

# FusionViewer Developer's Guide

February 26, 2007

1	Introduction.....	1
2	To Become a FusionViewer Developer.....	2
3	Subscribing to the Mailing List.....	2
4	Getting Started with FusionViewer CVS Repository.....	2
4.1	For Developers using Mac OS X.....	2
4.1.1	Anonymous Retrieve Source Code from CVS Repository.....	2
4.1.2	Access CVS Repository via SSH Authentication.....	3
4.2	For Developers using Windows XP.....	6
4.2.1	Anonymous Retrieve Source Code from CVS Repository.....	6
4.2.2	Access CVS Repository via SSH Authentication.....	6
4.3	For Developers using Linux.....	8
5	Building FusionViewer Project.....	8
5.1	Required Libraries.....	8
5.2	Build Instructions.....	8
5.2.1	Build the ITK Library.....	8
5.2.2	Build I2TK.....	9
5.2.3	Build FusionViewerJNI.....	9
5.2.4	Build FusionViewer2.....	9

## 1 Introduction

FusionViewer is an open source medical image display package developed by [Insightful Corporation](#) and [University of Washington](#). It is designed to improve the physician's ability to interpret the results of combined positron emission tomography (PET) and computed tomography (CT) studies by providing a platform-independent open source display application. As PET/CT combined studies are increasingly being used for the diagnosis, treatment decisions, and follow-up of cancer patients, improving the interpretive value of these studies would directly benefit cancer patients. This software is a display application for facilitating and improving visualization. The effective display of the fused data permits easy viewing of the anatomical information provided by the CT images while simultaneously displaying the functional data provided by PET.

This guide is a developer's guide for those who want to contribute to FusionViewer project. If you just want to use FusionViewer, please go through FusionViewer User's Guide and download the executable application at <http://fusionviewer.sourceforge.net/download.html#demo> instead.

## 2 To Become a FusionViewer Developer

Please visit project web page at <http://fusionviewer.sourceforge.net/getinvolved.html> and <http://fusionviewer.sourceforge.net/developerinfo.html> to find out how to become a member of development team.

## 3 Subscribing to the Mailing List

The first thing you need to do is to join the FusionViewer community. If you have not subscribed to FusionViewer mailing list yet, please do so at <https://lists.sourceforge.net/lists/listinfo/fusionviewer-users>, so that you can communicate with other members of FusionViewer.

## 4 Getting Started with FusionViewer CVS Repository

FusionViewer development takes place on a hosted CVS (Concurrent Versions System) repository at SourceForge ([https://sourceforge.net/cvs/?group\\_id=162712](https://sourceforge.net/cvs/?group_id=162712)). This section describes how to checkout the source code packages.

### 4.1 For Developers using Mac OS X

#### 4.1.1 Anonymous Retrieve Source Code from CVS Repository

To retrieve source code from CVS repository on SourceForge through anonymous checkout, you will need the CVS Client Software ([http://ximbiot.com/cvs/wiki/index.php?title=Main\\_Page](http://ximbiot.com/cvs/wiki/index.php?title=Main_Page)). Please read help document [http://sourceforge.net/docman/display\\_doc.php?docid=14026&group\\_id=1#cvs\\_client](http://sourceforge.net/docman/display_doc.php?docid=14026&group_id=1#cvs_client) for more information.

CVS is supplied with all current UNIX based operating systems (Linux, BSD and MacOS X are supported, among others). . If CVS is not currently installed, refer to your Operating System's manual for instructions. Following is the step-by-step instruction about how to use CVS.

- 1). Create a folder in which we will store a local working copy of FusionViewer CVS repository
- 2). Open the Terminal application and change the directory to the folder you created in step 1)
- 3). To check out a copy of FusionViewer repository, run the command:

```
cvs -z3 -d:pserver:anonymous@fusionviewer.cvs.sourceforge.net:/cvsroot/fusionviewer checkout  
FusionViewer
```

- 4). To bring your copy of FusionViewer up to date with the repository use the following command.

```
cvs -z3 update
```

A SourceForge.net user account is not required to access a repository via anonymous “pserver”. The anonymous user has no password associated with the account. Do not enter a password if prompted for one, just hit the enter key to bypass password entry.

