoisecvs.org/support.shtml) [http://www.tortoisecvs.org/support.shtml] for help with Tortoise operations.
- A quick introduction to command line CVS (by Gentoo Linux) [Gentoo Linux CVS Tutorial](http://www.gentoo.org/doc/en/cvs-tutorial.xml) [http://www.gentoo.org/doc/en/cvs-tutorial.xml].
- Those working on a project at dev.xoops.org should not forget the [CVS discussion forum!](http://dev.xoops.org/modules/newbb/viewforum.php?forum=27) [http://dev.xoops.org/modules/newbb/viewforum.php?forum=27]



Index

C

CVS_RSH, 3

P

PuTTY

 plink.exe, 3

Python, 3

T

TortoiseCVS, 3

W

WinCVS, 3

WinMerge, 4