

Obtaining Developmental Versions of 4Suite™ via CVS

Mike Brown, Fourthought, Inc.
Shatin Rai
Intern
Fourthought, Inc.
Uche Ogbuji
Fourthought, Inc.

This document can be freely translated and distributed. It is released under the LDP License [http://www.linuxdoc.org/manifesto.html].

	Revision History	
Revision 0.0	2000-12-20	UO
	Initial Release	
Revision 0.1	2002-01-27	SR
	0.12 Alpha release	
Revision 0.2	2002-07-01	UO
	0.12a3 release	
Revision 0.3	2002-11-22	MB
	Conversion to Simplified DocBook V1.0CR1; minor edits	
Revision 0.4	2003-02-28	MB
	minor edits	
Revision 0.5	2003-03-27	MB
	minor edits	
Revision 1.0	2003-08-21	MB
	added intro, TortoiseCVS info, many other things	

This document provides step-by step instructions for users to obtain developmental versions of 4Suite™ from the project's public CVS repository.

4Suite is a Python-based toolkit for XML and RDF application development. It features a library of integrated tools for XML processing, implementing open technologies such as DOM, RDF, XSLT, XInclude, XPointer, XLink, XPath, XUpdate, RELAX NG, and XML/SGML Catalogs. Layered upon this is an XML and RDF data repository and server, which supports multiple methods of data access, query, indexing, transformation, rich linking, and rule processing, and provides the data infrastructure of a full database system, including transactions, concurrency, access control, and management tools. It also supports HTTP (including SOAP and WebDAV), RPC, FTP, and CORBA.

Table of Contents

Introduction and Conventions	2
About CVS	2
Conventions	2
Mailing Lists	2
Viewing the 4Suite CVS Repository on-line	2
Prerequisite: a standard C compiler	2

Obtaining the Source Code	3
Getting a CVS Snapshot of the Source Code	3
Getting the Source Code with a CVS Client	3
Building 4Suite on Windows	8
Test the installation	9
Building 4Suite on Unix	10
Resources	10

Introduction and Conventions

This document contains instructions for using CVS to keep up-to-date with the latest 4Suite source code. There is no guarantee that this code will be free of bugs, or that it will even build, so please consider it to be of strictly "alpha" quality.

About CVS

CVS is a free, mature **version control system**: a client/server software system that developers use to store and track changes to source code. Fourthought maintains a CVS server that allows public access, so anyone can get the same code that the developers are using to test bug fixes & feature enhancements for the next version of 4Suite.

Conventions

Some command-line examples in this document may use "\$" or something like "C:\dev>" to represent the command shell prompt. The actual prompt varies from system to system, so please just consider a placeholder. And obviously, they aren't something that you're supposed to type.

Some command-line examples in this document may use the text "SNIP" in place of a great deal of output from an instruction.

Mailing Lists

If you are going to be keeping up with 4Suite via CVS, then you should consider subscribing to the 4suite-dev mailing list [<http://lists.fourthought.com/mailman/listinfo/4suite-dev>]. This is the mailing list on which the developers and other interested parties discuss issues pertaining to the development code, whereas the regular 4suite mailing list [<http://lists.fourthought.com/mailman/listinfo/4suite>] is mainly for users of release versions.

You may also want to know about the 4suite-checkins mailing list [<http://lists.fourthought.com/mailman/listinfo/4suite-checkins>]. This read-only list announces each commit to the CVS repository as it is made. At times, it can be high volume, and not very interesting reading, but its archives [<http://lists.fourthought.com/pipermail/4suite-checkins/>] have proven useful for quickly locating diffs and patches.

Viewing the 4Suite CVS Repository on-line

The 4Suite project CVS repository can be browsed on the web at <http://cvs.4suite.org/> [<http://cvs.4suite.org/cgi-bin/viewcvs.cgi/4Suite/>].

Prerequisite: a standard C compiler

Before proceeding, please note that you need an ANSI/ISO C compiler in order to build 4Suite from source. This is not a problem for most Unix users, since gcc is now standard on most POSIX platforms, including most (if not all) Linux distributions, all of the BSDs, Mac OS X, and Cygwin.

Windows, however, does not come with a C compiler; if you are on this platform, you must provide your own.

4Suite's developers usually use Microsoft Visual C++ 6.0®. Other compilers are supported, but not widely tested—please report any difficulties to the mailing list. After you have downloaded the 4Suite source code, you can invoke `setup.py --help-compilers` to see what compilers are supported.

Obtaining the Source Code

You have two options for obtaining the source code.