*Note: working copies generated in this manner cannot be used to perform write operations back to CVS Repository. If you want to write your work back to CVS repository later, checkout via SSH authentication is necessary, please see [http://sourceforge.net/docs/F03/en/#developer\\_config](http://sourceforge.net/docs/F03/en/#developer_config) for more information)*

#### 4.1.2 Access CVS Repository via SSH Authentication

To check out source code and later write back to FusionViewer CVS repository, you must be a member of our development team. Also you need both a SSH client (e.g. OpenSSH, <http://www.openssh.com/>) and a CVS client ([http://ximbiot.com/cvs/wiki/index.php?title=Main\\_Page](http://ximbiot.com/cvs/wiki/index.php?title=Main_Page)). Please read SourceForge document [http://sourceforge.net/docman/display\\_doc.php?docid=14026&group\\_id=1#cvs\\_over\\_ssh](http://sourceforge.net/docman/display_doc.php?docid=14026&group_id=1#cvs_over_ssh) for more information. Following is the step-by-step instruction about how to use CVS over SSH.

- 1). Open the Terminal application and configure an SSH key for authentication to the CVS server by running the following commands:

```
$ ssh-keygen -t dsa -C "YOURUSERNAME@shell.sf.net"
```

```
Generating public/private dsa key pair.
```

```
Enter file in which to save the key (/home/username/.ssh/id_dsa):
```

```
Created directory '/home/username/.ssh'.
```

```
Enter passphrase (empty for no passphrase): (be sure to enter a password for the key)
```

```
Enter same passphrase again:
```

```
Your identification has been saved in /home/username/.ssh/id_dsa.
```

```
Your public key has been saved in /home/username/.ssh/id_dsa.pub.
```

```
The key fingerprint is: f3:31:a8:c6:82:18:c8:0f:dd:6b:fb:27:98:83:3d:3b YOURUSERNAME@shell.sf.net
```

- 2). Go to the SSH key posting page ([http://sourceforge.net/docs/F02/en/#key\\_posting](http://sourceforge.net/docs/F02/en/#key_posting)) on the SourceForge.net site. Upload the contents of the id\_dsa.pub (in ~/.ssh/id\_dsa.pub by default), paste the key data to the form on the Account Maintenance page ([https://sourceforge.net/account/login.php?return\\_to=%2Faccount%2F](https://sourceforge.net/account/login.php?return_to=%2Faccount%2F)). Click on the "Update" button to complete the posting process.
- 3). Create folder to store a local CVS working copy
- 4). Change the directory to the folder you created in step 3).
- 5). To check out a copy of FusionViewer repository, run the following command

```
export CVS_RSH=ssh
```

```
cvscvs -z3 -d:ext:YOURUSERNAME*@fusionviewer.cvs.sourceforge.net:/cvsroot/fusionviewer checkout FusionViewer
```

*Enter passphrase (the password you set in step 1) for key “./ssh/id\_dsa” to connect to cvs server*

- 6). To update to the most current versions of FusionViewer, run the command

```
cvscvs -z3 update
```

- 5). To commit a change to cvs repository

Once a file has been added or edited, you must commit it before others will be able to retrieve it. When a commit is done you must specify a comment indicating why the change was made. You should prefix the change with one of the following indicators.

**BUG:** - a change made to fix an error in the code

**ENH:** - new functionality added to the software

**PERF:** - a performance improvement

**STYLE:** - a change made to improve style, comments, and compiler warnings

The cvs command to commit the change is:

```
cvscvs -z3 commit -m "BUG: fixed core dump when passed float data" filename
```

You can also use the syntax below, which omits the -m flag. In this case cvs will start up an editor for you to enter a comment on why you made the change.

```
cvscvs -z3 commit filename
```

- 6). To add a new file or directory

To add a new file or directory to cvs repository use the following commands. The -kb flag should be used for any binary files such as pdf, Word, tiff, etc. Please do not a directory without first checking with other developers

```
cvscvs -z3 add directoryName
```

```
cvscvs -z3 add fileName
```

```
cvscvs -z3 add -kb binaryFileName
```

- 7). To determine what files have changed in the repository

You can determine what files you have edited and also what files have changed since your last update by using update with the -n flag. This will show you what files you have modified with a "M" (modified), what files have changed in the repository since your last update "P" (patch) or "U" (updated), and what files have conflicts with changes you have made "C".

```
cvs -n update
```

- 8). To see your changes to a file

To see the changes that you have made to a file in the FusionViewer you can use the diff command. The first version shows you the changes between your current copy of the file and the version you last updated from. The second version shows the changes between your current version and the most recent copy in the repository. The third version shows changes in all directories.

```
cvs diff fileName  
cvs diff -rHEAD fileName  
cvs diff
```

- 9). To import new module, run the command

```
cvs -z3 -d:ext:YOURUSERNAME@fusionviewer.cvs.sourceforge.net:/cvsroot/fusionviewer import -m "Import  
Module MODULENAME" MODULENAME Insightful start
```

- 10). Additional information on a file

To see the change history of a file you can use the cvs log command as:

```
cvs log fileName
```

To see the current status of a file

```
cvs status fileName
```

To see who last edited each line of a file and when use:

```
cvs annotate fileName
```

Other CVS and SSH client software such as [TortoiseCVS](#), [CVSNT](#), [Cygwin](#) and [WinSCP](#) are also supported by SourceForge.

