ECLIPSE PTP

BASIC USER MANUAL

This manual has instructions on installing Eclipse PTP on Linux operating systems and using Eclipse PTP for developing and running parallel applications on supercomputers. Installation procedures for OS X and Windows OS are similar but may differ. Please, refer to the online manual at https://wiki.eclipse.org/PTP/release_notes/9.0#Install_PTP.
Illustrations are made from a fresh installation of Eclipse PTP on Ubuntu 14.04.
This manual describes work with Fujitsu supercomputers FX10 and K, but can be useful for other supercomputers also.
Symbol  shows mouse click point.

Contents

1. Installation and configuration *2*

1.1 Start Eclipse *3*

1.2 Add a Remote system connection *3*

1.2.1. Add SSH key *4*

2. Create synchronised project *5*

2.1 From code on remote location *5*

2.1.1. Synchronization *6*

2.1.2. Remote terminal *6*

2.2 From local code *6*

2.2.1. Import Fortran code *6*

2.2.2. Convert local project to synchronized *8*

2.3 Create synchronized project from git repository *10*

3. Build a project *13*

3.1 Build Configurations *13*

4. Run a project *14*

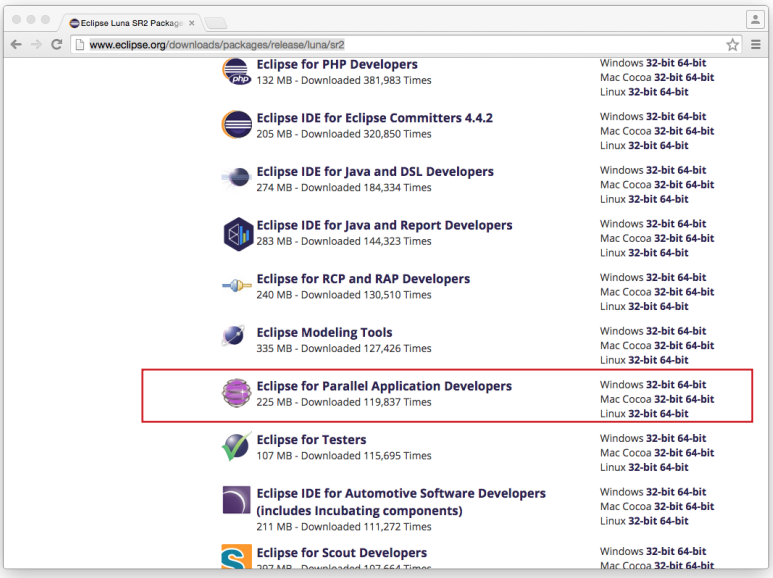
4.1 Run Configurations *14*

4.1.1. Import Target System Configurations *14*

4.1.2. Create Run Configuration *16*

1. Installation and configuration

Download Eclipse PTP distribution. We recommend using Luna version as it is more stable.
Go to http://www.eclipse.org/downloads/packages/release/luna/sr2, scroll down to Eclipse for Parallel Application Developers and select your platform:



Download archive and extract:

```
$ tar -zxvf  eclipse-parallel-luna-SR1a-linux-gtk-x86_64.tar.gz
$ sudo mv eclipse /opt
```

Create Link

```
ln -s /opt/eclipse/eclipse /usr/sbin/eclipse
```

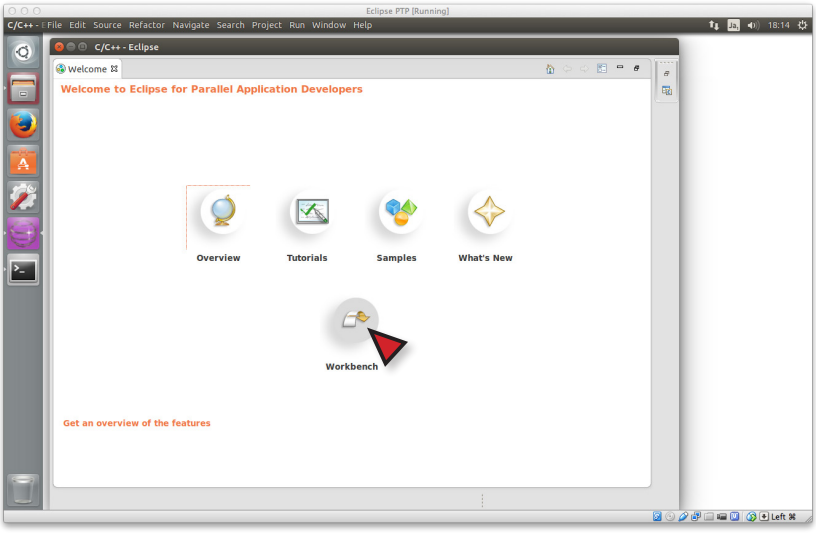
Install Java and git

```
$ sudo apt-get install default-jre
$ sudo apt-get install git
$ git config --global user.name "Eclipse User"
$ git config --global user.email "user@eclipse.ptp"
$ git config -l
user.name=Eclipse User
user.email=user@eclipse.ptp
```

Enter your name and e-mail address here

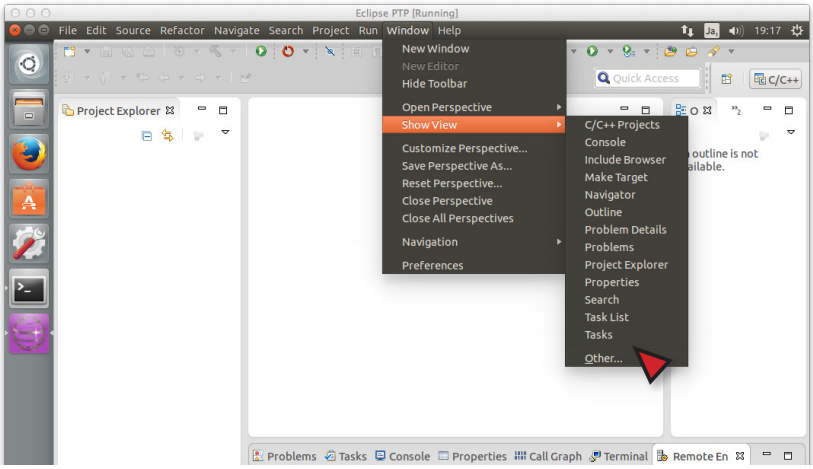
1.1 Start Eclipse

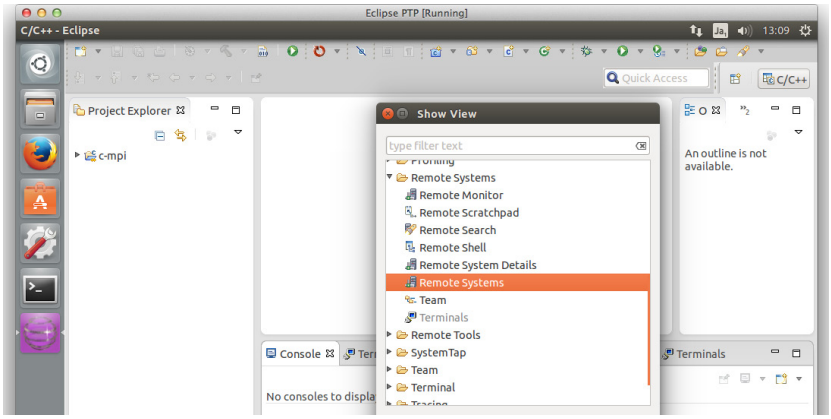
Start Eclipse PTP from extracted directory.
Set Workspace directory.
Eclipse window will show up.



