

Table of Contents

Instructions to configure the Eclipse IDE for LHCh.....	1
LXPLUS.....	1
Prerequisites.....	1
Java.....	1
Juno (Eclipse 4.2).....	1
Extensions.....	2
Configuration.....	2
General.....	2
C/C++.....	2
Pydev.....	3
Team.....	3
Customization.....	3
Repositories.....	3
Indigo (Eclipse 3.7) / Helios (Eclipse 3.6) / Galileo (Eclipse 3.5).....	4
Extensions.....	5
Configuration.....	5
General.....	5
C/C++.....	6
Pydev.....	6
Team.....	6
Customization.....	6
Repositories.....	6
Savannah.....	7
Ganymede (Eclipse 3.4).....	7
Basic installation.....	8
Using lhcb-managed eclipse version.....	8
Private laptop, private eclipse version:.....	8
Extensions.....	9
Subversion.....	9
Python.....	9
Mylyn Web Templates Connector.....	10
HTML/XML (optional).....	10
Color Editor (optional).....	10
Configuration.....	10
General.....	10
C/C++.....	10
Pydev.....	11
Team.....	11
Customization.....	11
Savannah.....	11

Instructions to configure the Eclipse IDE for LHCb



LXPLUS

I just prepared a shared installation of Eclipse Indigo with all the suggested plugins that is usable on lxplus. Just do:

```
ssh lxplus.cern.ch
eclipse
```

Please, report any problem or suggestions to me.

Prerequisites

Java

I have problems with the version of the Java Runtime Environment distributed with SLC5 (java-1.6.0-openjdk-1.6.0.0-0.30.b09), the version in SLC4 is simply too old.

I suggest to take the latest version of the JRE from <http://java.com> (at least 1.6.0.10). You can chose any kind of installation, just remember if you got the 32 or 64 bit version.

Juno (Eclipse 4.2)

Download "Eclipse IDE for C/C++ Developers" for your platform at <http://www.eclipse.org/downloads/>, go to a directory of your choice and unpack it the .tar.gz file. You will find the directory `eclipse`.

The first thing to do is to edit the file `eclipse.ini` in that directory. The original looks more or less like:

```
-startup
plugins/org.eclipse.equinox.launcher_1.3.0.v20120522-1813.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.gtk.linux.x86_64_1.1.200.v20120522-1813
-product
org.eclipse.epp.package.java.product
--launcher.defaultAction
openFile
-showsplash
org.eclipse.platform
--launcher.XXMaxPermSize
256m
--launcher.defaultAction
openFile
-vmargs
-Dosgi.requiredJavaVersion=1.5
-Dhelp.lucene.tokenizer=standard
```

```
-XX:MaxPermSize=256m  
-Xms40m  
-Xmx1024m
```

Change the line `-Xmx384m` with `-Xmx512m` and add the line

```
-Duser.name=<your name>
```

(replacing `your name` with your name).

Now you are ready to start Eclipse: go to the `eclipse` directory and launch `./eclipse` (there is no need to go there, you can also use the full path to the `eclipse` executable).

When it asks for a workspace, you probably want to use your `cmtuser` directory.

Probably, the first thing you want to do is to update the pre-bundled plugins: select the menu "Help->Check for Updates". Sometimes there are problems with some updates, but usually it is enough to remove and re-add the corresponding update site.

Extensions

There are a few extensions to Eclipse that are good to have. I put them in a custom update site, including a couple of meta-plugins for easy installation. To install them, select the menu entry "Help->Install new software", type `http://cern.ch/lhcbproject/dist/eclipse/update4.2` in the combo box at the top of the dialog and press enter. Select the entry `LHCb Standard Features Collection` (support for SVN, Python, C/C++, SetupProject, getpack, shell scripts and CMT requirements files, etc.) or `LHCb Extended Features Collection` (same as above with Web and XML support). If there are problems with the installation of the plugins, try to update your Eclipse first (menu "Help->Check for Updates"), if the problem persists, let me know.

Configuration

First of all I suggest to disable the automatic build. It is a very good feature: whenever you modify a file, Eclipse will recompile the project in background. Unfortunately it is a pain if the compilation takes a long time even when you modify a single file. So, select the menu "Project" and make sure that the entry "Build Automatically" is unchecked.

For the configuration, enter the preferences dialog from the menu bar "Window->Preferences".

Note that the preferences are workspace specific, so when creating a new workspace, you should specify that you want to copy the configuration, otherwise you will have to re-apply them to the new one (you can also make a copy of an empty workspace as a template).