**Note:** \* *YOURUSERNAME* is your SourceForge.net account username. If you don't have a SourceForge.net account, you can create one at [https://sourceforge.net/account/newuser\\_emailverify.php](https://sourceforge.net/account/newuser_emailverify.php).

## 4.2 For Developers using Windows XP

### 4.2.1 Anonymous Retrieve Source Code from CVS Repository

To retrieve source code from CVS repository on SourceForge through anonymous checkout, you only need a CVS client Software ([http://ximbiot.com/cvs/wiki/index.php?title=Main\\_Page](http://ximbiot.com/cvs/wiki/index.php?title=Main_Page)). Please read help document [http://sourceforge.net/docman/display\\_doc.php?docid=14026&group\\_id=1#cvs\\_client](http://sourceforge.net/docman/display_doc.php?docid=14026&group_id=1#cvs_client) for more information. Here we use WinCvs (<http://www.wincvs.org/>) to show the step-by-step instruction.

- 1). Create a folder in which we will store a local working copy of FusionViewer CVS repository
- 2). [Install WinCvs](#), and configure the local working directory to the folder you created in step 1)
- 3). Invoke WinCvs by double clicking the application icon
- 4). On Menu, select “Admin” → “Command Line...”
- 5). In the pop-up Command Line setting window, input following command to the blank below “Enter the command line (Enter for line break, Ctrl+Enter to confirm dialog):”

```
cvs -z3 -d:pserver:anonymous@fusionviewer.cvs.sourceforge.net:/cvsroot/fusionviewer checkout
FusionViewer
```

- 6). To update, run the command

```
cvs -z3 update
```

SourceForge.net user account is not required to access a repository via anonymous “pserver”. The anonymous user has no password associated with the account. Do not enter a password if prompted for one, just hit the enter key to bypass password entry.

*Note: working copies generated in this manner cannot be used to perform write operations back to CVS Repository. If you want to write your work back to CVS repository later, checkout via SSH authentication is necessary, please see [http://sourceforge.net/docs/F03/en/#developer\\_config](http://sourceforge.net/docs/F03/en/#developer_config) for more information)*

### 4.2.2 Access CVS Repository via SSH Authentication

To check out source code and later write back to FusionViewer CVS repository, you need both a SSH client (e.g. PuTTY, <http://www.chiark.greenend.org.uk/~sgtatham/putty/>) and a CVS client (e.g. WinCvs, <http://www.wincvs.org/>). Also you must be a member of FusionViewer development team. Please read [http://sourceforge.net/docman/display\\_doc.php?docid=14026&group\\_id=1#cvs\\_over\\_ssh](http://sourceforge.net/docman/display_doc.php?docid=14026&group_id=1#cvs_over_ssh) help file for more information. Following is the step-by-step instruction about how to use WinCvs over PuTTY.

- 1). [Install PuTTY](#).
- 2). Configure an SSH key for authentication to the CVS server
  - Invoke PUTTYGEN.EXE application by double clicking the application icon
  - Select the desired key type, "SSH2 DSA", within the "Parameters" section
  - Click on the "Generate" button
  - Follow the instruction to move the mouse over the blank area of the program in order to create random data used by PUTTYGEN to generate secure keys
  - Enter USERNAME\*@shell.sourceforge.net, Enter the desired passphrase in the "Key passphrase" and "Confirm passphrase" fields. If the key will be used for automation of operations (i.e. as part of a script), you may choose to omit this step from the key generation process.
  - 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".
  - Go to the [SSH key posting page](#) on the SourceForge.net site. Copy your public key data from the "Public key for pasting into OpenSSH authorized\_keys2 file" section of the PuTTY Key Generator, and paste the key data to the provided form on the [Account Maintenance page](#). Click on the "Update" button to complete the posting process.
  - Exit the PuTTY Key Generator (PUTTYGEN).
  - Key data sync to hosts from the SourceForge.net site occurs on regular intervals. Your key data will be synchronized to the designated servers (either shell and CVS, or the Compile Farm) after a [short delay](#) (10-30 mins)
- 3). Execute the [Pagent](#) application
- 4). Click "Add Key", add your saved private key file to [Pagent](#) and keep it open on the back.
- 5). Create folder to store a local CVS working copy.
- 6). [Install WinCvs](#), and configure the local working directory to the folder you created in step 1).
- 7). Execute WinCvs application
- 8). Select "Admin" → "Command Line...".
- 9). In the pop-up Command Line setting window, input following command to the blank below "Enter the command line (Enter for line break, Ctrl+Enter to confirm dialog):"