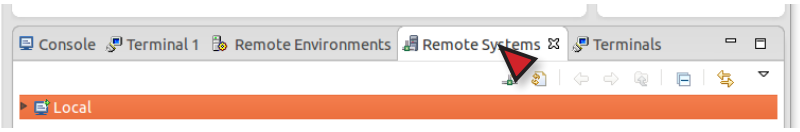
1.2 Add a Remote system connection

Open Remote Systems view:
Window > Show View > Other > Remote Systems

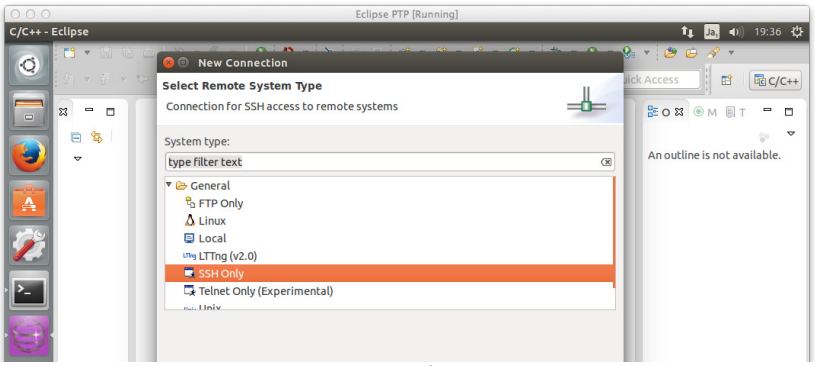




Click on  to add a connection to remote system.



Select SSH Only.

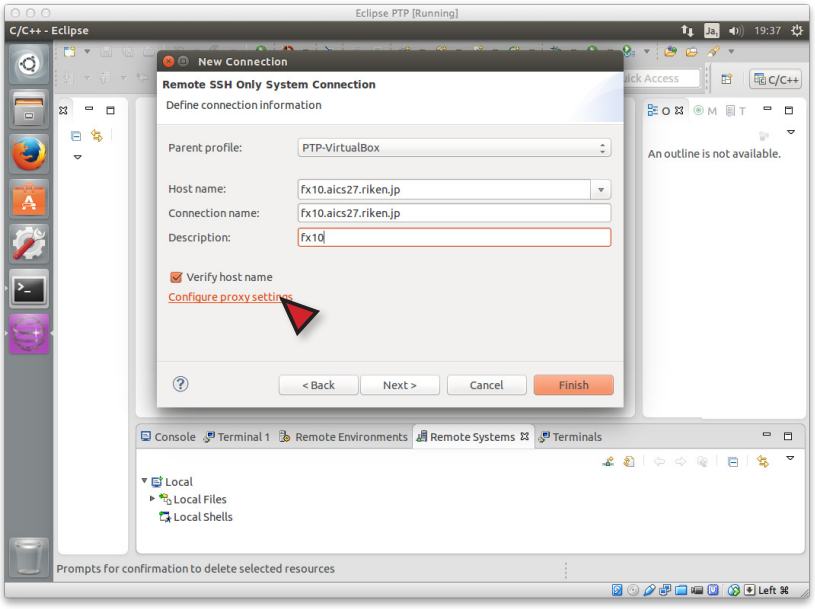


Connection name and Description are arbitrary

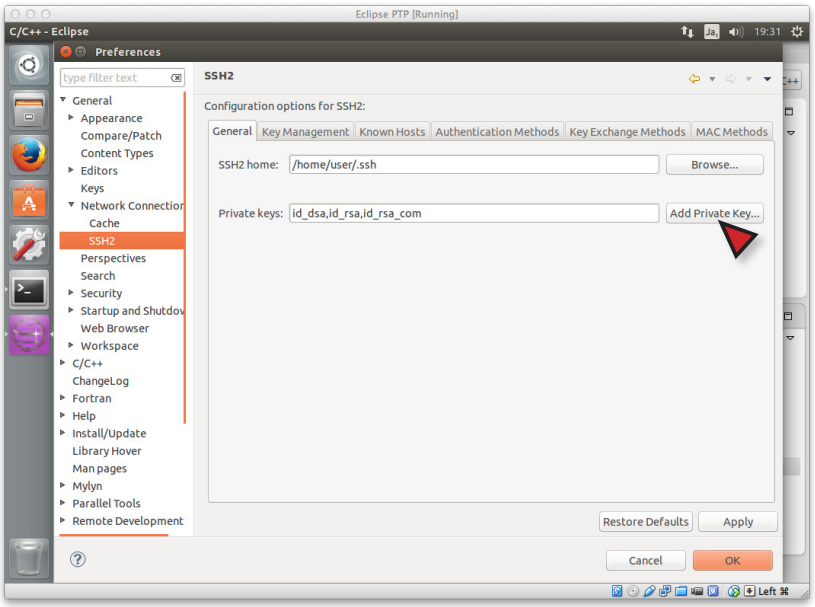
Enter Host name, Connection name and Description.

1.2.1. Add SSH key

Click Configure proxy settings.



Unfold Network connection, select SSH2.
Click Add Private Key and select file with the SSH key you use.

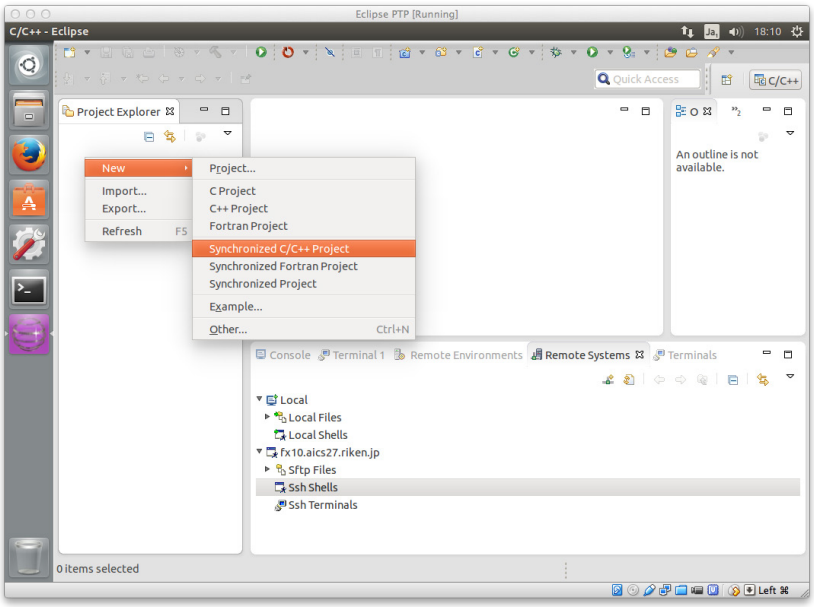


After you added SSH key, click OK, then click Finish. New Remote System will be added.

