

DEVELOPING SASVIEW WITH ECLIPSE

Steve King, March 2014, Rev 6.1

Installation & Setup

Download & install the Eclipse SDK (eg, eclipse-SDK-4.3.2-win32) to a folder of your choice

Eclipse name their versions: 4.3.1 = Ganymede
 4.3.2 = Kepler
 4.4 = Luna

Do not install a version of Eclipse earlier than 4.3.1

Note that there are some subtle variations in the menu trees between versions! Most of this document was originally written whilst using Ganymede.

If your internet access is normally via a proxy server, locate and edit your
C:\Users\userid\AppData\Roaming\Subversion\servers file and add under the [global] section:

http-proxy-host = site_proxy_server_address

http-proxy-port = site_proxy_server_port

*To see if you use a proxy server, and to get it's address, open Internet Explorer, then go Tools
→ Internet Options → Connections → LAN Settings and look to see if you are using an
automatic configuration script. If so it will probably be configured to use port 8080. If in
doubt, contact your IT Support.*

Launch the program

File → Switch Workspace → Other... → Select a workspace (the default is
drive:\path\Eclipse_workspace)

Window → Open Perspective → Other → PyDev

If PyDev is not listed as a perspective it will be necessary to install it:

*Help → Install New Software... and in the Work With drop-down select the PyDev Update
Site. Let the list of available software refresh.*

Check the box next to PyDev and click Next. The plug-in will install.

If this doesn't work:

*Download the PyDev certificate from http://pydev.org/pydev_certificate.cer and install it (in
Windows, just right-click and select Install certificate)*

Now get PyDev from <http://sourceforge.net/projects/pydev/files/>

If you have Java 7 installed, you must install PyDev 3.4.1 or later.

Extract the zip file to the Eclipse dropins folder and re-start Eclipse

Downloading the Trunk

To do this it is necessary for Eclipse to have a Subversive plug-in installed. Earlier versions of Eclipse (e.g. 4.3.1) seemingly incorporated an SVN plug-in in the default distribution. Later versions (e.g. Kepler) do not!

To install Subversive:

Help → Install New Software... and in the Work With drop-down select the site for the Eclipse release you are using. Let the list of available software refresh.

Scroll down the list of available software and locate Subversive SVN Team Provider Sources (or similar). Check the box next to it and click Next. The plug-in will install.

Restart Eclipse.

Window → Preferences → Team → SVN → SVN Connector

Check the drop-down. If present, select JavaHL (for Windows x86). If this is not present it will be necessary to install the connector.

To install the Subversive Connector:

Help → Install New Software

Alongside the Work With drop-down, click Add and enter a URL in the Location box, e.g:

<http://community.polarion.com/projects/subversive/download/eclipse/3.0/kepler-site/>

(or visit <http://www.polarion.com/products/svn/subversive/download.php> for links to earlier versions of Subversive for Eclipse)

Let the list of available software refresh.

Scroll down the list of available software and locate a JavaHL Binaries plug-in appropriate to your OS. Check the box next to it and click Next. The plug-in will install.

Restart Eclipse.

File → Import → SVN → Checkout Projects from SVN → Next

IF no_SVN_repository_shown:

Check 'Create a new repository location' → Next → Enter a URL for a SVN repository → Next

If Eclipse fails to download the trunk, try repeating the import process but this time using an existing repository (see below)

ELSE:

Check 'Use existing repository location' → Select a SVN repository from the list shown → Next

The SasView SVN repository is hosted by Source Forge at: <https://svn.code.sf.net/p/sasview/code>

Aspiring developers need a) to be registered with Source Forge, and b) authorised by Paul Butler

Eclipse will attempt to download the trunk. In the process it will be necessary to enter your Source Forge login details several times (because you are required to authenticate at each level of the repository) so check the 'Save details' box to speed up the authentication process!

Enter a Workspace Location

This can be any name you like, but will be the name given to your local trunk, and the name of the folder it is stored in; ie, drive:\path\Eclipse_workspace\workspace_location

If you want you can also give this location a custom label.

Do not change any of the other default settings on this dialog box!

Click OK

Preparing to Build SasView

NB: YOU MUST BE CONNECTED TO THE INTERNET!

NB: THIS MAY NOT WORK PROPERLY IF YOU ARE BEHIND A FIREWALL BECAUSE THE BUILD MAKES USE OF PYTHON EASY_INSTALL WHICH DYNAMICALLY DOWNLOADS PACKAGES FROM THE WEB

Install Python

NB: the official version of Python for building SasView is 2.6, though it will build with 2.7

Install NumPy, SciPy, and the additional Python site packages for building SasView

NB: if you are using a Win64 version of Python ensure you also download and install a Win64 version of wxPython compatible with your Python 2.x installation!

Install the other additional packages for building SasView

Install MINGW

In Windows Explorer:

Navigate to drive:\path\Python2x\lib\site-packages and delete any existing sasview .egg file

Back in Eclipse:

Window → Preferences → General → Editors → Text Editors

Check 'Insert spaces for tabs'

Check 'Show print margin' and set the 'Print margin column' to 80

Check 'Show line numbers'

Click OK

Window → Preferences → PyDev → Interpreter – Python

If no Python interpreter is shown, click 'New...', enter a name for your Python installation and 'Browse...' to its python.exe file

Click OK

In the PyDev Package Explorer pane:

Navigate to the sansview folder within the trunk

Right-click on sansview.py → Run As → Python Run

If this fails with an error message in the console pane it means that there is no existing SasView build – which is what you want! If, on the other hand, SasView starts up, search for a sasview .egg file! Then repeat this test.