General

Editors

Text Editors

set "Displayed tab width" to 8, enable "Use spaces for tabs" and enable "Show print margin" (this last one is optional)

C/C++

Build Console

increase the number of lines for the output (e.g. 8192)

Code Style

a predefined Code Style file for LHCb is attached to this page, you just have to download it and import it: https://twiki.cern.ch/twiki/pub/LHCb/EclipseConfiguration/LHCb_C++_CodeStyle.xml

Editor

Folding

Enable all folding types, select only "Macros" and "Inactive Preprocessor Branches" for initial folding (uncheck the others).

File Types

add *.icc and *.icpp as "C++ Source File"

Task Tags

add `FIXME` with priority "High"

Pydev

Interpreter - Python

add (at least) a python interpreter, probably you want to add the one you find with `which python` (which is the one that LbLogin adds to the path)

Team

Ignored Resources

add the following patterns to the list

```
◇ *-slc5-*
◇ *-slc6-*
◇ setup.*
◇ cleanup.*
◇ version.cmt
◇ *.make
◇ *.makesav
◇ Makefile
◇ NMake
◇ install.history
◇ gdd.dtd
◇ genConf
◇ QMTest
◇ *.launch
```

Customization

Repositories

It is useful to add the standard Subversion repositories at this point (not needed if you use the **GetPack Wizard**)

From the menu "File->New->Other..." select "SVN/Repository Location". Copy the repository URL into the first field, it should automatically split it correctly. Confirm the data by clicking on "Finish". It will ask for username and password to connect (it doesn't use by default kerberos or ssh keys). The complete list of Subversion repositories is:

- `svn+ssh://svn.cern.ch/repos/gaudi`
- `svn+ssh://svn.cern.ch/repos/lhcb`
- `svn+ssh://svn.cern.ch/repos/dirac`

or the following, if you prefer to use anonymous access:

- `http://svn.cern.ch/guest/gaudi`
- `http://svn.cern.ch/guest/lhcb`
- `http://svn.cern.ch/guest/dirac`

-- MarcoClemencic - 12-Jul-2012

Indigo (Eclipse 3.7) / Helios (Eclipse 3.6) / Galileo (Eclipse 3.5)

Download "Eclipse IDE for C/C++ [Linux] Developers" for your platform at <http://www.eclipse.org/downloads/>, go to a directory of your choice and unpack it the .tar.gz file. You will find the directory `eclipse`.

The first thing to do is to edit the file `eclipse.ini` in that directory. The original looks more or less like:

```
-startup
plugins/org.eclipse.equinox.launcher_1.2.0.v20110502.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.gtk.linux.x86_64_1.1.100.v20110505
-product
org.eclipse.epp.package.linuxtools.product
--launcher.defaultAction
openFile
-showsplash
org.eclipse.platform
--launcher.XXMaxPermSize
256m
--launcher.defaultAction
openFile
-vmargs
-Dosgi.requiredJavaVersion=1.6
-XX:MaxPermSize=256m
-Xms40m
-Xmx384m
```

Change the line `-Xmx384m` with `-Xmx512m` and add the line

```
-Duser.name=<your name>
```

(replacing `<your name>` with your name).

If you have problems with the version of Java installed on your system, you can install another one and add, just before the line `-vmargs`, (if not present)

```
-vm
<path/to/preferred/java>
```

The value of `<path/to/preferred/java>` is system dependent (see <http://wiki.eclipse.org/Eclipse.ini> for more information).

Note: the preferred JRE, can also be selected creating a symlink called `jre` in the eclipse directory pointing to the actual jre location, thus making the `-vm` part of `eclipse.ini` unnecessary.

At the end you should have something like (that's mine):

```
-startup
plugins/org.eclipse.equinox.launcher_1.2.0.v20110502.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.gtk.linux.x86_64_1.1.100.v20110505
-product
org.eclipse.epp.package.linuxtools.product
--launcher.defaultAction
openFile
-showsplash
```

```
org.eclipse.platform
--launcher.XXMaxPermSize
256m
--launcher.defaultAction
openFile
-vmargs
-Dosgi.requiredJavaVersion=1.6
-Duser.name=Marco Clemencic
-XX:MaxPermSize=256m
-Xms40m
-Xmx512m
```

Now you are ready to start Eclipse: go to the `eclipse` directory and launch `./eclipse` (there is no need to go there, you can also use the full path to the `eclipse` executable).

When it asks for a workspace, you probably want to use your `cmtuser` directory.

Probably, the first thing you want to do is to update the pre-bundled plugins: select the menu "Help->Check for Updates". Sometimes there are problems with some updates, but usually it is enough to remove and re-add the corresponding update site.