2. Create synchronized project

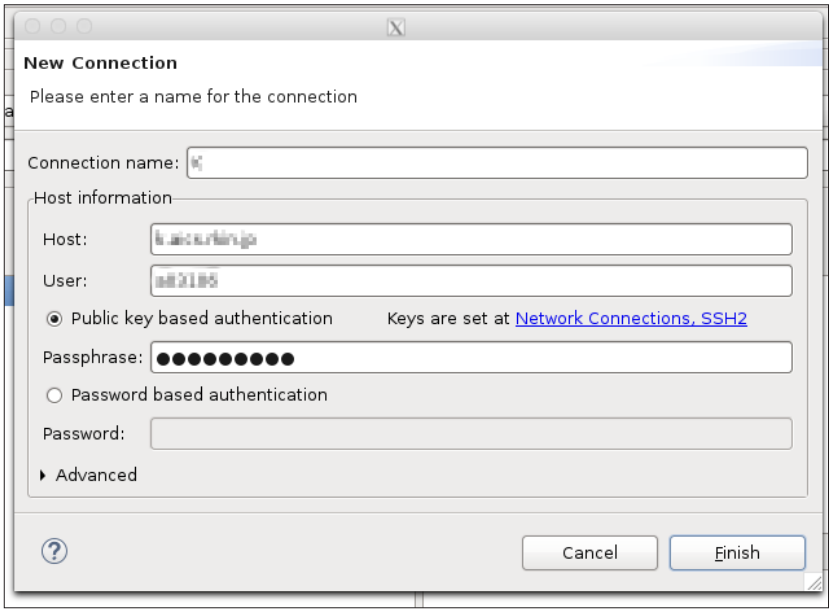
2.1 From code on remote location

In the Project Explorer, right-click on empty space, select New > Synchronized C/C++ Project or Synchronized Fortran project.

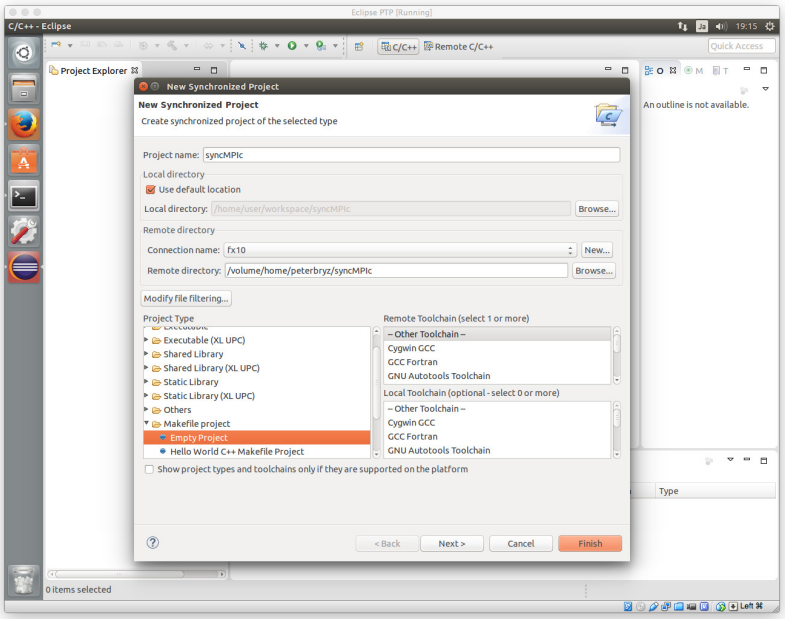


Enter Project name. Select Connection name (create new one if necessary).

New connection dialog



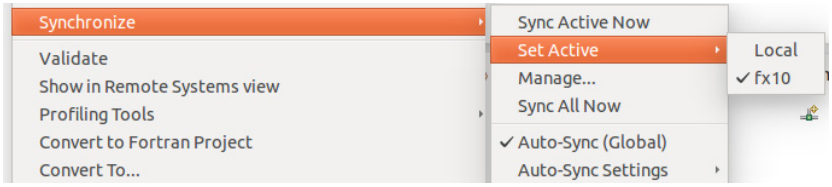
Set Remote directory, Project Type: Makefile project/Empty Project. Click Finish.



Your new project will appear in Project Explorer.
Note your project icon. Synchronized project icon has a bidirectional arrow sign: ▾ c-mpi

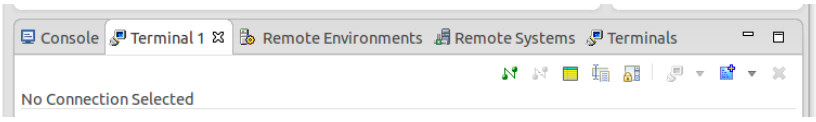
2.1.1. Synchronization

To synchronize local and remote versions of a synchronized project right click on the project in Project Explorer, select Synchronize and Sync Active Now. You can also select a project and press button on the toolbar.
To change connection used for synchronization, select one under Set Active menu.



2.1.2. Remote terminal

To launch terminal: in Terminal1 view, click New Terminal Connection icon:

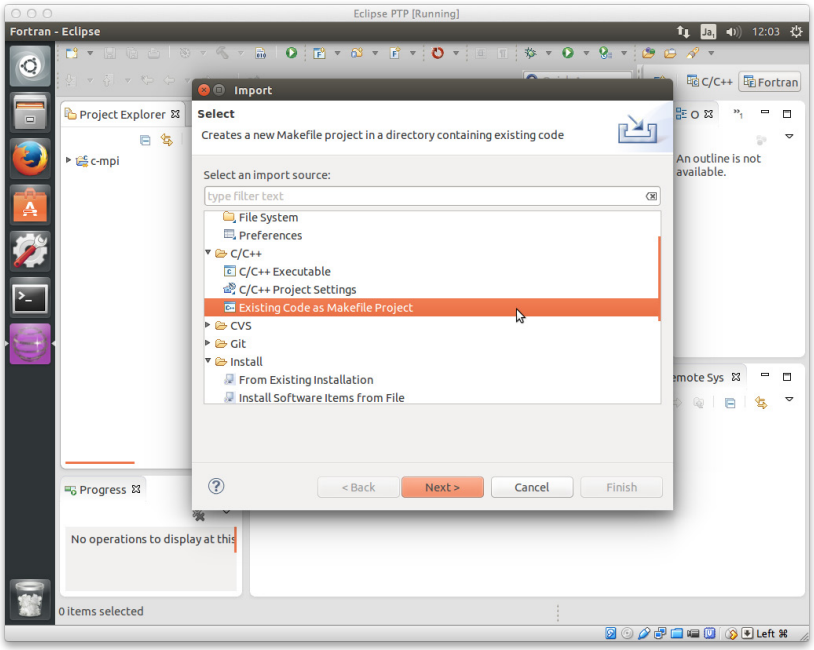


2.2 From local code

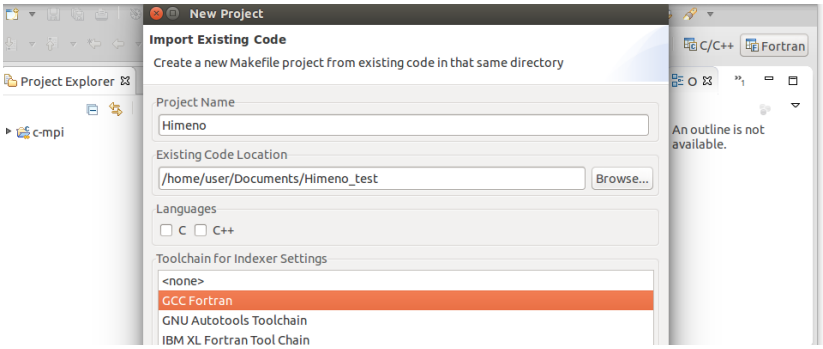
2.2.1. Import Fortran code

Local code is any C/C++/Fortran program source code that is not part of any Eclipse project.
To convert local source code to Eclipse project right-click on empty space in Project Explorer and select Import...

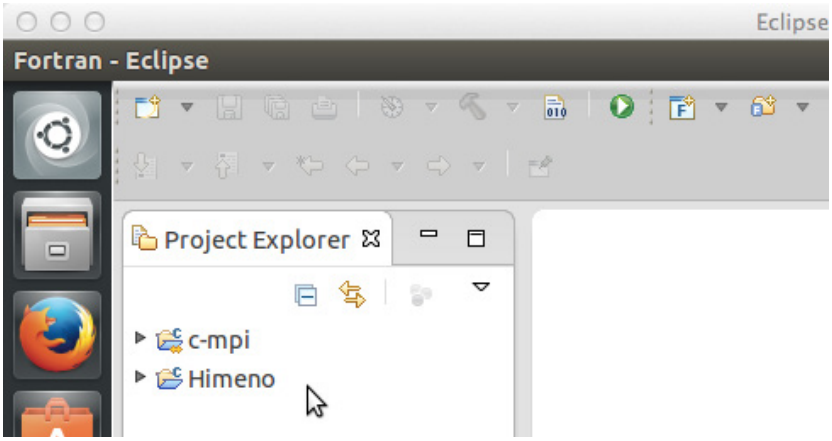
Unfold C/C++ and select Existing Code as Makefile Projects.



Click Next.
Give your project a name, select directory with your code, and select Toolchain.

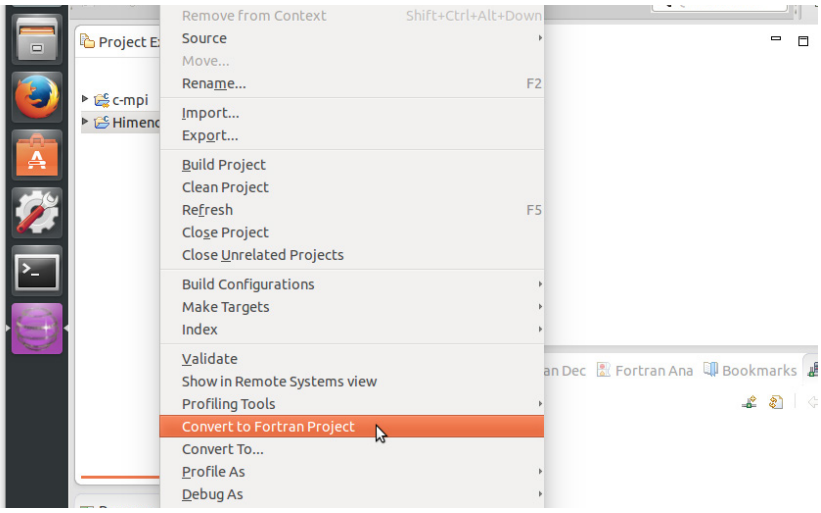


Click Finish.



Now you have a new project in Eclipse, but the project is not synchronized and marked as a C project.

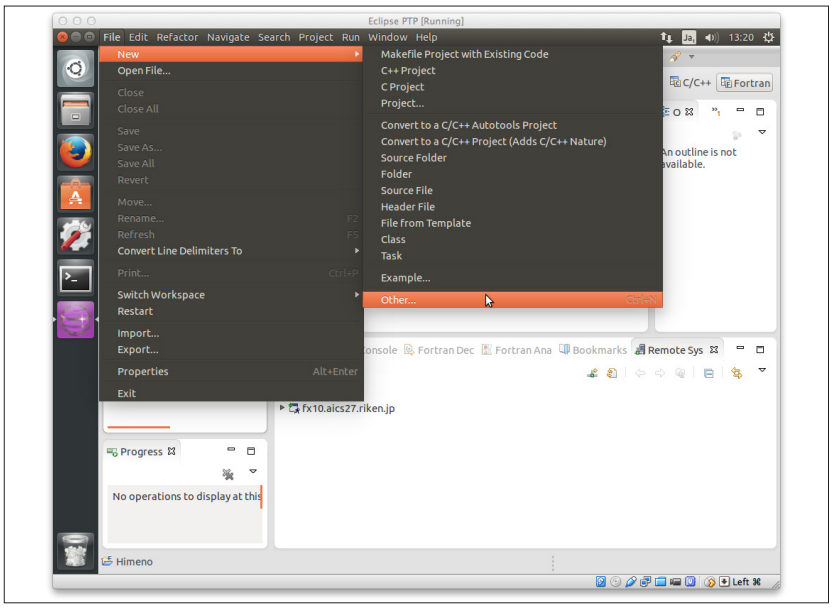
Convert it into a Fortran project. Right-click on the project in Project Explorer and select Convert to Fortran project.



