

Eclipse Setup



If you are building sipXecs older than 4.5, look at this page's history for any specific details.

General Remarks

You should feel free to use whatever development environment suits your needs. If you happen to use NetBeans, emacs or anything else please share your experience. Let sipx-dev@sipfoundry.org know if you would like to keep configuration files for your environment in sipXconfig subversion repository.

While I am pretty sure most modern IDEs have the following features, the major reasons why I am using Eclipse today are here:

- excellent code browser
- built-in refactoring support
- code formatter that can be configured to comply with sipX coding standards
- unit test support - writing and executing unit test from IDE
- good subversion plugin - commit, update, diff and graphical merge support
- other plugins for libraries and technologies used in sipXconfig: SQL Explorer, Spring, Tapestry, XML etc.
- decent remote debugging support
- decent GWT debugging support

Installation

Get the latest stable build of Eclipse and install it on the system. Some Linux distributions (Fedora Core 4) include eclipse. However Eclipse install without any problems on Linux, Windows and Mac OS X.

Go to [Eclipse Download Site](#), click on latest stable 3.4 build and download Eclipse SDK version appropriate for your platform.

Eclipse has a notion of workspaces and project. Workspace is a set of related projects that share common options (compilation, code formatting etc.). From my experience it's best to have one workspace per subversion branch (i.e. a separate workspaces for main, 2.8, 2.6 etc.)

Configuration

Set common configuration options.

Classpath

1. Go to Windows->Preference screen
2. Select Java->Build Path->Classpath Variables and add the following

```
SIPX_SRCDIR=path to top source directory
```

```
SIPXCONFIG_SRCDIR=SIPX_SRCDIR/sipXconfig used to avoid some circular imports in Eclipse
```

```
SIPX_BUILDDIR=path to top build directory
```

```
SIPX_JAVADIR=path to the share/java directory where you install sipxecs. Example: /usr/local/sipx/share/java
```

```
LIBSRC=path where you want to store source code for all the dependencies. Not required, but very useful for stepping through code. Example: /home/dhubler/libsrc
```

You need a separate Eclipse workspace for each source code directory, since each branch will have a different SIPX_SRCDIR setting.

Classpath variables used in other projects:

```
OPENFIRE-ROOT=path to openfire root (usually /opt/openfire)
```

Java Code Style Options

Go to Windows->Preferences->Java->Code Style

- add m_ prefix for fields
- uncheck "qualify all generated field access with this."

Go to Windows->Preferences->Java->Code Style->Clean Up

- import sipXconfig/meta/eclipse.code-cleanup.xml - this will make Eclipse cleanup your code (remove extra spaces, add missing annotations etc.)

Go to Windows->Preferences->Java->Code Style->Code Formatter

- import sipXconfig/meta/eclipse.code-formatter.xml - this will make Eclipse format your code according to sipX project coding standards (press Esc Ctrl-F to format your code)

Go to Windows->Preferences->Java->Code Style->Code Templates

- Import templates from: sipXconfig/meta/eclipse.code-templates.xml - this will make Eclipse generate SIPfoundry-compliant file headers

Consider configuring Save Actions to clean-up changed files and format changed lines in the file.

Java Compatibility Options

SipXconfig requires Java 1.6:

1. Go to Windows->Preferences->Java->Compiler and check that the "compiler compliance level" is set to 1.6.
2. Go to Windows->Preferences->Java->Installed JREs and select a 1.6 JRE. If there is no 1.6 JRE on your computer, then you will need to install one.

Other useful options

Plugins

Installing these plugins is optional, but they do have plenty of features that would simplify development.

Some plugins have "update sites" - you can easily update them through Eclipse UI.

1. Help->Software Updates->Search and Install
2. press Import Sites and select sipXconfig/meta/eclipse.update-sites.xml

You can now get the following plugins:

- [Spring IDE](#) : Spring framework support (bean file verification)

Other plugins have to be downloaded and install manually (by copying the plugin to /path/to/eclipse/plugins directory).

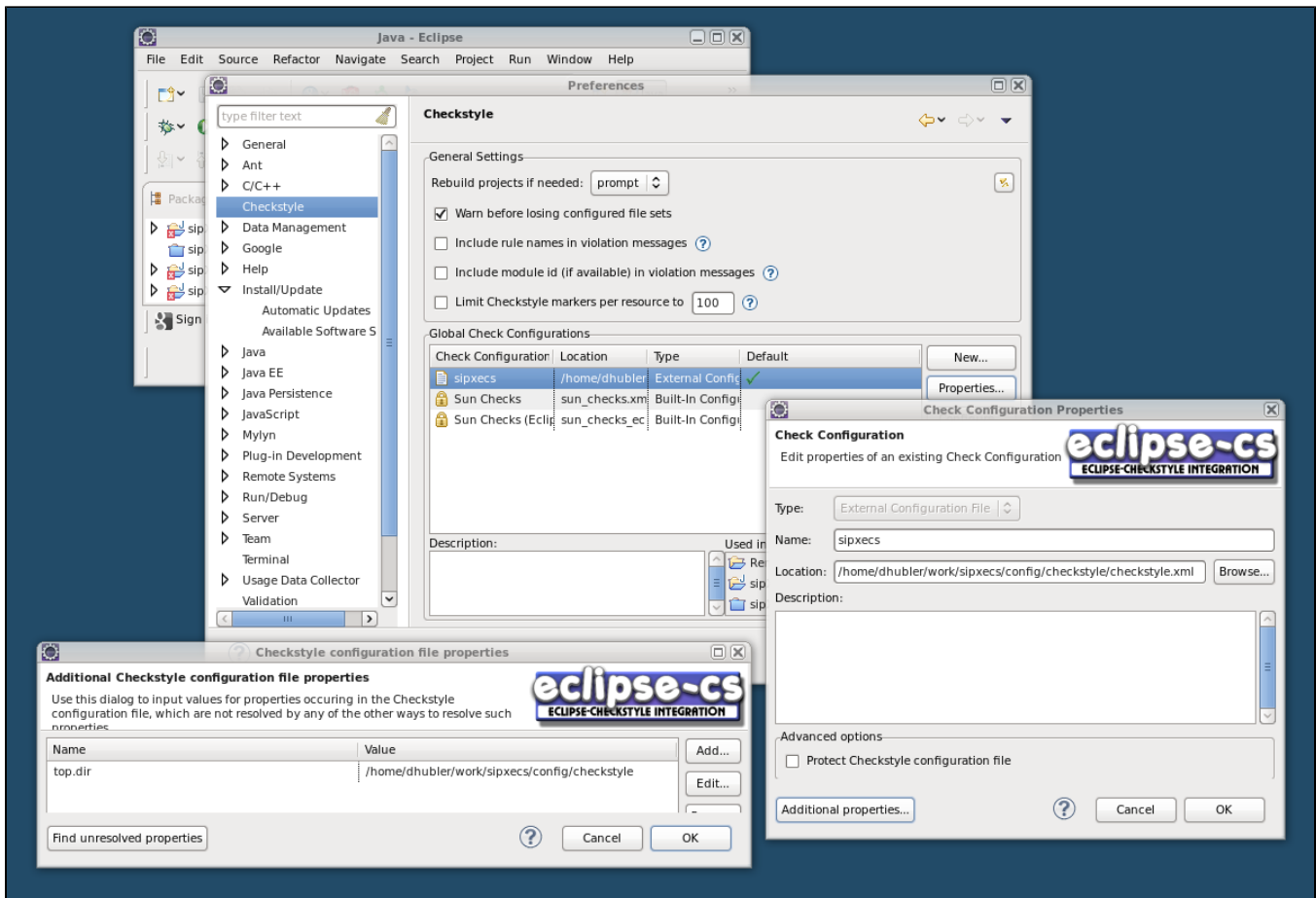
Checkstyle Plugin

Step 1. Install checkstyle plugin from with update URL: <http://eclipse-cs.sf.net/update/>

Step 2. Go to Window->Preferences->Checkstyle

Step 3. Click "New..." in the "Global Check Configurations" and add new "External Configuration File" entry call "sipxecs" with location specifying path to *your-source-directory/config/checkstyle/checkstyle.xml*. Before leaving this dialog, click on "Additional properties", and specify *your-source-directory/config/checkstyle* for variable "top.dir"

Step 4. Click on the new "sipxecs" Check Configuration and "Set as Default" button.



Projects

The last step is opening sipXconfig projects. All necessary configuration files are kept in subversion repository (these are .project and .classpath file in sipXconfig subdirectories) - it's easy to open projects in Eclipse by importing them.

- go to File->Import..
- select General->Existing Project into Workspace
- browse for your sipXecs source code directory (the same as set in SIPX_SRCDIR property above) and press OK
- Eclipse will find all eclipse projects, select only the project you think you'll need. You will almost always need sipXcommons project because most project will require that project to be loaded.
- press Finish

All projects are automatically compiled in background. You are ready to go, at least for the Java projects. You need to add C++ support to be able to build everything from Eclipse; here's information; [Eclipse for C++ projects](#)

Problems

If building doesn't work; some tips:

- It's required to first build the whole project once from the command line in the build directory you specified
- If errors like 'Resource 'sipxcommons' does not exist' keep popping up look at the properties of the project; you might need to delete and add it, Eclipse will show you what it can't find, just delete and add again.
- Same goes for libraries.
- At this moment (SipXcom 15.06) it seems not possible to compile all projects in Eclipse using the above settings and the C++ project things. You might want to stick to the java Projects.