Extensions

There are a few extensions to Eclipse that are good to have. I put them in a custom update site, including a couple of meta-plugins for easy installation. To install them, select the menu entry "Help->Install new software", type <http://cern.ch/lhcbproject/dist/eclipse/update3.7> (for Helios use <http://cern.ch/lhcbproject/dist/eclipse/update> and for Galileo <http://cern.ch/marco.clemencic/lhcb/updates>, but they are not up-to-date) in the combo box at the top of the dialog and press enter. Select the entry `LHCb Standard Feature Set` (support for SVN, Python, C/C++, SetupProject, getpack, shell scripts and CMT requirements files) or `LHCb Extended Feature Set` (same as above with Web and XML support). If there are problems with the installation of the plugins, try to update your Eclipse first (menu "Help->Check for Updates"), if the problem persists, let me know.

Configuration

First of all I suggest to disable the automatic build. It is a very good feature: whenever you modify a file, Eclipse will recompile the project in background. Unfortunately it is a pain if the compilation takes a long time even when you modify a single file. So, select the menu "Project" and make sure that the entry "Build Automatically" is unchecked.

For the configuration, enter the preferences dialog from the menu bar "Window->Preferences".

Note that the preferences are workspace specific, so when creating a new workspace, you should specify that you want to copy the configuration, otherwise you will have to re-apply them to the new one (you can also make a copy of an empty workspace as a template).

General

Editors

Text Editors

set "Displayed tab width" to 8, enable "Use spaces for tabs" and enable "Show print margin" (this last one is optional)

C/C++

Build Console

increase the number of lines for the output (e.g. 8192)

Code Style

a predefined Code Style file for LHCb is attached to this page, you just have to download it and import it: https://twiki.cern.ch/twiki/pub/LHCb/EclipseConfiguration/LHCb_C++_CodeStyle.xml

Editor

Folding

Enable all folding types, select only "Macros" and "Inactive Preprocessor Branches" for initial folding (uncheck the others).

File Types

add *.icc and *.icpp as "C++ Source File"

Task Tags

add FIXME with priority "High"

Pydev

Interpreter - Python

add (at least) a python interpreter, probably you want to add the one you find with `which python` (which is the one that LbLogin adds to the path)

Team

Ignored Resources

add the following patterns to the list

```

◇ *-slc5-*
◇ slc4*_gcc*
◇ win32_vc71_dbg
◇ osx104*_gcc*
◇ setup.*
◇ cleanup.*
◇ version.cmt
◇ *.*make
◇ *.*makesav
◇ Makefile
◇ NMake
◇ install.history
◇ gdd.dtd
◇ genConf
◇ QMTest
◇ *.launch

```

Customization

Repositories

It is useful to add the standard Subversion repositories at this point.

From the menu "File->New->Other..." select "SVN/Repository Location". Copy the repository URL into the first field, it should automatically split it correctly. Confirm the data by clicking on "Finish". It will ask for username and password to connect (it doesn't use by default kerberos or ssh keys). The complete list of Subversion repositories is:

- svn+ssh://svn.cern.ch/repos/gaudi
- svn+ssh://svn.cern.ch/repos/lhcb
- svn+ssh://svn.cern.ch/repos/dirac

or the following, if you prefer to use anonymous access:

- <http://websvn.cern.ch/guest/gaudi>
- <http://websvn.cern.ch/guest/lhcb>
- <http://websvn.cern.ch/guest/dirac>

If you still need the CVS repository, from the menu "File->New->Other..." select "CVS/CVS Repository Location". Copy the string `:ext:isscvcs.cern.ch:/local/repos/lhcb` into the first field (host), it should automatically split it correctly. You have to set the username before being able to complete the creation with the button "Finish".

Savannah

The "Web Templates" connector allows the connection of the Mylyn feature to the savannah repository.

These are the steps needed to add "Gaudi Patches"

- Open the "Task Repository" view (from the dialog "Window->Show View->Other")
- Click on the "Add Task Repository" button (or the entry in the context menu of the view)
- Select "Web Template", then "Next"
- Fill the form with the values:

Server

`https://savannah.cern.ch/patch/?group=gaudi`

Label

Gaudi Patches

Additional Settings

add the parameters `group -> gaudi` and `tracker -> patch`

Advanced Configuration

Task URL

`https://savannah.cern.ch/${tracker}/?`

New Task URL

`${serverUrl}&func=additem`

Query Request URL

`${serverUrl}&chunksz=150&report_id=100`

Query Pattern

`<tr.*?prior[a-z]({Status}closed)?.*?#({Id}[0-9]+).*/td>\n<td ><a.*?>`

- Click on "Finish" and say "Yes" when asked to create a new query on the created repository.
- Accept the defaults in the form for the new query.
- Open the "Task List" view from the dialog "Window->Show View->Other".