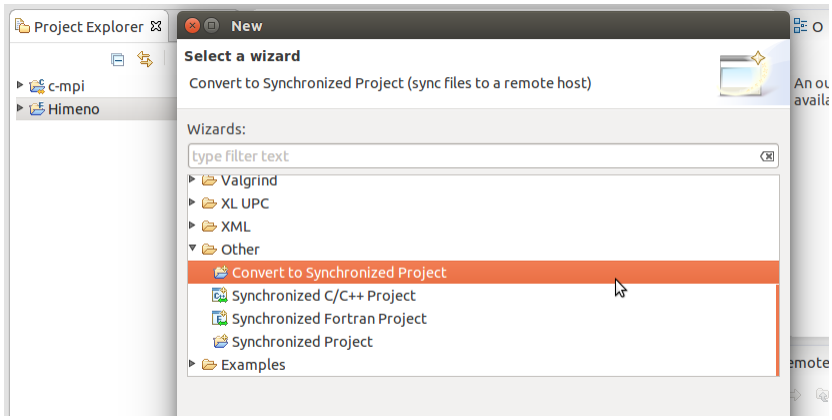
Make sure, the project folder icon changed to  . There is no arrow, because the project is not synchronized (local).

2.2.2. Convert local project to synchronized

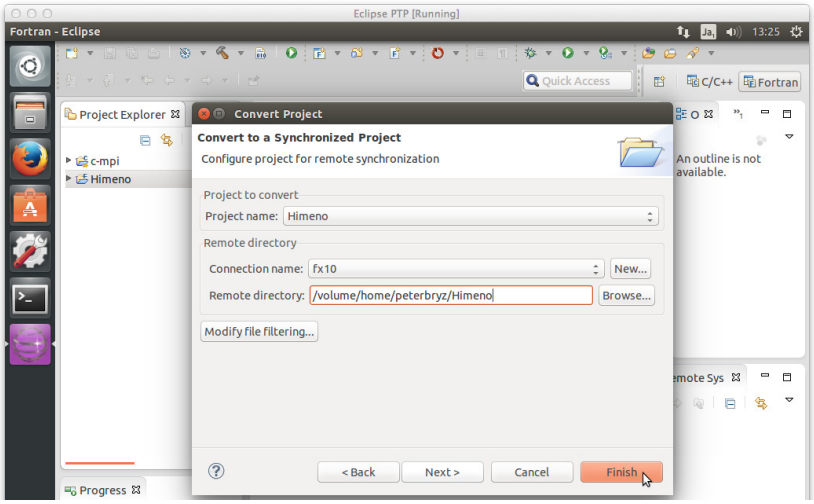
To convert project to synchronized, go to top menu, select File > New > Other ...



Unfold Other and select Convert to Synchronized project.



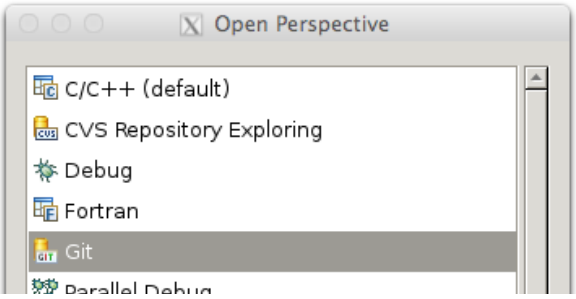
Select your project in Project Name, select connection and remote directory. Change filtering settings if necessary.
Click Finish.



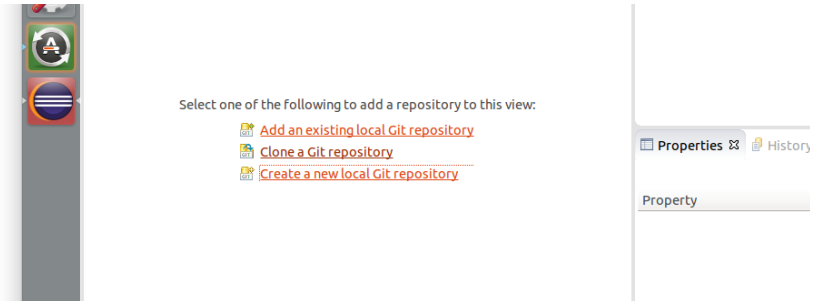
Note, that project icon changed to  .

2.3 Create synchronized project from git repository

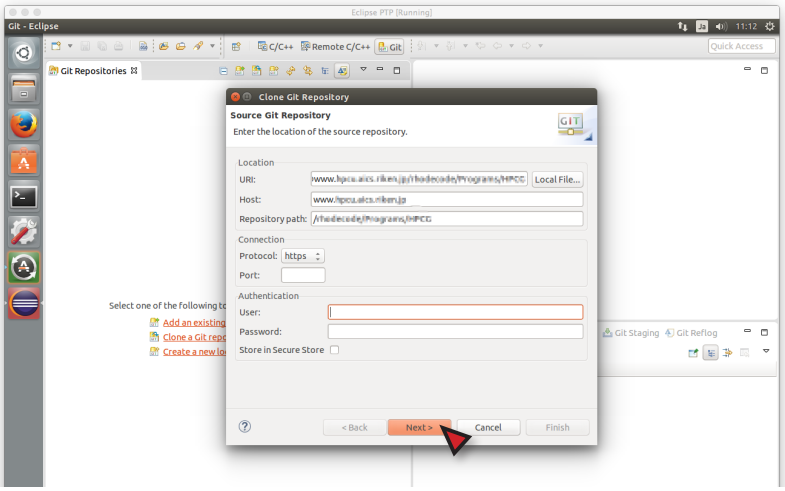
To create a new synchronized project from existing remote git repository: Open git perspective from top menu: Window > Open perspective > Other and select Git.



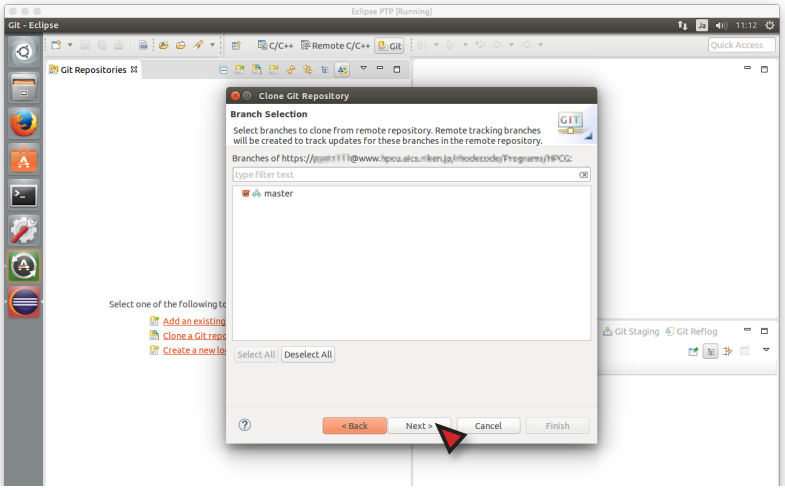
In Git perceptive select Clone a Git repository.



Fill in repository URI and other fields (some fields will be filled automatically after you fill URI). Click Next.



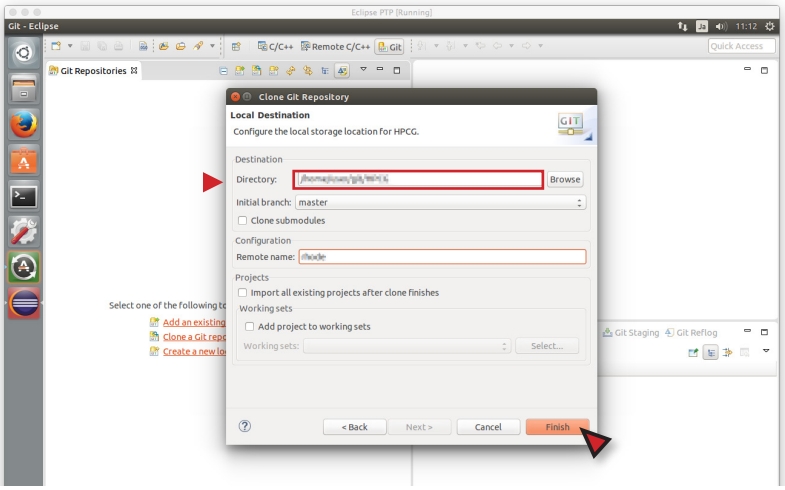
Select git branch and click Next.



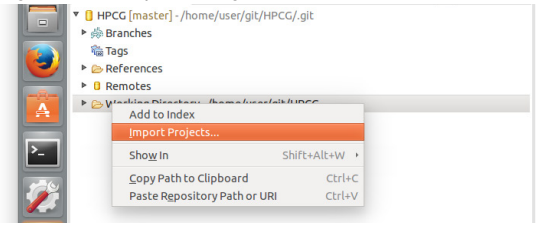
Note! *It is not recommended to make git repositories from Eclipse projects and clone them on another machine. Project settings from one machine are likely to cause problems on another.*

Select Destination Directory. This will be the directory of your project, so we recommend to select one inside Eclipse workspace folder. Also note, that directory name will be the project name.
Remember Destination Directory. Click Finish.

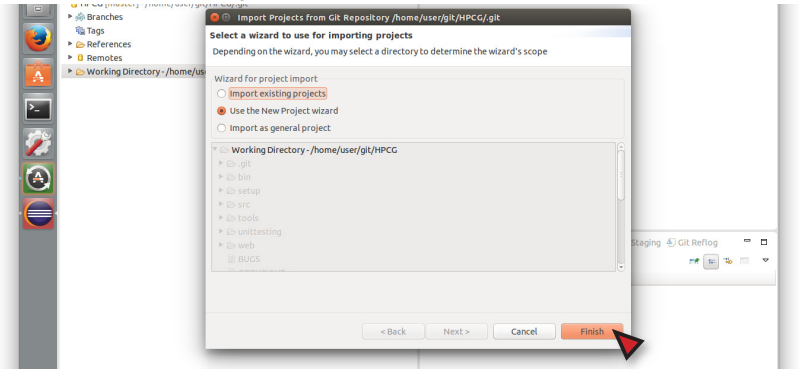
Remeber this path, you will need it later



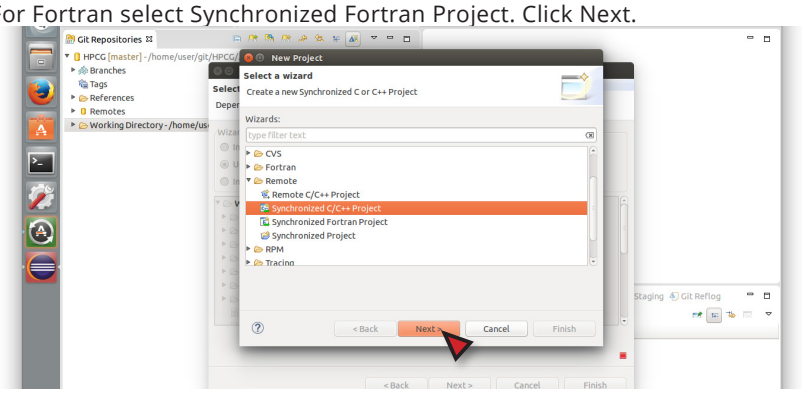
Repository is cloned.
Open git perspective and right-click Working Directory of your new repository. Select Import Projects...



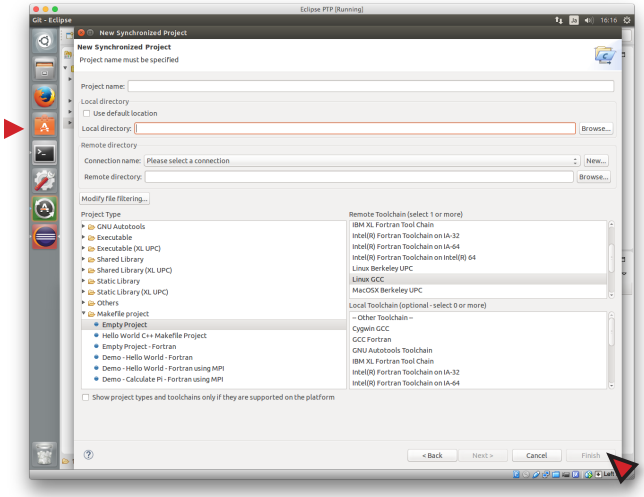
Select Use the New Project wizard and click Finish.



In case of C project select Synchronized C/C++ Project in Remote section. For Fortran select Synchronized Fortran Project. Click Next.



Uncheck Use default location. Select local repository location instead.
Set the project name to the directory name. For example, if you cloned repository into /home/user/workspace/my_project directory, you project name will be “my_project”.
Select connection. Remote location should be filled automatically now. You can change it to whatever remote path you like.
Select project type (for example Empty Project). Click Finish.



Your new project now should be visible in Project Explorer (C/C++ or Remote C/C++ perspective).

In the same way **Synchronized Fortran Project** can also be created.

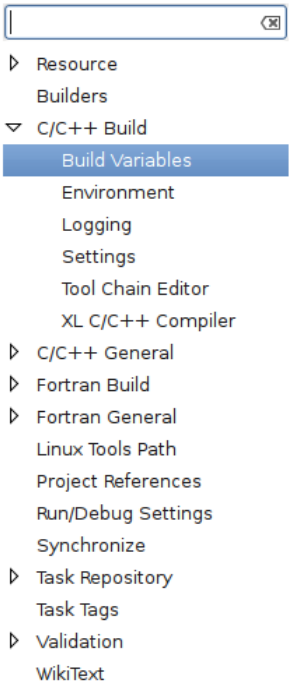
3. Build a project

To build a project you need to provide necessary parameters in a Build Configuration. Projects can be *Makefile-based* or *Managed*.
Makefile-based project contains its own build command – a makefile or build script. For Managed projects Eclipse manages build process, no makefile required by the user.
A Build configuration provides the necessary information to build the project. The build configuration information is specified in the project properties. Projects can have multiple build configurations.

3.1 Build Configurations

Open project properties by right-clicking on the project name in the Project Explorer view and selecting Properties. Or you can select the project select from menu File > Properties.