`cvs -z3 -d:ssh:USERNAME*@fusionviewer.cvs.sourceforge.net:/cvsroot/fusionviewer checkout FusionViewer`

*Enter passphrase (the password you set in step 2) for key "../ssh/id\_dsa" to connect to cvs server*

- 10). To adding files or write your changes back to CVS, please refer to [WinCvs -- Daily Use Guide](#)
- Other CVS and SSH client softwares such as [TortoiseCVS](#), [CVSNT](#), [Cygwin](#) and [WinSCP](#) are also supported by SourceForge.net.

**Note:** \* *YOURUSERNAME* is your SourceForge.net account username. If you don't have a SourceForge.net account, you can create one at [https://sourceforge.net/account/newuser\\_emailverify.php](https://sourceforge.net/account/newuser_emailverify.php).

### 4.3 For Developers using Linux

FusionViewer project has not been implemented on Linux system yet. Please join the developer's team if you are interested

## 5 Building FusionViewer Project

After you checkout the FusionViewer module, you can build a project using the retrieved source code packages. This section provides the basic instructions for building FusionViewer project on Window and Mac OS. If you have problems or questions during compilation, please send a detail description to the FusionViewer mail list: [fusionviewer-users@lists.sourceforge.net](mailto:fusionviewer-users@lists.sourceforge.net).

### 5.1 Required Libraries

There are some libraries and tools required to build FusionViewert. Please download those tools and libraries (source code or binary files) from the links given below.

- CMake 2.4.5 (<http://www.cmake.org/HTML/Download.html>)
- ITK 3.0 (<http://www.itk.org/HTML/Download.php>)
- JOGL (<https://jogl.dev.java.net/>);
- JDOM (<http://www.jdom.org/downloads/index.html>);
- JGoodies Looks (<http://www.jgoodies.com/freeware/looks>);

*Note: JOGL v1.1b09 binaries for Windows and Mac OS X, JDOM v1.0, and JGoodies Looks R1.2.2 are included in the ../FusionViewer/lib folder. You may try these as default. If those don't work, you may have to download the newer versions from the proper websites.*

### 5.2 Build Instructions

#### 5.2.1 Build the ITK Library

ITK uses CMake for its build system. To build ITK, you will first have to compile and install CMake, or download pre-compiled binaries. Instructions on building and running CMake can be found at <http://www.cmake.org/HTML/Documentation.html>. ITK is required for I2TK building. We recommend building ITK with BUILD\_SHARED\_LIBS off and replacing all /MD compiler flags with /MT. This will simplify installation and avoid potential DLL version issues. Click Show Advanced Values and under \*\_LINKER\_FLAGS replace /INCREMENTAL: YES with /OPT:REF (there should be 3 instances of this flag). This flag will reduce the size of the DLL by eliminating un-referenced data.

## 5.2.2 Build I2TK

I2TK is an extension package from ITK. It also uses CMake for its build system. The build process is similar to the ITK build process. Set `ITK_DIR` to the ITK build directory.

## 5.2.3 Build FusionViewerJNI

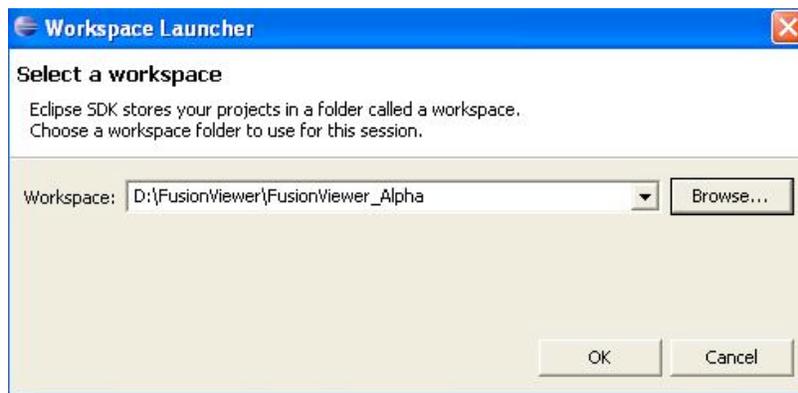
FusionViewerJNI is the Java Native Interface (JNI) used to permit java programs (FusionViewer2) to communicate with and ITK and I2TK libraries. FusionViewerJNI also uses CMake for its build system. Set the following paths:

- `I2TK_DIR` - the I2TK (ITK extension library) build directory
- `ITK_DIR` - the ITK build directory
- `JAVA_HEADERS` (Windows only) - location of the Java headers, such as `jni.h`
- `JAVA_PROJECT_DIR` - the FusionViewer2 directory

## 5.2.4 Build FusionViewer2

Eclipse (<http://www.eclipse.org/>) is an open source platform-independent Integrated Development Environment (IDE), which is recommended for building FusionViewer2.