Locate setup.py in the top level of the trunk

Right-click on run.py → Properties → Run/Debug Settings

IF no_launch_configuration shown:

New → Enter a name for the launch configuration

Select the 'Main' tab → Under 'Project', browse to the name of your trunk

Select the 'Arguments' tab → Type 'install' (without the quotes) in the box marked 'Program arguments'

Click OK

Building SasView the First Time

Right-click on setup.py → Run As → Run Configurations...

Or highlight your_trunk_project setup.py and then choose Run → Run As → Python Run

SasView should build from your local trunk. *If the build fails it will most likely be because a Python package has not installed, in which case install it manually (eg, using easy_install)*

If you get an error about msvc90.dll or msvc90d.dll not being found, copy these from the Visual C++ Redistributable installation (if present) or from a reputable website and put in your MinGW/bin folder.

Re-building SasView after Development

Navigate to the setup.py within the trunk

Right-click on **setup.py** → Run As → Python Run

SasView should build!

The locally-built version of SasView should appear in drive:\path\Eclipse_workspace\workspace_location\dist as something similar to sasview-3.0.0_r0-py2.7-win32.egg

Running SasView from Eclipse

Navigate to the run.py within the trunk

Right-click on **run.py** → Run As → Python Run

SasView should start up!

NB: This is running SasView from compiling and building the Python/C/C++, it is not installing SasView

NB: If you run SasView at this point it may stop an Installer application from building, so if you want an Installer application, create that first (see below)!

To Create a SasView Installer Package

NB: You would not normally need to do this because that is the function of the Build Server!

Navigate to the sansview folder within the trunk

Right-click on **setup_exe.py** → Run As → Python Run

This should create a file **SasView.exe** in the **sansview\dist** folder of the trunk

Double-click on **installer.iss** to bring up the Inno Setup Compiler

Click the green 'play' button to build a SasView installer

This should create a file **setupSasView.exe** in the **sansview\Output** folder of the trunk

Running Unit Tests

Locate an appropriate unit test in your_trunk_project\module\test

Right-click on the utest_test-name.py file → Run As → Python unit-test

NB: unit tests operate on the last built version of SasView, so if you have modified any modules or unit tests you must re-build SasView first!

The unit test your_trunk_project\test\utest_sansview.py executes all the unit tests

NB: you must have previously installed unittest_xml_reporting (eg, easy_install unittest_xml_reporting)

Also see the section on 'Checking the Build Server Tests' below

Committing Code Changes to the SVN Server

NB: run appropriate unit tests first!

Save your changes to file.py

Right-click on file.py → Team → Update (just in case someone has done something to the same file!)

Right-click on file.py → Team → Commit

Add a comment to describe your changes

To See/Compare Code Changes

Either:

Right-click on file.py → Compare with → Revision... / Local History...

Or:

Go to Source Forge

Login

Projects → SasView → Code

NB: if you haven't been coding for a while, right-click on the top level of the trunk and do Team → Update

To Tidy Up the Trunk

Right-click on the top level of the trunk

Team → Cleanup

New Content

Right-click on an appropriate place in the trunk

New → Folder / File

NB: if adding new content, ensure the folder contains: \docs, \src, \sans, \'module_name', \test

After Adding New Content

Right-click on the new content

Team → Add to Version Control

Checking the Build Server Tests

The Build Server automatically runs the unit tests as part of the build process, and reports the success/failure

See, for example:

http://download.mantidproject.org/jenkins/view/All/job/sasview_snowleopard_32bit/

If a build has failed unit tests the 'ball' will be red

Click on the date/time string next to the ball to get a view of what changes were committed to that build and what tests failed

Click on a failed test link to see more detail about why a test failed

Branches

Long-term code development projects, or modifications carrying a significant risk of breaking the code, should not be committed to the trunk, but conducted in a branch of the trunk instead

Note that a branch is not simply a sub-folder of trunk

The top-level of the SasView code repository contains:

- Web - the sasview.org web pages
- Trunk - the code in the current *development* build
- Releases - the code in the current *official release* build
- Branches - copies of trunk being used for special development projects

Only when branches are debugged, working and tested should they be merged back into the trunk.

NB: developers using branches should be aware that their branch copy of trunk will naturally fall behind the current trunk and should therefore periodically update their branch copy of trunk

To create a branch in Eclipse:

Right-click on your_trunk_project → Team → Branch/Tag

Modify the URL shown by adding /branches/name_of_branch

Click Next

Select the HEAD Revision

Click OK

To check if a branch has been created/exists:

Switch to SVN view (top right corner of the Eclipse screen)

To work with a different branch:

In PyDev view, right-click on the top-level of the current trunk → Team → Switch to another Branch/Tag/Revision...

Modify the URL shown to point to the appropriate branch

Click OK