Note: Fortran projects are a superset of C/C++ projects, so they have properties for both.



On the left you can see Project Properties menu.

C/C++ Build
Main properties page. Configure the build command.
Build Variables
Create/manage variables that can be used in other build configuration pages.
Environment
Modify/add environment variables passed to build.
Logging
Enable/disable build logging.
Settings
Binary parser selection (used to display binaries in Project Explorer).
Error parser selection (used to parse the output from compiler commands)
Tool Chain settings (managed projects only) .
Tool Chain Editor
Allows the tools in a particular tool chain to be modified.
XL C/C++ Compiler
Compiler settings for XL C/C++ compilers (if installed).
C/C++ General/Preprocessor Include Paths...
Set include paths here.

The active build configuration will be used when the build button  is selected. The Build Configurations project context menu can be used to change the active configuration. Right click on project, then select the build configuration from the Build Configurations > Set Active menu.

4. Run a project

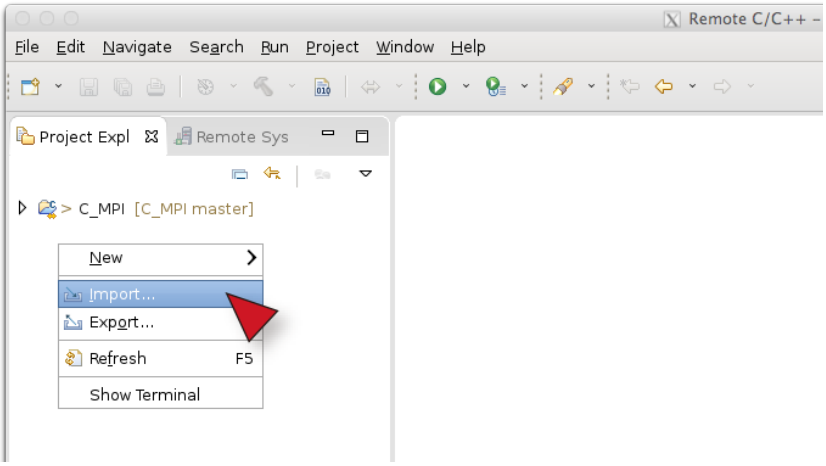
Before running a project it is necessary to set up at least one Run configuration. Run Configurations are used to define various parameters of running a Synchronized project as a parallel application on a remote system.

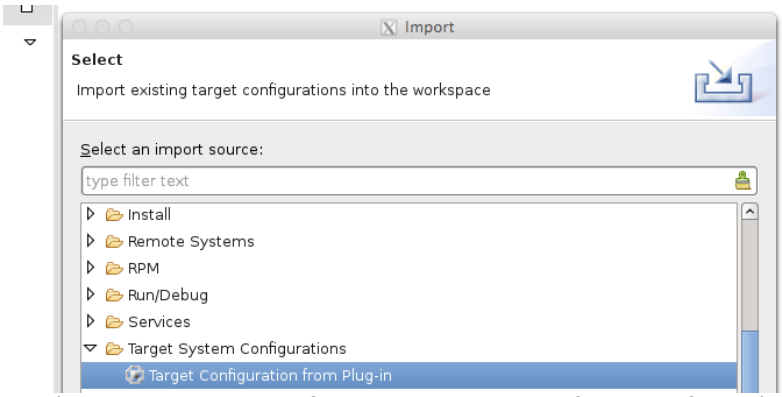
4.1 Run Configurations

Run configurations are dependent on the target system. For K, FX10 and other systems with Fujitsu “Parallelnavi” job scheduler you have to use PJM-* Target System Configurations (TSC) created by HPC Usability Research Team of AICS RIKEN. See below for instructions on how to install these TSCs.

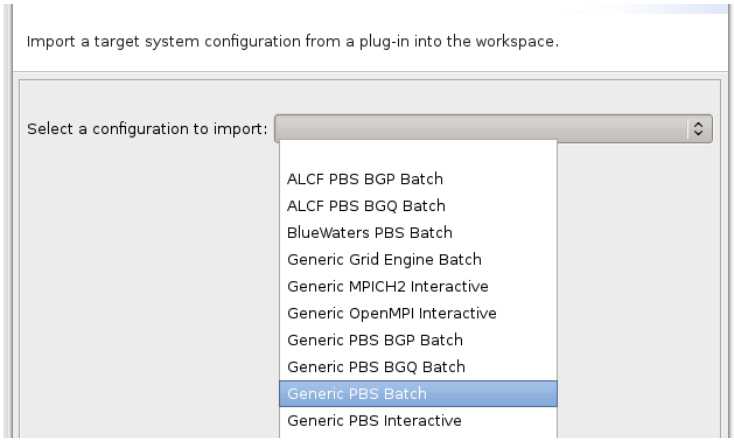
4.1.1. Import Target System Configurations

To use PJM-* (and other custom) Target System Configurations you need to create targetConfigurations project in the following way:
1. Right-click empty space of Project Explorer window and select Import, or select File > Import menu.

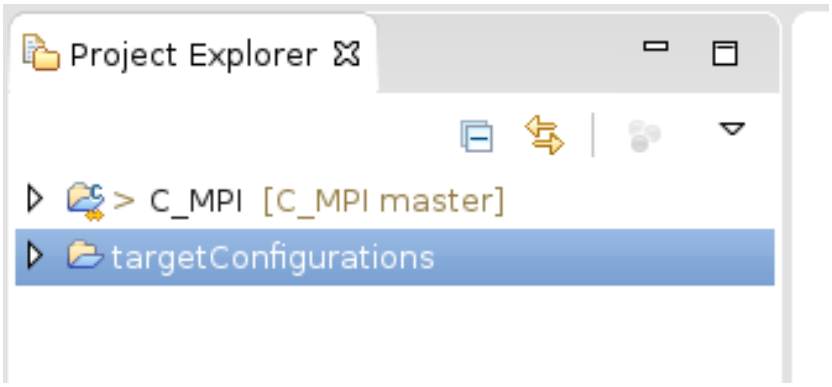




2. Select Target System Configurations > Target Configuration from Plug-in.
3. Click Next and select any configuration from drop-down list. Click Finish.



4. In dialog press “Yes” and you will see a new project in Project Explorer.



5. Find directory of targetConfigurations project in your Eclipse workspace directory on your disk. Clone github.com/pyotr777/EclipsePTP_Parallelnavi_TSC repository into this directory with the following commands:
remove all files

```
> rm *
```

move .project file to parent folder

```
> mv .project ..
```

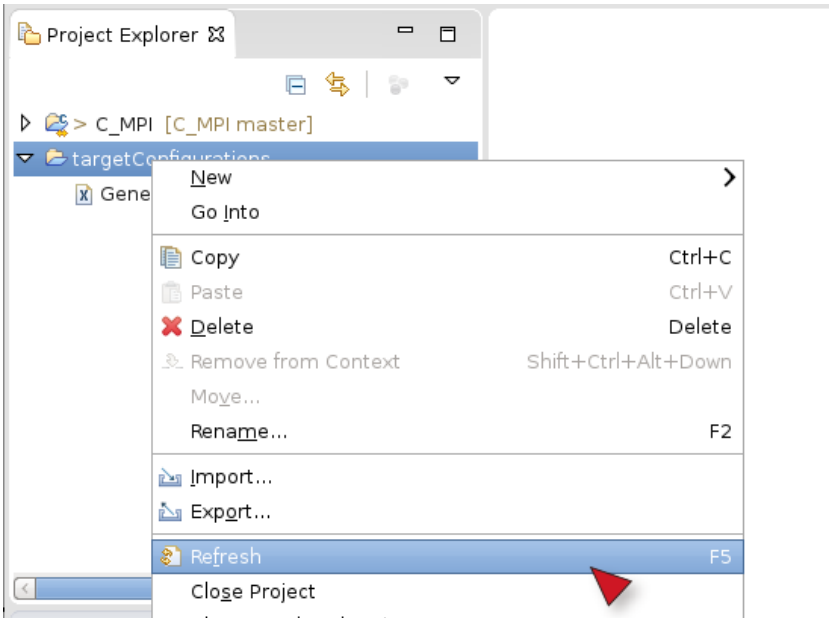
clone git repository (don’t forget the last dot)

```
> git clone https://github.com/pyotr777/EclipsePTP_Parallelnavi_TSC.git .
```

move .project file back

```
> mv ../.project ./
```

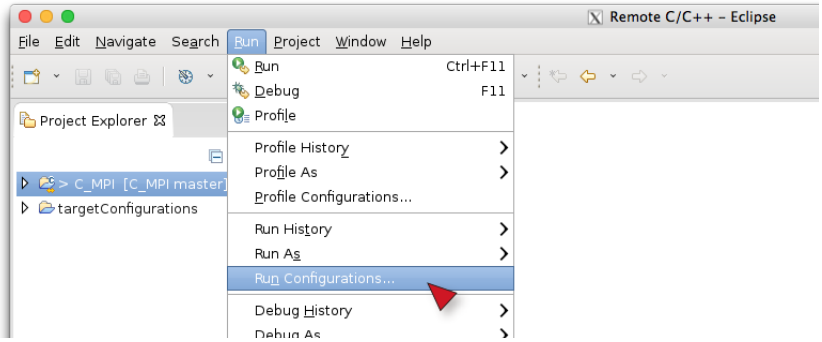
6. Return to Eclipse, right-click targetConfigurations in Project Explorer and select Refresh.



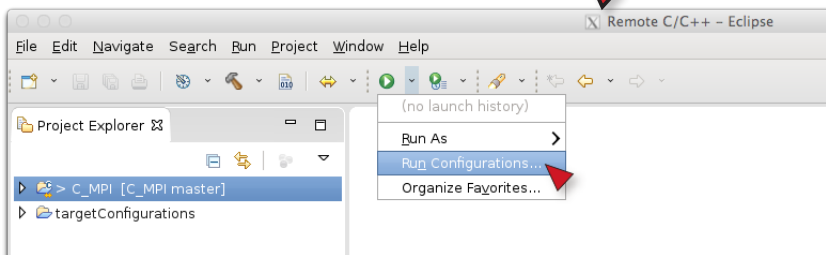
Now in Project Explorer you should be able to see PJM-* files in targetConfigurations project.

4.1.2. Create Run Configuration

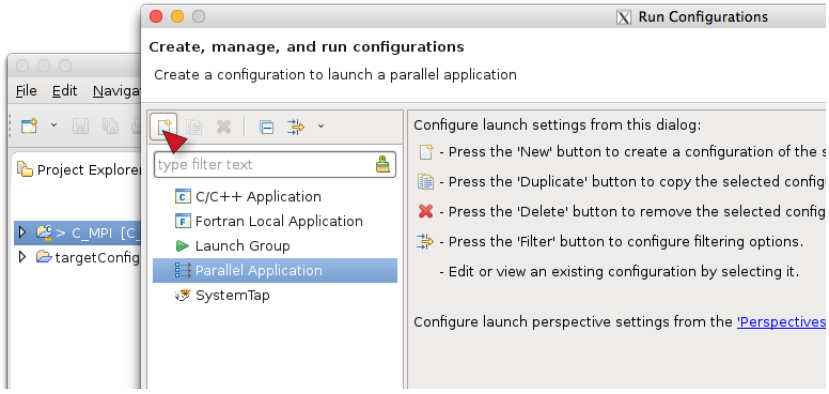
1. Select your project in Project Explorer and select Run>Run Configurations... menu



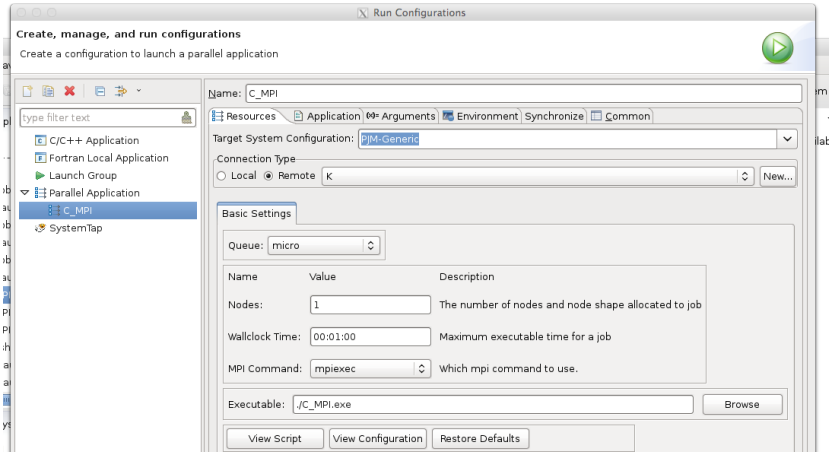
or click the arrow mark next to the Run button  on the toolbar.



2. Select Parallel Application and click New button.



3. Give your new configuration a name, select Target System Configuration, Connection to your Remote system (the one you created at 2.1) or create a new connection.
On Basic Settings tub select Queue, number of nodes, estimated job run time (Wallclock time), MPI command (mpiexec) and select executable file from your project.
Executable field should not include absolute path!



4. On Application tab set Application program to /usr/bin/pjsub.
5. On Arguments tab set Working directory to the absolute path of your project on remote system with Browse button.
6. Define program arguments on this tab and environment variables on Environment tabs, if necessary.
7. Click Run button.

A dialog will appear, asking if you want to switch to Monitoring view. Monitoring view present a graphical representation of compute node and a list of running and waiting jobs. On FX10 all jobs are visible, where as on the K computer only user's own jobs will be displayed. Beware, that on K computer it takes about one minute or more before compute nodes scheme will be rendered in Monitoring view.

Sample monitoring screen for the K conputer