- 1). Invoke the Eclipse application by double clicking the application icon
- 2). The “Workspace Launcher” window pops up to let you set the workspace. Browse to the directory where FusionViewer2 is (`..\FusionViewer_Alpha`) and click “OK” button.

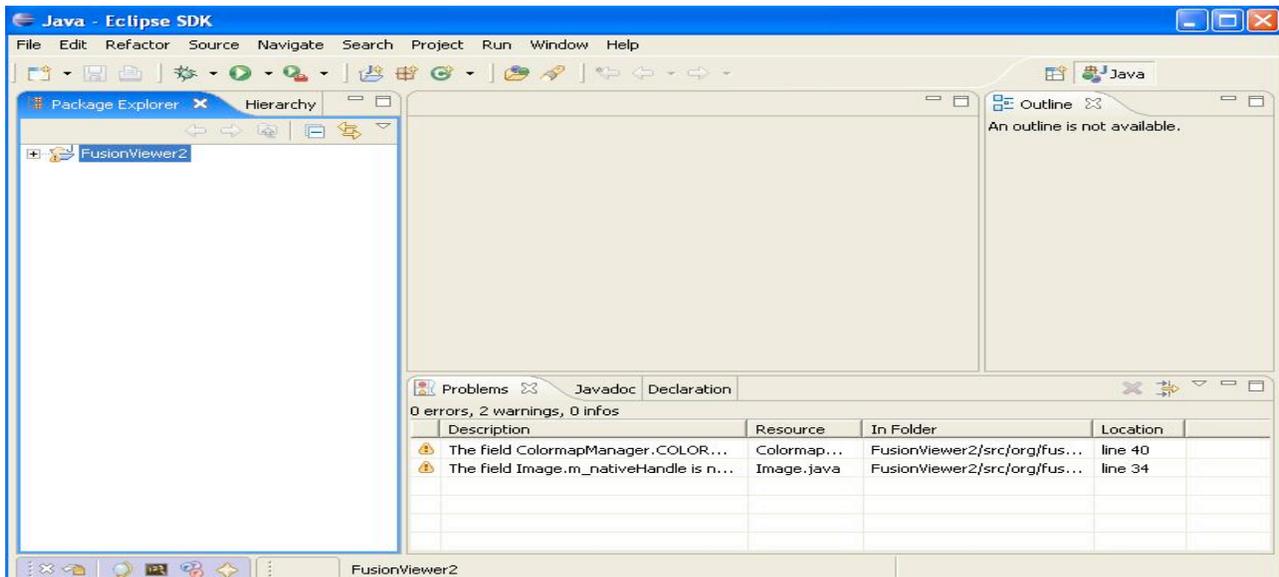


- 3). Eclipse Welcome page appears and click the “workbench” icon at the top-right corner.



- 4). On Eclipse SDK menu, “File” → “New” → “Project...”
- 5). In New Project Wizard, select “Java Project from Existing Ant Buildfile”, click “Next”

- 6). Browse to Ant Buildfile (..\FusionViewr\_Alpha\FusionViewer2\build.xml), click “Finish”.



- 7). In “Package Explorer”, click “+” left to “FusionViewer2” package and expand the project structure.

An Apache Ant build file is included in FusionViewer2 package. Please see the documentation at the top of build.xml.

- ant backupCode – make a code backup
- ant clean -- deletes .class files
- ant [compile] -- compiles the classes
- ant headers -- generates JNI headers (assumes FusionViewerJNI is in the parent directory)
- ant init-- Initialize the timestamp
- ant package -- creates a double-clickable jar file to launch FusionViewer in the dist directory
- ant run – invoke JVM and run FusionViewer application

- 8). On Menu, select “Window” → “Show View” → “Ant”. Drag “build.xml” in the “Package Explorer” and drop it to “Ant View” on the right. You can try those ant actions described above.

More information about how to use Eclipse can be found at <http://www.eclipse.org/documentation/>.