Other repositories can be created using the same "Advanced Configuration" and changing "Server" and "Additional Settings" (reminder: the tracker is one of `patch`, `task` or `bugs`).

It is possible to create advanced queries (e.g. bugs assigned to somebody)... TODO

Note: This is not the only way to add a Task Repository for Savannah, but it is the best I could find. One can use a single "repository" for all the trackers, but then it is not possible to create new tasks from Eclipse.

-- MarcoClemencic - 23-Sep-2011

Ganimede (Eclipse 3.4)

 This part is kept for historical reasons and not updated. Please, use Galileo.

Basic installation

Using lhcb-managed eclipse version

If you have the lhcb-software environment available to you, you will probably want to use the lhcb-software version of eclipse.

Eventually you will be able to use `install_project` as with all other software. For the time being if you should follow the ManualInstallLHCbEclipse instructions.

Private laptop, private eclipse version:

If you don't have access to the lhcb software environment, or you want a fully configurable eclipse version for your own choosing, follow these instructions.

Download "Eclipse IDE for C/C++ Developers" for your platform at <http://www.eclipse.org/downloads/>, go to a directory of your choice and unpack it the .tar.gz file. You will find the directory `eclipse`.

The first thing to do is to edit the file `eclipse.ini` in that directory. The original looks more or less like:

```
-showsplash
org.eclipse.platform
-vmargs
-Dosgi.requiredJavaVersion=1.5
-Xms40m
-Xmx256m
-XX:MaxPermSize=256m
```

Change the line `-Xmx256m` with `-Xmx1024m` and add the line

```
-Duser.name=<your name>
```

(replacing `with your name`). Just before the line `-vmargs`, add (if not present)

```
--launcher.XXMaxPermSize
256m
-vm
<path/to/preferred/java>
```

The value of `<path/to/preferred/java>` is system dependent. If the version of java that comes with the system is good enough (on Win32 is the one you installed), you can skip that line, otherwise you have to specify the path to the java executable (see <http://wiki.eclipse.org/Eclipse.ini> for more information). Note: the preferred JRE, can also be selected creating a symlink called `jre` in the eclipse directory pointing to the actual jre location, thus making the `-vm` part of `eclipse.ini` unnecessary.

At the end you should have something like (that's mine):

```
-showsplash
org.eclipse.platform
--launcher.XXMaxPermSize
256m
-vm
/scratch/marcocle/contrib/jre/bin/java
-vmargs
-Dosgi.requiredJavaVersion=1.5
-Xms128m
-Xmx1200m
-Duser.name=Marco Clemencic
```

Now you are ready to start Eclipse: go to the `eclipse` directory and launch `./eclipse` (there is no need to go there, you can also use the full path to the `eclipse` executable).

Probably, the first thing you want to do is to update the pre-bundled plugins. Select the menu "Help->Software Updates...", select the tab "Installed Software" and click on the button "Update...". Sometimes there are problems with some updates, but usually it is enough to remove and re-add the corresponding update site.

Extensions

There are a few Eclipse Update Sites that should be added to Eclipse. They are listed in the following sections, but, for commodity, I attached an XML file that can be imported in the configuration page for the management of size to add all of them:

https://twiki.cern.ch/twiki/pub/LHCb/EclipseConfiguration/update_sites.xml

Subversion

To be able to work with a Subversion (SVN) repository, you need to install a "Team Provider" and a "Connector".

There are two possible "Team Provider" plugins:

- [Subclipse](#)
- [Subversive](#)

I'm not going to discuss which is best, I choose Subversive because it seems better to me.

Add the following update site to your list:

Subversive SVN Connectors Site

<http://www.polarion.org/projects/subversive/download/eclipse/2.0/update-site/>

Then select the components:

Eclipse

- ◇ Subversive SVN Team Provider
- ◇ Subversive SVN Integration for the Mylyn Project

Subversive SVN Connectors Site

- ◇ Subversive SVN Connectors
- ◇ SVNKit 1.2.2 Implementation

Python

To develop Python code, you need [PyDev](#).

Add the Pydev Update Site to the list:

- <http://pydev.org/updates>

and install "Pydev for Eclipse" and "Pydev Mylyn Integration".

If interested, you may also install the "Pydev Extensions" (*non-free*, read carefully the license agreement when you install it).