One option is to download the source code with a **CVS client**. For an end user, the main advantage to using a CVS client is that you only download the complete source code once. Thereafter, you can use the client to easily download and automatically apply patches for only the files that have changed since your last update.

The other option is to use anonymous FTP to get one of the nightly **CVS snapshots**. A snapshot is the complete source code in one standard .tar.gz file. A new snapshot is generated daily. The downside of getting a snapshot is that you have to download an entire snapshot each time you want to get an update, and it's difficult to tell what changed. Also, if you get a snapshot, you can't just switch to using CVS to get updates, because you won't have some necessary files that are created when you use a CVS client.

Getting a CVS Snapshot of the Source Code

If you prefer to get a CVS snapshot, then this is all you need to do:

1. Get the latest snapshot at `ftp://ftp.4suite.org/pub/cvs-snapshots/4Suite-CVS.tar.gz`. (If you are specifically looking for an older snapshot, feel free to browse `ftp://ftp.4suite.org/pub/cvs-snapshots/`).
2. Unzip the file somewhere. For example, on UNIX, the command `tar xzf 4Suite-CVS.tar.gz` should suffice. Windows users can use a third-party zip utility; the popular ones all support this format. Those who download with IE 5 may find that the filename loses its .gz extension, so be sure to correct the name before trying to unzip it.
3. Windows users, consult the Building 4Suite on Windows section of this document to see how to do the build.

Everyone else, consult the Installing 4Suite on UNIX [UNIX.xml] document.

Getting the Source Code with a CVS Client

Quick Summary for Power Users (any platform)

If you know your way around a CVS client, then here's a summary of the info you need:

- CVS host: **cvs.4suite.org**
- CVS root: **/var/local/cvsroot**
- Authentication type: **password server (pserver)**
- Username: **anonymous**
- Password: **(no password)**

The above translates to the CVSROOT **:pserver:anonymous@cvs.4suite.org:/var/local/cvsroot**

The module to checkout is: **4Suite**

For example, if you are using a command-line version of CVS (e.g. standard cvs on Unix or Cygwin, or CVSNT on Windows), simply run these commands from a shell prompt, pressing Enter when prompted for a password:

```
cvcs -d:pserver:anonymous@cvcs.4suite.org:/var/local/cvsroot login
```

```
cvcs -d:pserver:anonymous@cvcs.4suite.org:/var/local/cvsroot get 4Suite
```

A 4Suite folder will be created in the current directory, and the source code will be downloaded into it.

In the future, you can get the latest changes by simply running `cvcs update -dP`.

Next step: Windows users, consult the Building 4Suite on Windows section of this document to see how to do the build. Everyone else, consult the Installing 4Suite on UNIX [UNIX.xml] document.

Detailed Instructions for Windows Users

Using Windows CVS clients to get 4Suite source code

There are several very good CVS clients available for Windows. The sections below explain where to get and how to use the clients that the 4Suite developers recommend for Windows users. If you want to use a CVS client to download 4Suite, then please pick a client from the list below and follow the appropriate instructions.

- TortoiseCVS - integrates into the Windows file explorer
- WinCvs - standalone GUI
- CVSNT - command-line interface, just like on Unix

TortoiseCVS

TortoiseCVS is a free CVS client that works from context (right-click) menus in the Windows Explorer shell. It's quite good, sporting many more features than are mentioned here.

1. Download and install TortoiseCVS. You can get it from [tortoisecvs.org](http://www.tortoisecvs.org/) [http://www.tortoisecvs.org/].
2. In Windows Explorer, navigate to a location where you'd like the 4Suite code to live. For example, C:\dev.
3. At this point, you may wish to familiarize yourself with TortoiseCVS and adjust its configuration a bit.

Right-click in the Explorer window somewhere, not on a file. On the context menu, navigate to CVS and then to Preferences. A TortoiseCVS Preferences dialog will open. In this dialog, there are a couple of things you'll probably want to change. None of these changes are absolutely necessary, but they are recommended.

- Pick visually pleasing overlay icons. Click on the first "Browse" button in the "Main" tab and pick an overlay set that you like. Your 4Suite source code files will have these overlays on their icons, and chances are, the default overlays are not what you'd prefer. TortoiseCVS will say that you need to reboot to see the new icons, but if you know how to kill the Explorer process and start a new Explorer from the Task Manager, you can just do that.
- If you typically edit text files in something other than Notepad, and especially if there's a chance that you'll be submitting patches to the developers, then do the following: In the "Advanced" tab of the preferences dialog, set "Unix line endings", and unset "Check not Unix sandbox". This ensures that the text files you download with TortoiseCVS will contain single-character newlines (LF, rather than CR+LF), which is the convention the developers use.
- You might want to install a diff application (something that shows the differences between two text files), so that you can compare different versions of files. The FAQ on the TortoiseCVS site has recommendations. Configure TortoiseCVS to use the one you installed by entering the path to the appropriate file in the "External diff application" box on the "Main" tab of the preferences dialog.

4. Now you can do the checkout. Don't worry — "checkout" is CVS's terminology for "download a copy". It's not like other version control systems where the server marks files as being checked out by you and uneditable by others.

Right-click in the Explorer window somewhere, not on a file. On the context menu, pick "CVS Checkout...". The Checkout Module dialog will appear.

5. In the Checkout Module dialog's "Module" tab, enter the following in the CVSROOT box:

:pserver:anonymous@cvs.4suite.org:/var/local/cvsroot

In the "Module" box at the bottom, enter "4Suite". Be sure to capitalize the S in 4Suite. You don't need to set anything in the other tabs, but you might want to look at them to see what is in them, for future reference. Click the "OK" button.

6. A new window, titled "Checking out in C:\dev - TortoiseCVS", will appear, and the 4Suite source code should begin downloading. When it's finished, it should say "Success, CVS operation completed". Click "OK".

That's it! You now have a C:\dev\4Suite folder with all the source code. The folder and its contents will have the overlay icons, identifying the files as having come from a CVS repository. If you modify a file, the overlay icon will change.

From now on, whenever you want to get the latest code, just navigate to the 4Suite folder, right-click on the folder (or in the folder window, if it's open), and choose "CVS Update".

Go ahead and skip down to the Building 4Suite on Windows section of this document to see how to do the build.

WinCvs

WinCvs is a CVS client with a full GUI. Although some users prefer it because it is a separate application, it is not without its quirks, and we are no longer recommending that people use it, now that TortoiseCVS is available.

However, if you wish to try it, download the latest stable version (or whatever version you're comfortable with) from cvsgui.sourceforge.net [http://cvsgui.sourceforge.net/] and install it according to its own instructions.

Then follow these instructions to use it, keeping in mind that they were written for WinCVS 1.2:

1. Start WinCvs.
2. From the menu, select Create.
3. Select Checkout Modules.
4. Click on the Checkout Settings tab.
 - Enter the module name and path on the server: **4Suite**
 - Local folder to check out to: **C:\dev**
5. Click on General tab
 - Enter the CVSROOT: **anonymous@cvs.4suite.org:/var/local/cvsroot**
 - Authentication: "passwd" file on the cvs server

- Use version: `cvs 1.10` (Standard)

6. Click on `Globals`

- Checkout read-only: `unchecked`
- If you typically edit text files in something other than Notepad, and especially if there's a chance that you'll be submitting patches to the developers, then do the following:

Checkout text files with Unix LF: `check`

Ignore the warning it presents when you check the box. In the `WinCvs` tab, change the "Default viewer used to open files" to something other than Notepad.

7. Click `OK`

8. Now you can do the checkout. Don't worry — "checkout" is CVS's terminology for "download a copy". It's not like other version control systems where the server marks files as being checked out by you and uneditable by others.

From the menu bar, select `Admin`.

9. From the drop-down menu, select `Login...`

- If prompted to pick a home directory, choose any directory that is reasonably private, like `C:\Documents and Settings\Your Name`. This is where a password file will live, not where you will be checking out to.
- If presented with a "Password authentication" dialog, just click `OK` (do not enter a password).

You should see these messages in the message window at the bottom:

```
cvs login
(Logging in to anonymous@cvs.4suite.org)
*****CVS exited normally with code 0*****
```

10. From the `Create` menu (at the top), choose `Checkout module...`

A "Checkout settings" dialog will open. Enter the module name **4Suite**. Be sure to capitalize the **S** in 4Suite. Enter **C:\dev** as the local folder to checkout to. Click the "OK" button.

The message window at the bottom should fill up with messages indicating that the download is proceeding, finally ending with:

```
*****CVS exited normally with code 0*****
```

That's it! You now have a `C:\dev\4Suite` folder with all the source code.

From now on, whenever you want to get the latest code, navigate in the left pane to the `4Suite` folder, right-click on it, and choose "Update selection...". In the dialog that pops up, make sure that "Create missing directories that exist in the repository" and "Get the clean copy" are checked. Then click "OK".

Go ahead and skip down to the Building 4Suite on Windows section of this document to see how to do the build.

CVSNT

If you are familiar with the command-line version of CVS, you might want to use CVSNT. CVSNT is a complete native Windows port of the standard CVS server and client, just like on Unix. You can use just the client portion of it.

1. Download the latest stable version of CVSNT from <http://www.cvsnt.com/>.

Go to <http://www.cvsnt.com/> and download and install the stable version. Use the 'Windows install (.exe)' link.

2. Check your environment variables and make sure that your PATH contains C:\Program Files\CVS for NT (or wherever you installed the software). The installer sometimes has trouble modifying your PATH, so you might need to do this yourself.
3. You should now be able to run cvs from the command line. Open a command shell prompt and change to the directory where you want to download the source (e.g., C:\dev).
4. Run the following commands. Just press Enter when prompted for a password.

```
cvs -d:pserver:anonymous@cvs.4suite.org:/var/local/cvsroot login
```

```
cvs -d:pserver:anonymous@cvs.4suite.org:/var/local/cvsroot get 4Suite
```

A 4Suite folder will be created in the current directory, and the source code will be downloaded into it.

That's it! You now have a C:\dev\4Suite folder with all the source code.

In the future, you can get the latest changes by simply running **cvs update -dP** in that folder.

Go ahead and skip down to the Building 4Suite on Windows section of this document to see how to do the build.

Detailed Instructions for Unix Users

CVS comes standard on nearly all Unix systems, so there is no extra software to install. Any user should have access to the cvs command-line client.

1. Open a command shell prompt and change to the directory where you want to download the source (e.g., ~/devel).
2. Run the following commands. Just press Enter when prompted for a password.

```
cvs -d:pserver:anonymous@cvs.4suite.org:/var/local/cvsroot login
```

```
cvs -d:pserver:anonymous@cvs.4suite.org:/var/local/cvsroot get 4Suite
```

A 4Suite folder will be created in the current directory, and the source code will be downloaded into it.

That's it! You now have a ~/devel/4Suite folder with all the source code.

In the future, you can get the latest changes by simply running **cvs update -dP** in that folder.

Now consult the Installing 4Suite on UNIX [UNIX.xml] document to see how to do the build.

Notes for Cygwin Users

You need to follow the instructions for Unix users, but first take note:

- Although obtaining the source code with CVS is relatively painless, when building on Cygwin, there are several prerequisites and package incompatibilities to watch out for. For now, these are detailed in the user-contributed notes and documentation [http://twistedmatrix.com/users/jh.twistd/xml-sig/moin.cgi/FtSuiteWiki_2fUserNotes] in the 4Suite Wiki [<http://twistedmatrix.com/users/jh.twistd/xml-sig/moin.cgi>].
- It is recommended that you only use the command-line CVS client from Cygwin's `cv`s package, rather than one of the native Windows clients, due to the trickiness of end-of-line conventions between Windows and Cygwin. The `cv`s package is available from the 'devel' section of the Cygwin installer (`setup.exe`). You can run the installer even if you already have Cygwin installed; it will allow you to modify your existing installation.

Building 4Suite on Windows

As mentioned above, you need a C compiler. You also need:

- An appropriate version of Python (2.1 or higher). If PyXML is not installed, your Python must have `pyexpat` and `expat` support compiled in. This is usually not a problem, but occasionally, people don't have it.
- An appropriate version of PyXML, if you want to use it (it's not a prerequisite). Acceptable PyXML versions are 0.7.1 or higher, with 0.8.2 or higher recommended. Avoid 0.8.1, which causes crashes. Also, if you are going to install PyXML, but haven't yet, it is recommended that you use the `--without-xpath` and `--without-xslt` options when running PyXML's `setup.py`.

Ensure that your C compiler executables are in your `PATH` and will run from the command line. The default compiler assumed by the installer is MS Visual C++, which uses the executable `cl.exe`. This compiler happens to require `LIB` and `INCLUDE` environment variables be set to point to `MSVC's Lib` and `Include` directories, respectively. Other compilers may have different requirements.

You can specify a different compiler by passing an appropriate `--compiler=compilerID` argument to 4Suite's `setup.py`. To see what other compilers are supported, change to the directory where the source code was downloaded, e.g. `C:\dev\4Suite`, and run `python setup.py --help-compilers`.

Once you have the compiler ready, all you have to do is this:

1. Open a command shell, and change to the directory where the source code was downloaded, e.g. `C:\dev\4Suite`.
2. Run `python setup.py install`.

Alternatively, you could first run `python setup.py build`, if you wanted to just build without installing afterward. It should be safe either way, though, because the install will not proceed if the build fails.

If you got the source with a CVS client, then the installer will detect that you are in 'developer mode' due to the presence of CVS directories, and will turn off verbose output by default. The non-verbose output is the equivalent of passing a `-q` flag to `setup.py`, and will look like this, if all goes well:

```
running install
running build
running config
running build_py
running build_ext
number.c
    Creating library build\temp.win32-2.2\Release\Ft.Lib.number\number.lib
```



```

    and object build\temp.win32-2.2\Release\Ft.Lib.number\number.exp
escapequotes.c
    Creating library build\temp.win32-2.2\Release\Ft.Lib.EscapeQuotesc\EscapeQuotesc.lib
    and object build\temp.win32-2.2\Release\Ft.Lib.EscapeQuotesc\EscapeQuotesc.exp
SNIP
running build_scripts
stubmain.c
running build_tests
running build_ll0n
running install_lib
running install_scripts
running install_data
running install_sysconf
running install_localstate
running install_tests

```

If in developer mode, in order to build with verbose output you'll want to append a `-v` flag to `setup.py`. That is, run `python setup.py -v install`.

If you have trouble with the build or install, post to the 4suite or 4suite-dev mailing lists.

4Suite on Windows will all go into one `Ft` directory. Typically, this will be placed in `C:\Python21`, `C:\Python22\Lib\site-packages`, or `C:\Python23\Lib\site-packages`, depending on the version of Python in use. You can specify a different base directory by passing the `--root=otherDirectory` option to the `install` command, e.g. `python setup.py install --root=C:\dev\otherPython\Lib\site-packages`.

3. The last thing to do is to set your `PATH` environment variable so that the 4Suite command-line tools can be run from anywhere. The place where you configure the path varies with each version of Windows, but is generally available through the System control panel. Right-click on My Computer on the desktop and choose Properties to jump straight to the control panel, and then navigate to the environment variable section from there.

The path to add, as of 4Suite 1.0b1, is `{PYTHONHOME}\Scripts`. So for a typical Python 2.3 install, you would want to add `C:\Python23\Scripts` to your `PATH`. **Note: This location is different than it was in all versions of 4Suite through 1.0a3.** So if you are upgrading, remove the old `{PYTHONHOME}` or `site-packages\Ft\Share\Bin` directory from your `PATH`.

Alternatively, you could just copy the files from the `Scripts` directory to another location that is already in your `PATH`.

Test the installation

You should now be able to test the installation. Change to a directory outside of the 4Suite source, and start the Python interactive interpreter:

```

C:\dev\4Suite>cd ..
C:\dev>python
Python 2.2.2 (#37, Oct 14 2002, 17:02:34) [MSC 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from Ft.Xml import Domlette
>>> from Ft.Xml.Xslt import Processor
>>> print Processor
<module 'Ft.Xml.Xslt.Processor' from 'C:\Python22\Lib\site-packages\Ft\Xml\Xslt\Processor.'
>>>

```

If there are any problems, report them on the 4suite mailing list.

If all you wish to use in 4Suite are the basic command-line tools and Python libraries, then you are done with installation and can proceed to use them. To make sure things are working, try running a few of the command-line tools with the `-h` option to get usage information.

```
4xml -h
```

```
4xslt -h
```

```
4xpath -h
```

```
4rdf -h
```

More thorough regression test suites for the 4Suite Python libraries are included with 4Suite, and you are welcome to try them out. Ideally, in a regular release version of 4Suite, they should all run to completion without errors. Warnings are OK. In alpha and beta versions or CVS snapshots, some errors are to be expected, and do not need to be reported unless the tests cannot even run to completion.

```
C:\>cd \Python22\Share\4Suite\tests
C:\>python test.py -v
SNIP
```

Be prepared for a lot of output. Omit the `-v` if you want somewhat less detail. You can also specify certain tests by appending the relative path to a subdirectory or `.py` file.

If you want to use the XML/RDF repository and server features of 4Suite, then you will need to customize and install a configuration file, initialize a repository, and do some further tests. These topics are covered in the Quick Start [`./QuickStart.xml`] documentation.

Building 4Suite on Unix

The procedure for building from source on Unix does not change, no matter how the source was obtained, and no matter whether the source is for a release or developmental version.

Consult the Installing 4Suite on UNIX [`UNIX.xml`] document to see how to do the build.

Resources

- 4Suite Project Page [<http://sourceforge.net/projects/foursuite/>] - Bug Reports, Feature Requests, Patches
- 4suite.org [<http://4suite.org/>] - Documentation, Downloads, News, CVS, Developer Resources
- 4Suite Users Mailing List [<http://lists.fourthought.com/mailman/listinfo/4suite>] - Peer & Developer Support
- 4Suite Announcement Mailing List [<http://lists.fourthought.com/mailman/listinfo/4suite-ann>] - Infrequent Announcements about 4Suite
- Fourthought, Inc. [<http://fourthought.com/>] - Commercial Development, Consulting & Support