Mylyn Web Templates Connector

There is a very interesting Connector in the the incubator repository of Mylyn: Web Templates [?](#). It allows (a basic) usage of savannah from Eclipse (once configured).

Add the update site <http://download.eclipse.org/tools/mylyn/update/incubator> [?](#) and install the plugin "Mylyn Connector: Web Templates".

HTML/XML (optional)

The plug-ins to work with XML and HTML are in the main Eclipse Update Site, so there is no need to add a new one.

Plugins to install (with their dependencies):

Eclipse XML Editor and Tools
 collection of plugins to work with XML files.
 Web Developer Tools
 collection of plugins to work with web pages.

It may also be interesting to install the "XSL Editor and Tools" plugin, for working with XML Style Sheets.

Color Editor (optional)

The Color Editor [?](#) plug-in does not have an update site, but it easy enough to install anyway.

Download the latest .jar file from the download page [?](#) (currently it is http://gstaff.org/colorEditor/cbg.editor_1.2.6.jar [?](#)) and copy it in the directory `eclipse/plugins`, then start/re-start eclipse.

I attached to this page a patched version of the .jar file with support for the file `project.cmt` and `requirements`. The only problem is that the association of the file types to the editor have to be added by hand, via "Window->Preferences->General->Editors->File Associations" (the name of the editor is "Syntax Coloring Editor").

Configuration

Enter the preferences dialog from the menu bar "Window->Preferences".

Note the the preferences are workspace specific, so you will need to re-apply them to every new workspace (you can also make a copy of an empty workspace as a template).

General

Editors

Text Editors

set "Displayed tab width" to 8, enable "Show print margin" (optional)

C/C++

Build Console

increase the number of lines for the output (e.g. 8192)

Code Style

a predefined Code Style file for LHCb is attached to this page, you just have to download it and import it: https://twiki.cern.ch/twiki/pub/LHCb/EclipseConfiguration/LHCb_C++_CodeStyle.xml

Editor

Folding

Enable all folding types, select only "Macros" and "Inactive Preprocessor Branches" for initial folding (uncheck the others).

File Types

add *.icc and *.icpp as "C++ Source File"

Task Tags

add `FIXME` with priority "High"

Pydev

Interpreter - Python

add (at least) a python interpreter, probably you want to add the one you find with `which python` (which is the one that LbLogin adds to the path)

Team

Ignored Resources

add the following patterns to the list

```
◇ *-slc5-*
◇ slc4_*_gcc*
◇ win32_vc71_dbg
◇ osx104_*_gcc*
◇ setup.*
◇ cleanup.*
◇ version.cmt
◇ *.*make
◇ *.*makesav
◇ Makefile
◇ NMake
◇ install.history
◇ gdd.dtd
◇ genConf
◇ QMTest
◇ *.launch
```

Customization

Savannah

The "Web Templates" connector allows the connection of the Mylyn feature to the savannah repository.

These are the steps needed to add "Gaudi Patches"

- Open the "Task Repository" view (from the dialog "Window->Show View->Other")
- Click on the "Add Task Repository" button (or the entry in the context menu of the view)
- Select "Web Template", then "Next"
- Fill the form with the values:

Server

```
https://savannah.cern.ch/patch/?group=gaudi
```

Label

```
Gaudi Patches
```

Additional Settings

```
add the parameters group -> gaudi and tracker -> patch
```

Advanced Configuration

EclipseConfiguration < LHCb < TWiki

Task URL

`https://savannah.cern.ch/${tracker}/?`

New Task URL

`${serverUrl}&func=additem`

Query Request URL

`${serverUrl}&chunksz=150&report_id=100`

Query Pattern

`<tr.*?prior[a-z]({Status}closed)?.*?#({Id}[0-9]+).*/td>\n<td ><a.*?>`

- Click on "Finish" and say "Yes" when asked to create a new query on the created repository.
- Accept the defaults in the form for the new query.
- Open the "Task List" view from the dialog "Window->Show View->Other".

Other repositories can be created using the same "Advanced Configuration" and changing "Server" and "Additional Settings" (reminder: the tracker is one of `patch`, `task` or `bugs`).

It is possible to create advanced queries (e.g. bugs assigned to somebody)... TODO

Note: This is not the only way to add a Task Repository for Savannah, but it is the best I could find. One can use a single "repository" for all the trackers, but then it is not possible to create new tasks from Eclipse.

-- MarcoClemencic - 2009-09-27

This topic: LHCb > EclipseConfiguration

Topic revision: r26 - 2012-07-12 - MarcoClemencic



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback