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 to the one described here 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.

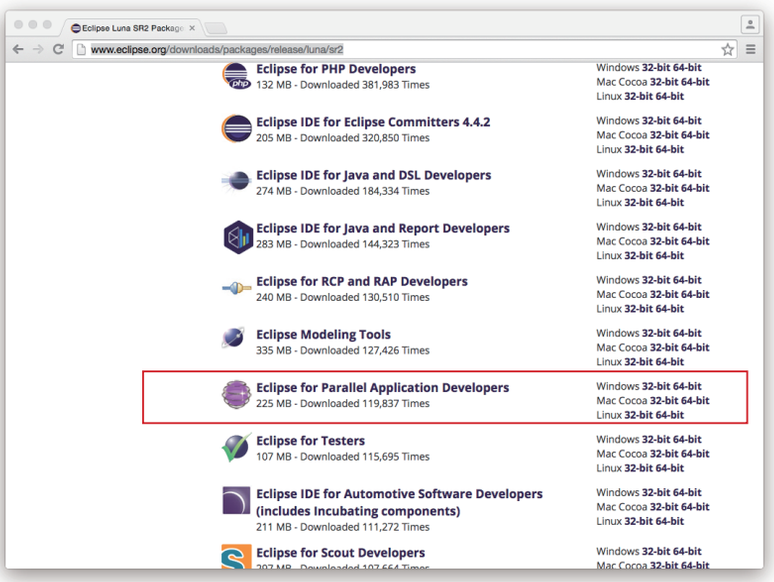
Contents

1. Installation and configuration

1.1 Download and install Eclipse PTP

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 to Eclipse PTP executable

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

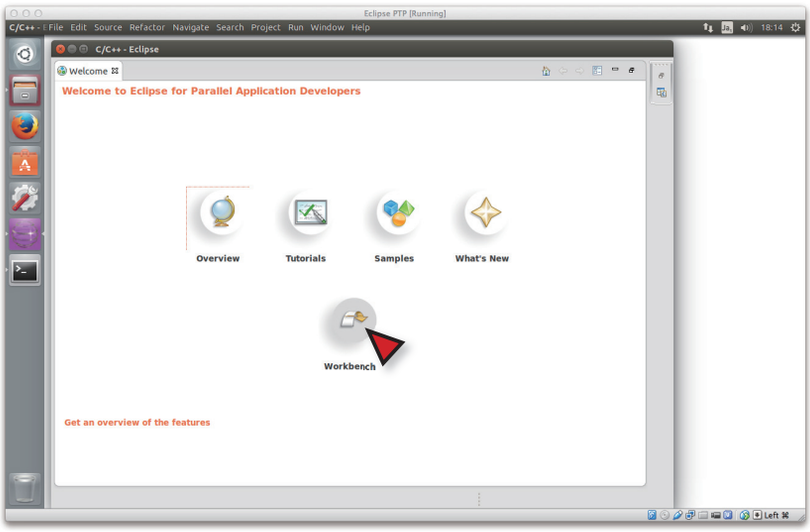
Install Java and git

```
$ 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

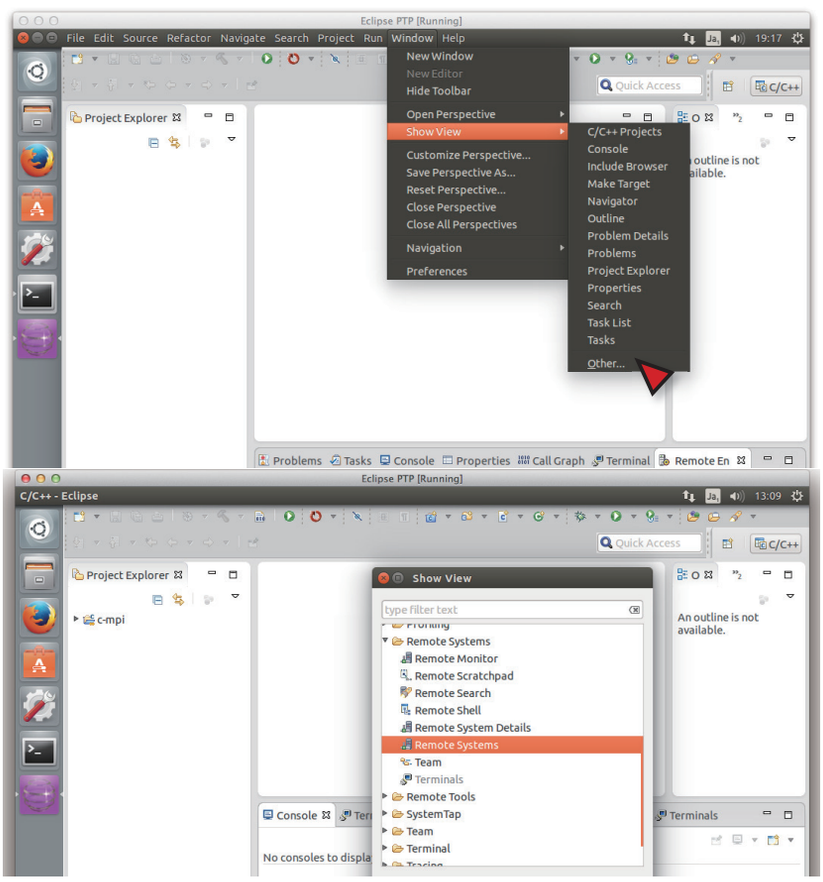
1.2 Start Eclipse PTP

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

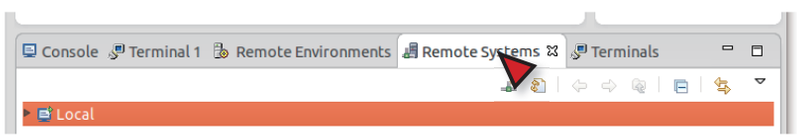


1.3 Add a Remote system connection

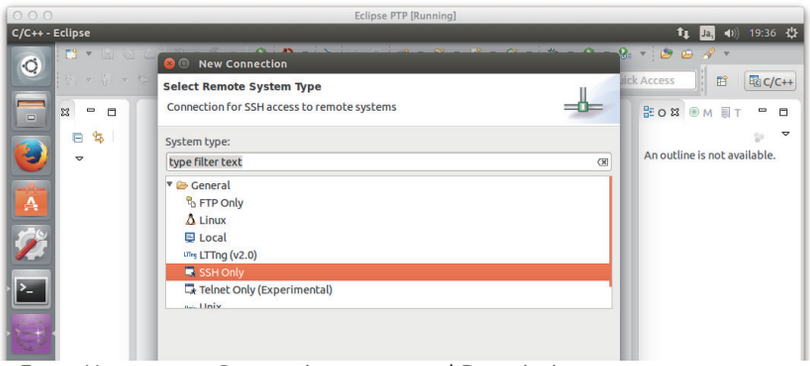
Open Remote Systems view:
Window > Show View > Other > Remote Systems



Click on  to add a connection to remote system.



Select SSH Only.

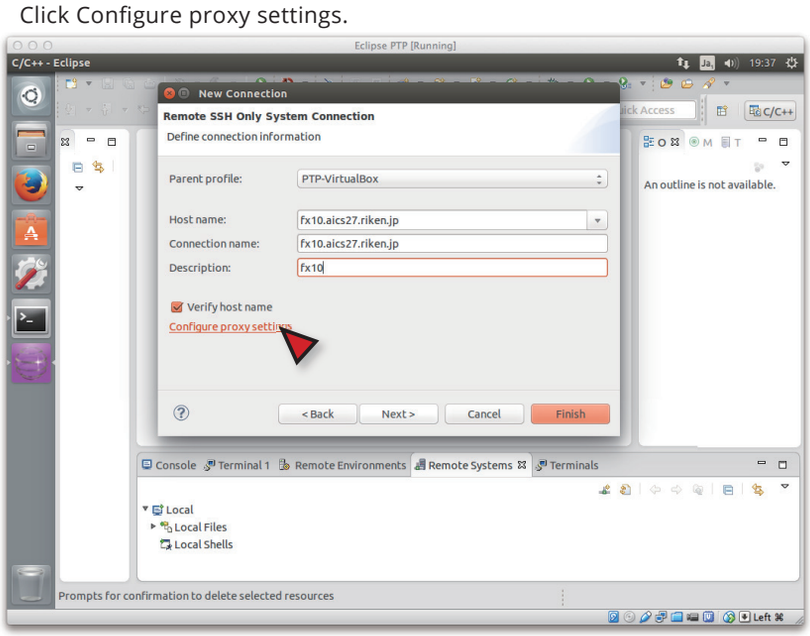


Connection name and Description are arbitrary

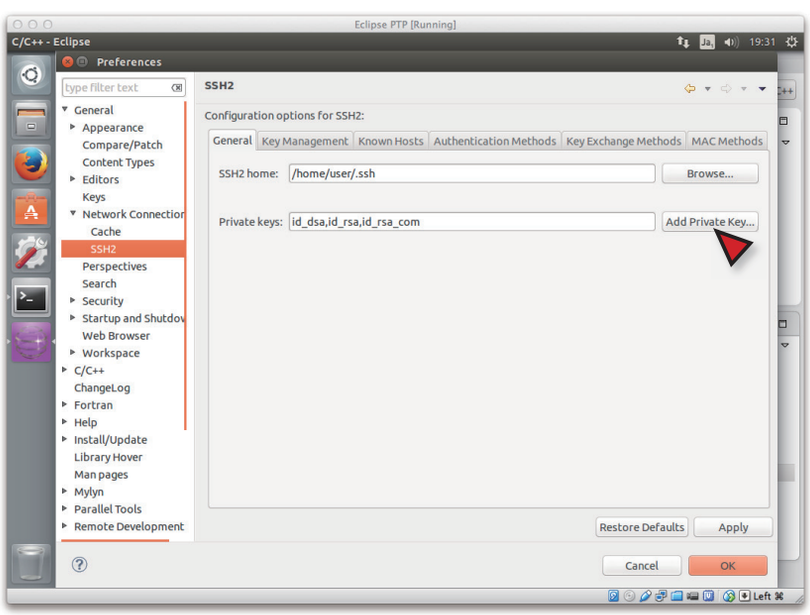
Enter Host name, Connection name and Description.

1.3.1. 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.

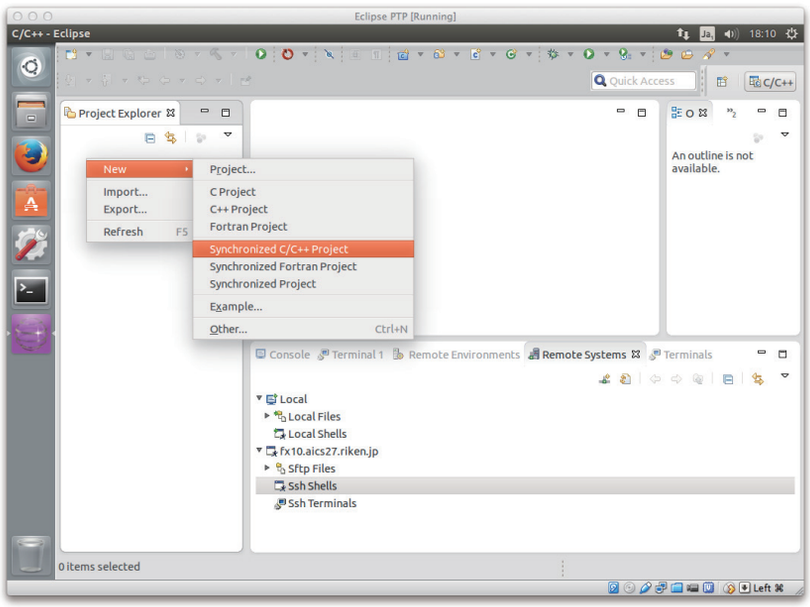


After you added SSH key, click OK, then click Finish. New Remote System will be added.

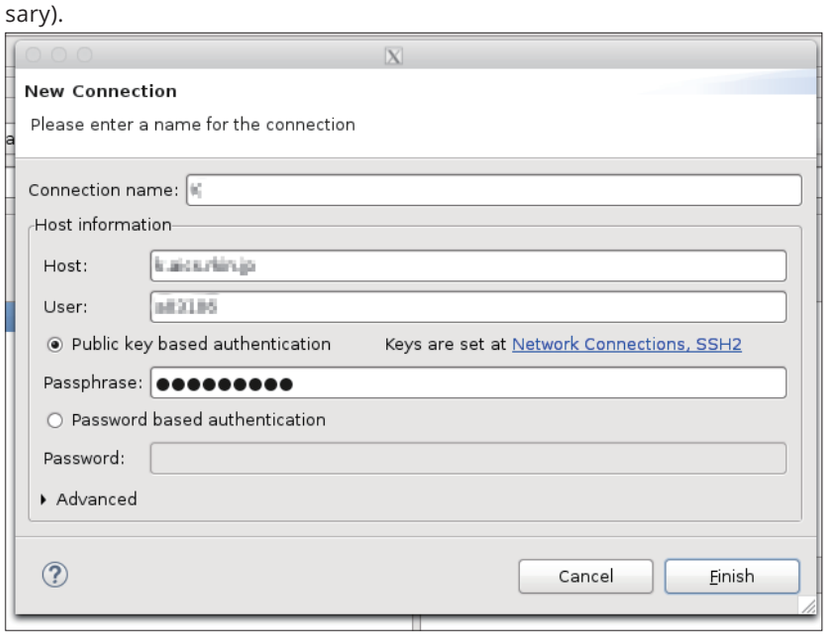
2. Create synchronized project

2.1 From code on remote location

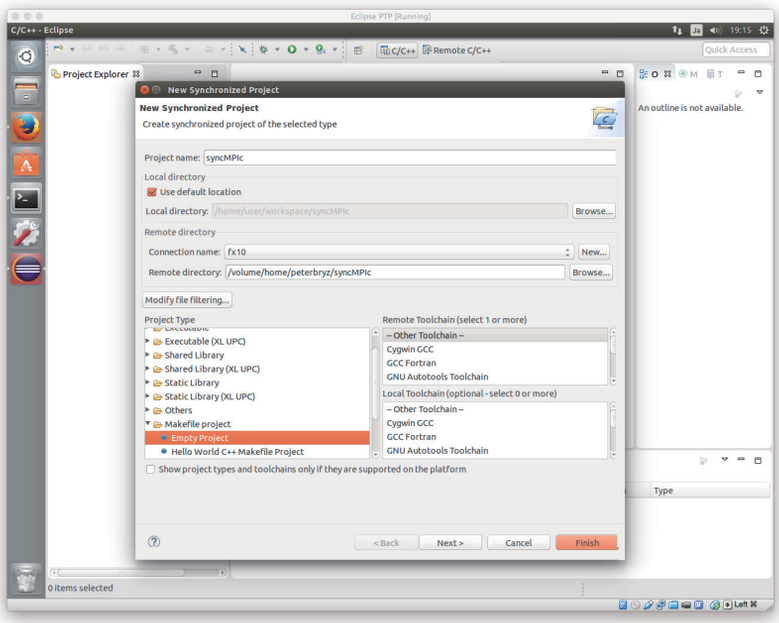
In the Project Explorer, right-click on empty space, select New > Synchronized C/C++ Project or Synchronized Fortran project.



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



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: 

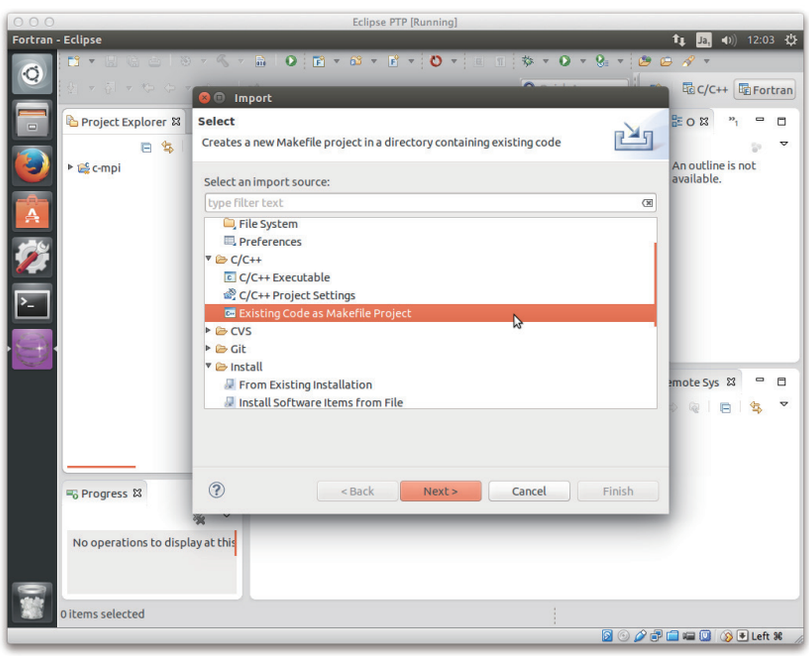
2.2 From local code

2.2.1. Import Fortran code

Local code is any C/C++/Fortran program source code that is not part of any Eclipse project.

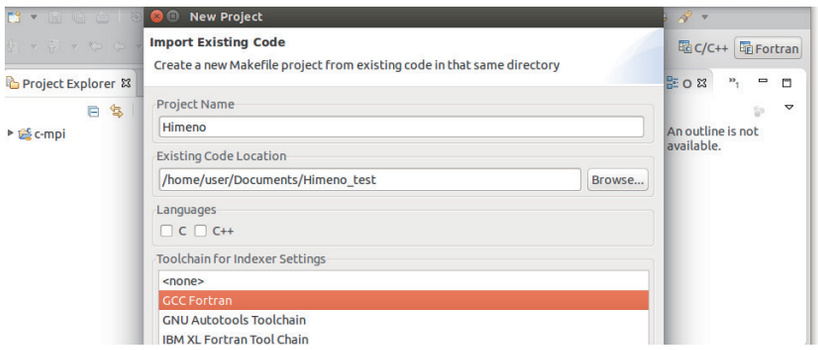
To convert local source code to Eclipse project right-click on empty space in Project Explorer and select Import...

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

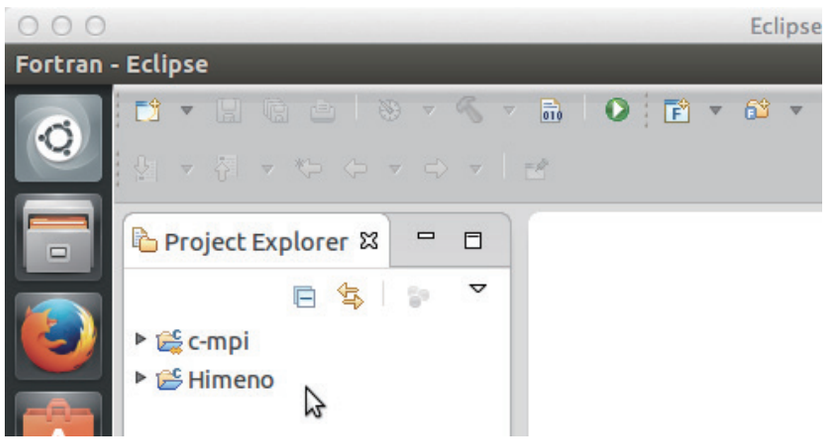


Click Next.

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

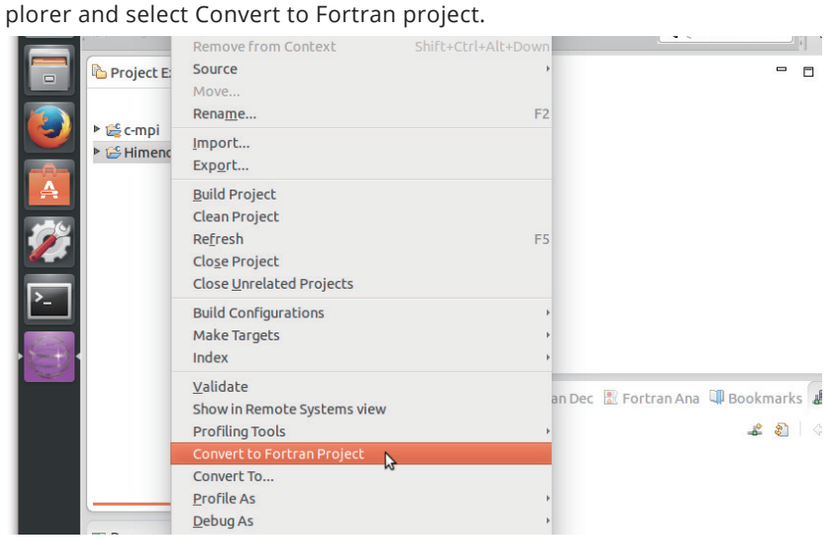


Click Finish.



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

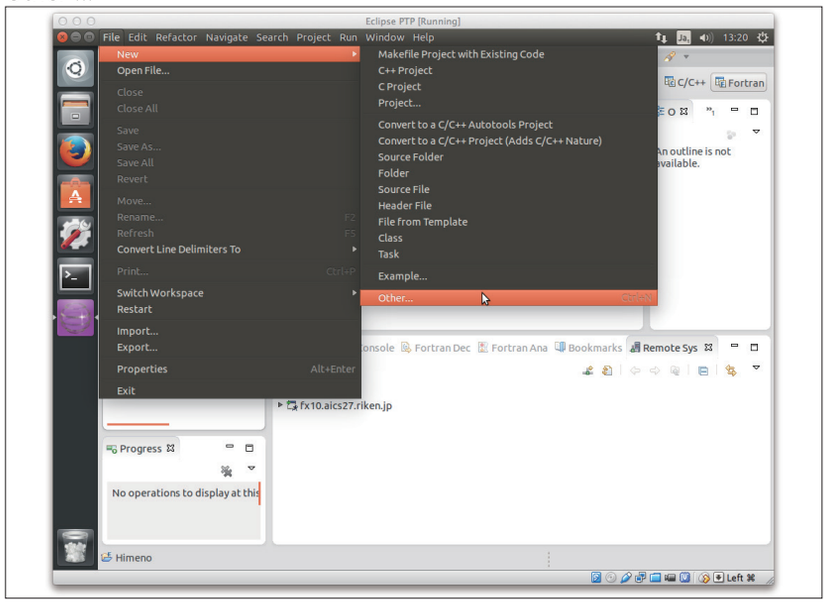
Convert it into a Fortran project. Right-click on the project in Project Explorer and select Convert to Fortran project.



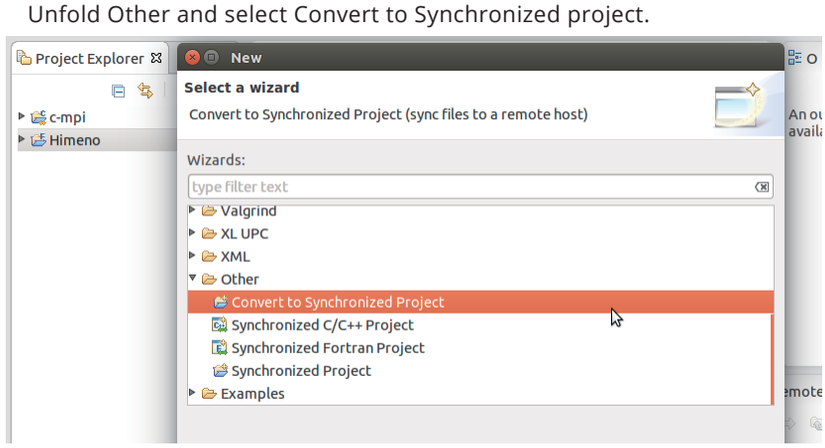
Make sure, the project folder icon changed to  . There is no arrow, because the project is not synchronized (local).

2.2.2. Convert local project to synchronized

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

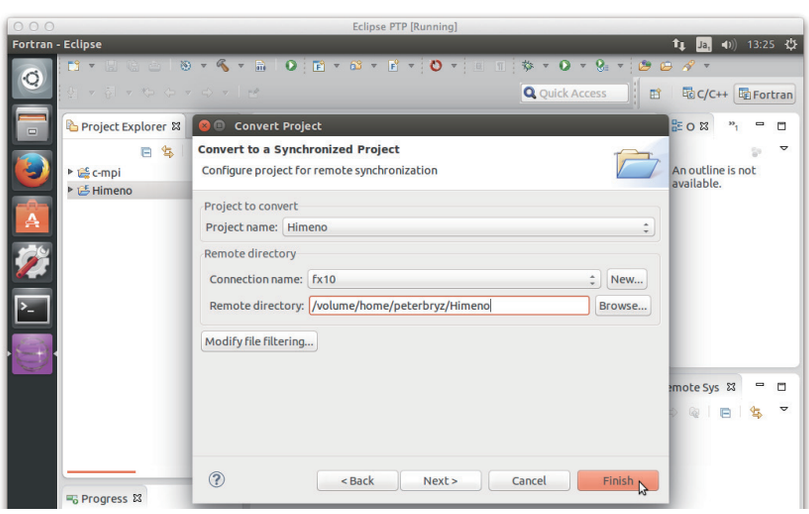


Unfold Other and select Convert to Synchronized project.



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

Click Finish.

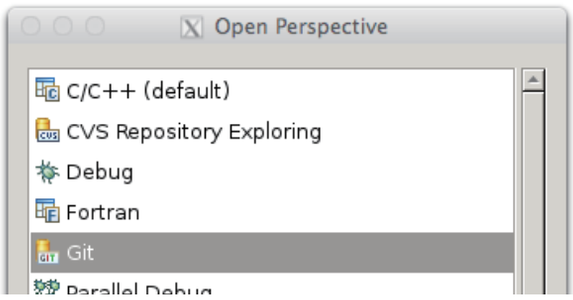


Note, that project icon changed to  .

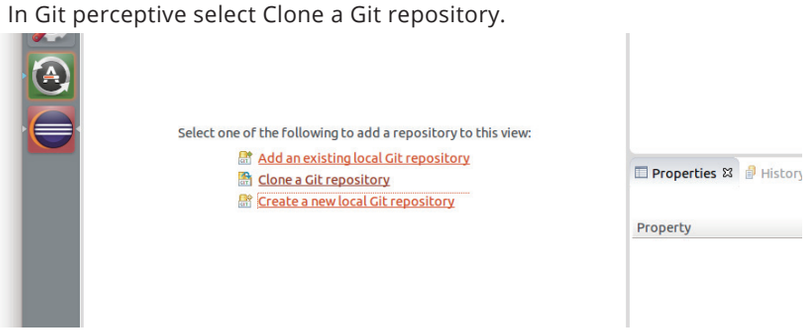
2.3 Create synchronized project from git repository

To create a new synchronized project from existing remote git repository:

Open git perspective from top menu: Window > Open perspective > Other and select Git.

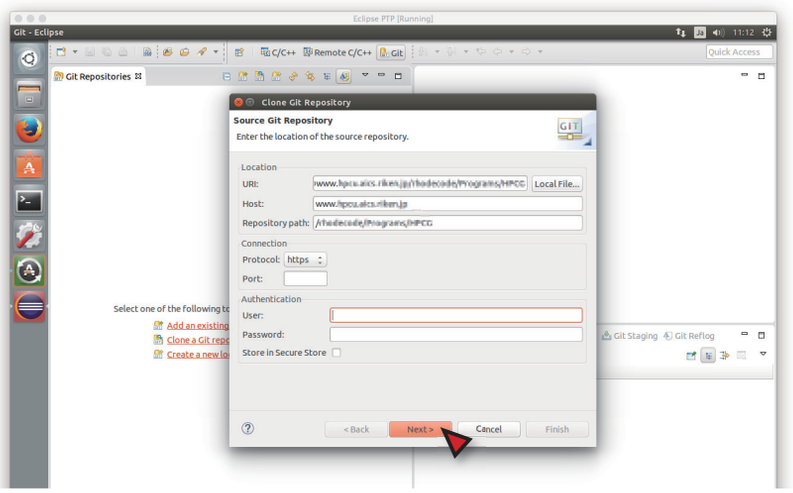


In Git perceptive select Clone a Git repository.

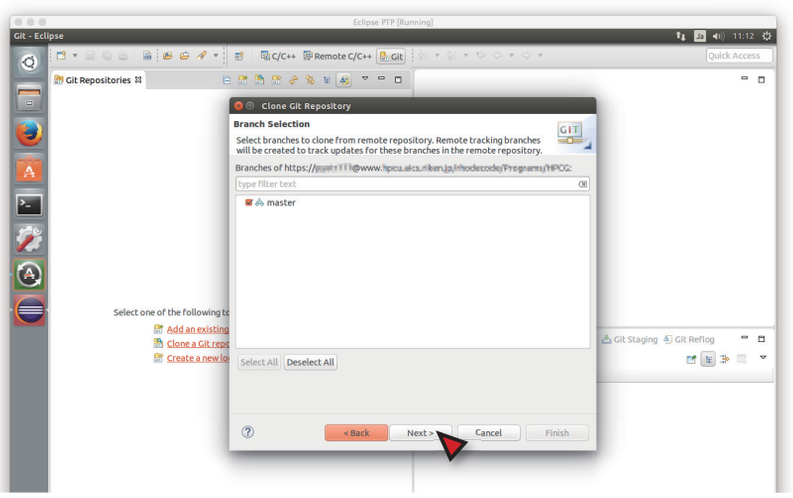


Fill in repository URI and other fields (some fields will be filled automatically after you fill URI). Click Next.

Cloning git repository



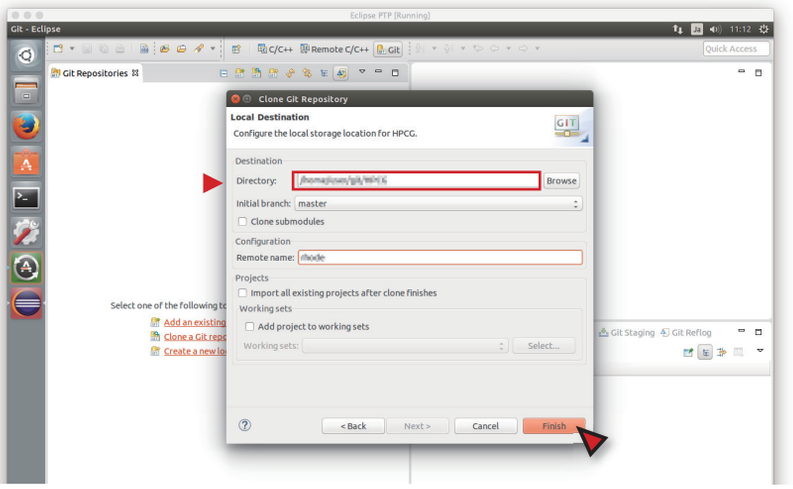
Select git branch and click Next.



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.

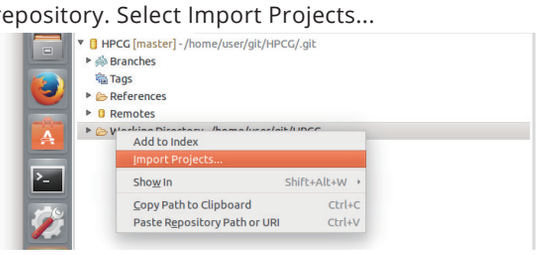
Remeber this path, you will need it later



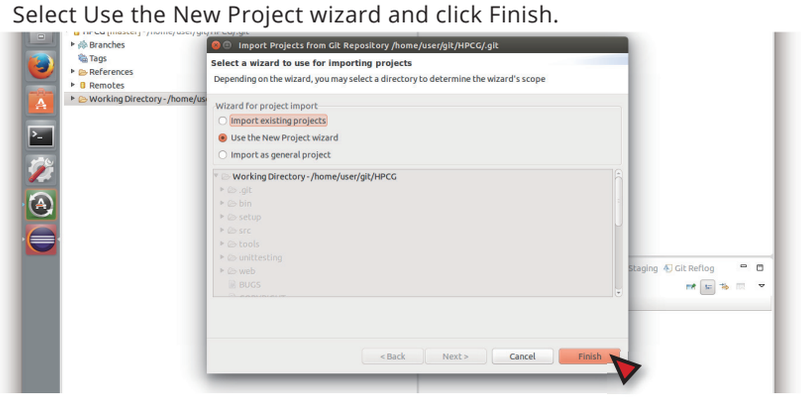
Repository is cloned.

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

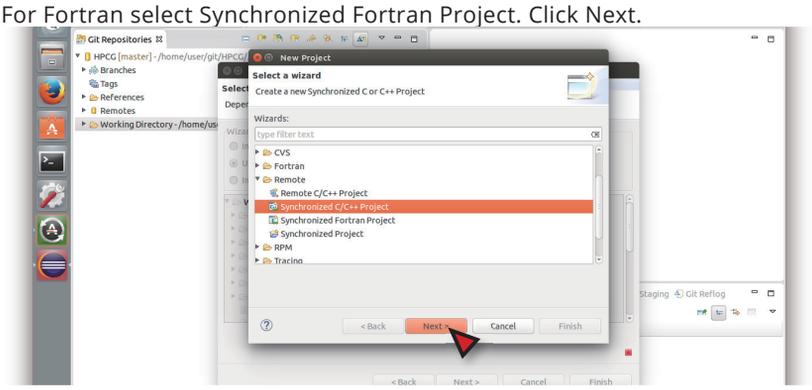
Importing new project from git repository



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.



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.*

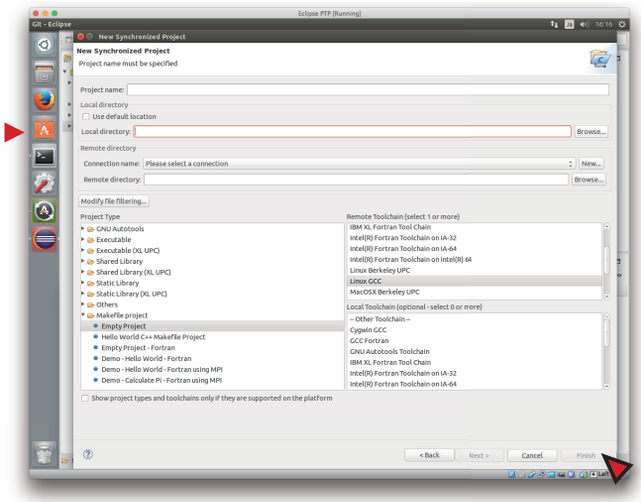
Uncheck Use default location. Select local repository location instead.

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.

New synchronized project wizard



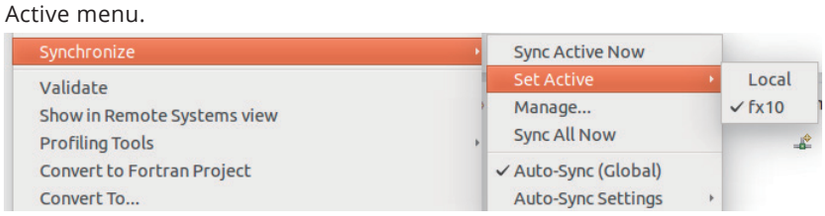
Your new project now should be visible in Project Explorer (C/C++ or Remote C/C++ perspective).

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

3. Synchronization

To synchronize files in local and remote locations 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.

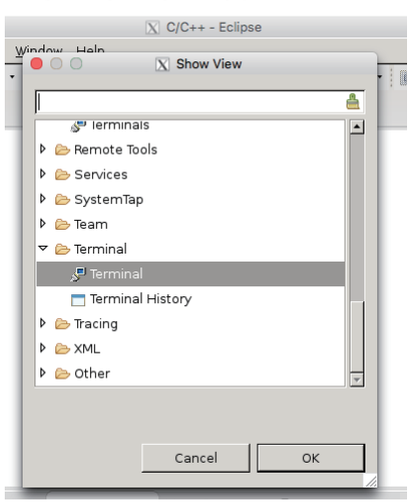
To change connection used for synchronization, select one under Set Active menu.



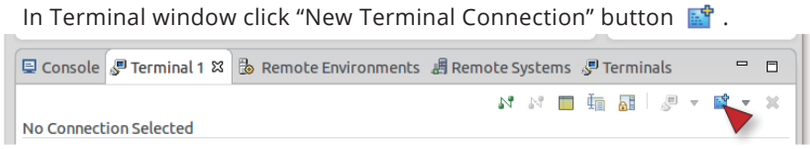
4. Remote terminal

With remote terminal you can login to remote system with SSH.

Open Terminal window from menu Window / Show view / Other ... Terminal / Terminal.

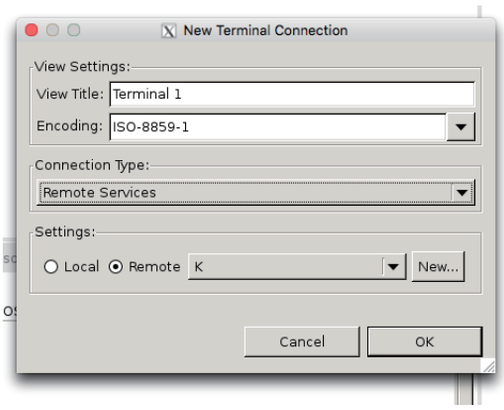


In Terminal window click “New Terminal Connection” button .



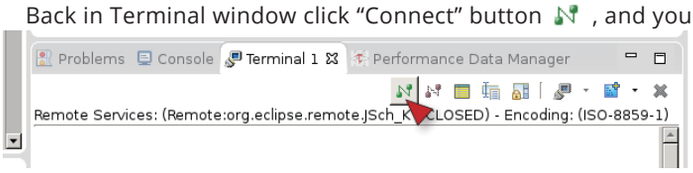
and select connection you created before or create a new connection.

Selecting a remote connection for Terminal

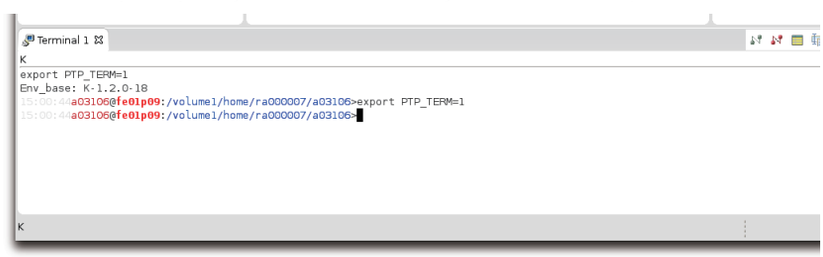


After you selected a connection click OK button.

Back in Terminal window click “Connect” button , and you will see a command prompt in your home directory on remote computer.



Terminal to a remote system



5. Build a project

To build a project you need to set build command, compiler options, environment variables and other compilation parameters in a Build Configuration.

Projects can be *Makefile based* or *Managed*.

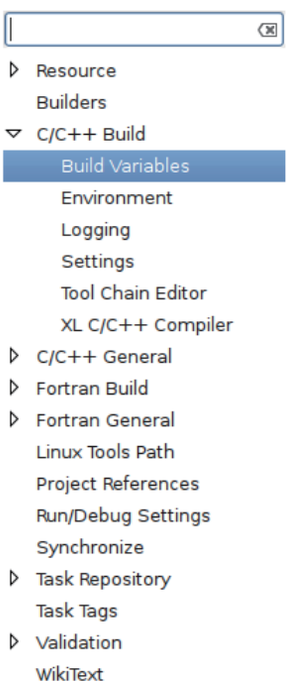
Makefile based project contains its own build command – a makefile or build script. For Managed projects Eclipse manages build process, no makefile required.

A Build configuration provides the necessary information for building the project. Build configuration information is specified in the project properties. Projects can have multiple build configurations.

5.1 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 from menu File / Properties.

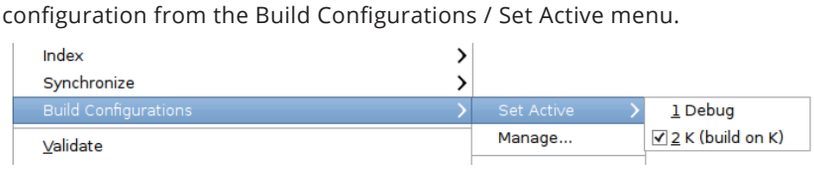
Note: Fortran projects are a superset of C/C++ projects, so they also have properties of C projects.



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.

Active build configuration will be used when the build button  is clicked. 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.



6. Run a project

To run a project you need 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.

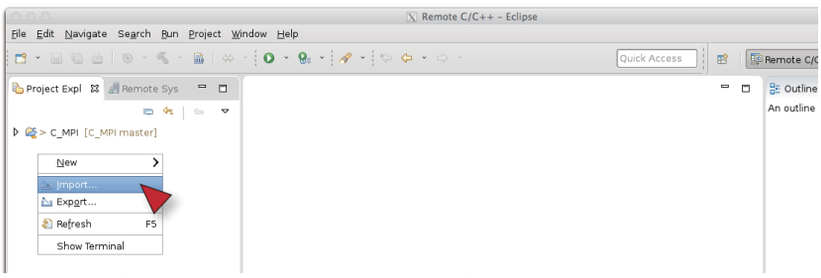
6.1 Run Configurations

Run configurations depend on the target system. For K, FX10 and other systems with Fujitsu “Parallelnavi” job scheduler you can 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.

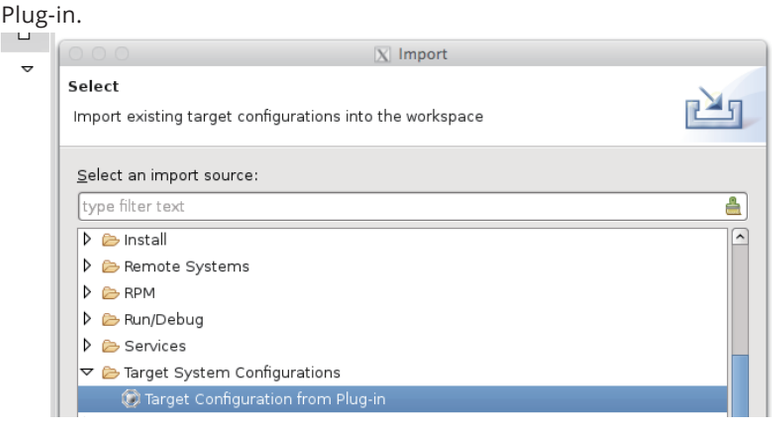
6.1.1. Import Target System Configurations

To use PJM-* (and other custom) Target System Configurations you need to create “targetConfigurations” project in the following way:

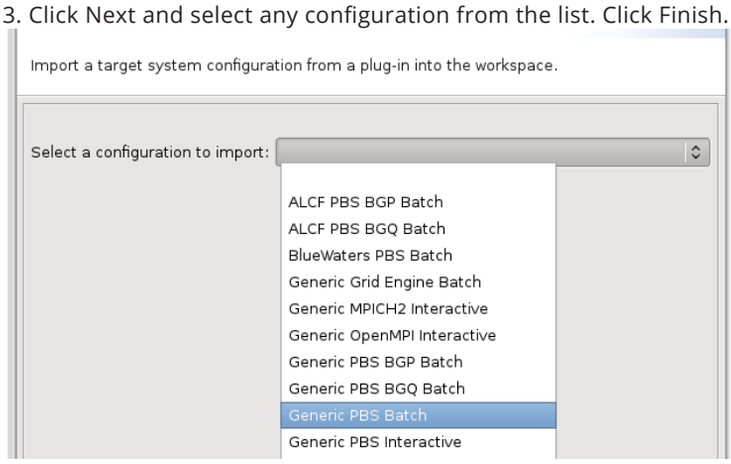
1. Right-click empty space of Project Explorer window and select Import, or select File / Import menu.



2. Select Target System Configurations / Target Configuration from Plug-in.



3. Click Next and select any configuration from the 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. Download Target System Configuration for the K computer “PJM-Generic-MPI” into this directory with the following

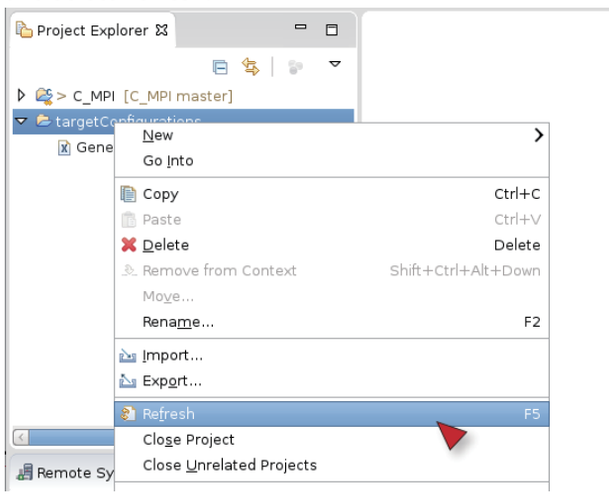
command:

```
> wget https://github.com/pyotr777/EclipsePTP_Parallelnavi_TSC/raw/master/PJM-Generic-MPI.xml
```

For FX10 use another TSC:

```
> wget https://github.com/pyotr777/EclipsePTP_Parallelnavi_TSC/raw/master/PJM-Generic-MPI-FX10.xml
```

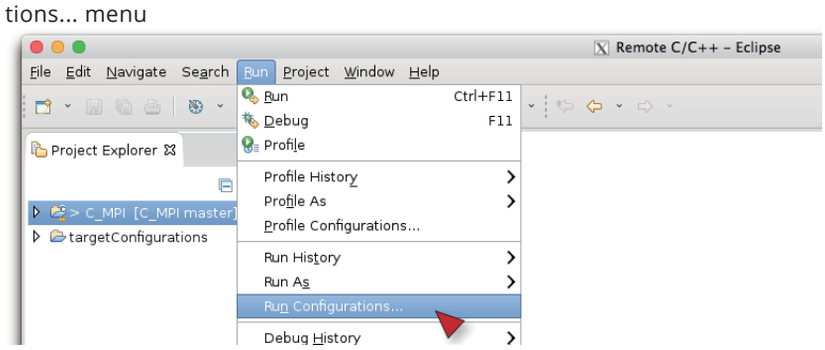
6. Return to Eclipse, right-click targetConfigurations in Project Explorer and select Refresh.



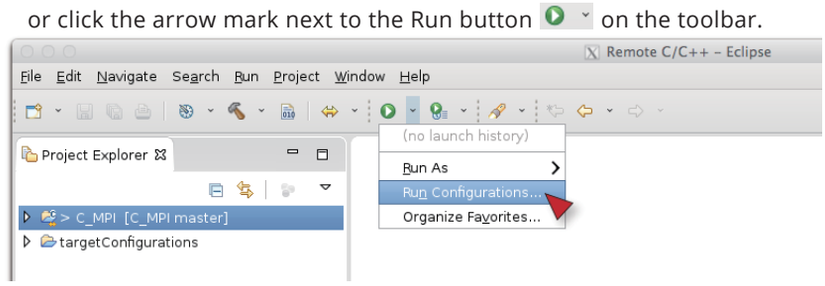
Now in Project Explorer you should be able to see PJM-* files in targetConfigurations project.

6.1.2. Create Run Configuration

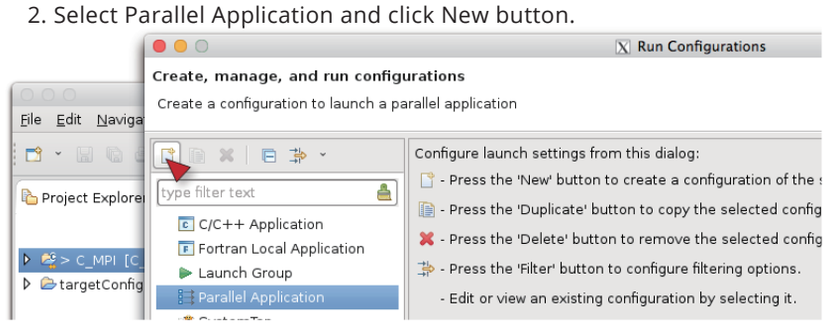
1. Select your project in Project Explorer and select Run / Run Configurations... menu



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



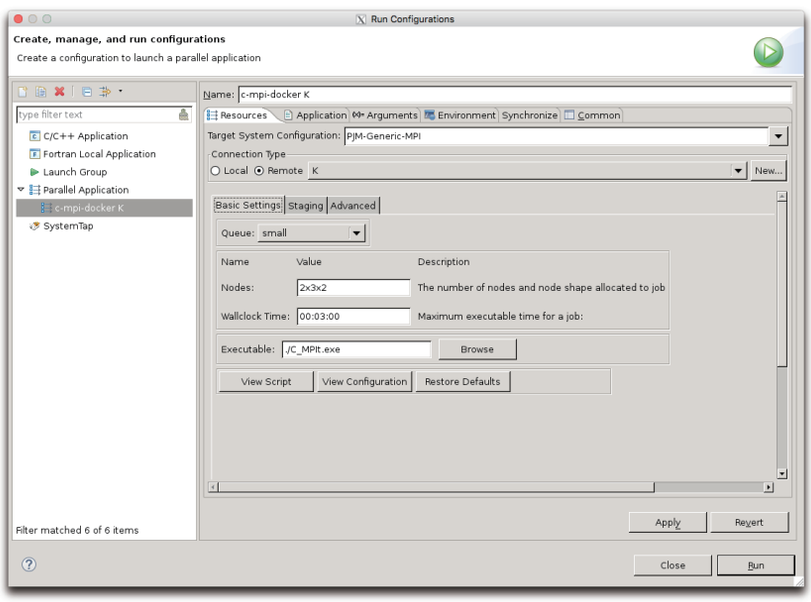
2. Select Parallel Application and click New button.



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 tab select Queue, number of nodes, estimated job run time (Wallclock time), 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.

Your job will be submitted for execution on the remote system.

A dialog will appear, asking if you want to switch to Monitoring view.

6.1.3. Submitting a job with custom job script

Sometimes job script can be generated automatically by your project, not by Eclipse. In such cases, when you do not want Eclipse to generate a job script, you can submit your job manually.

After job script is generated, make sure project is synchronised.

Login to remote computer using Terminal window.

See Chapters 3 and 4 for synchronization and Terminal login instructions.

Change directory to the directory with your custom job script and submit a job with pjsub command. For example

```
> pjsub run.sh
```

After your job finishes execution you have to synchronize your project manually once again.

Now you should be able to see new files in your project in Project Exlorer window.

Note! Jobs submitted manually are not recognised by Eclipse. This means, in particular, that your project is not synchronised by Eclipse and you cannot control your job in Eclipse “System Monitoring” perspective (see 7.3).

7. Monitoring

Monitoring view displays a graphical representation of compute nodes 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.

7.1 Installation

Monitoring uses scripts and configuration files on remote computer. To enable monitoring you need to have these files in .eclipse directory in your home directory on remote computer.

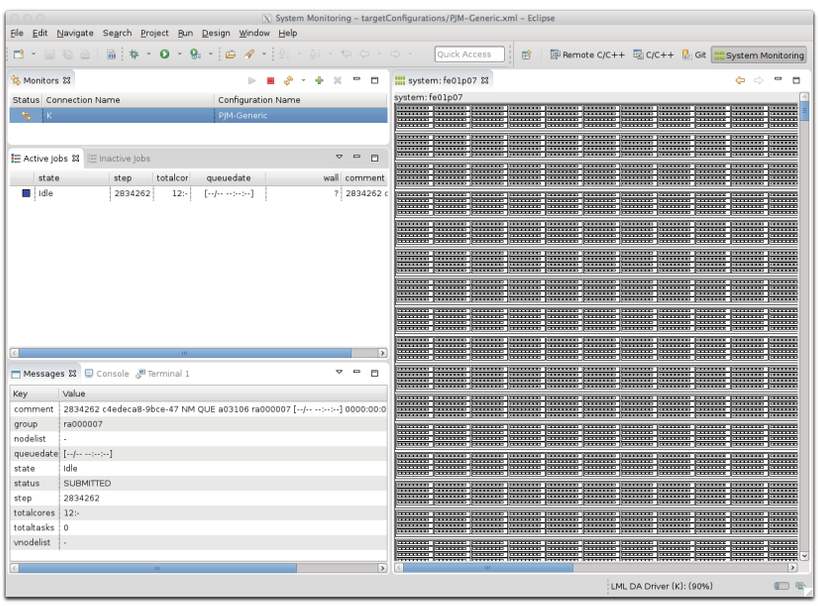
For the “K” computer execute the following in your home directory:

```
> mkdir .eclipse
> cd .eclipse
> git clone https://github.com/pyotr777/eclipse_on_K.git --branch K .
```

(don't forget the last dot ".")

For FX10 computer replace "--branch K" with "--branch fx10".

Sample monitoring screen for the "K" computer



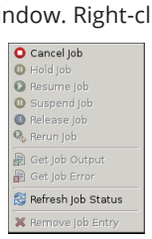
7.2 Monitoring

Monitoring scripts run at predefined intervals (about 1 to 1.5 minutes) collecting information about computing nodes and jobs. These scripts will continue running even if you switch to another perpective in Eclipse PTP and Monitoring view will be hidden.

To stop monitoring scripts select running connection in the Monitors window and click on the red square above. Click on the green trianlge to start monitoring again.

7.3 Job control

You can cancel your jobs from Active Jobs window. Right-click on your job and select Cancel Job from the pop-up menu.



8. Performance analysis of MPI applications

To analyse parallel program performance Eclipse PTP can use TAU (Tuning and Analysis Utilities)
https://www.cs.uoregon.edu/research/tau/home.php

8.1 TAU installation on local computer

To use TAU with Eclipse PTP we recommend to install TAU on your local computer. It will let you store TAU profiles in a DB for easy access, and open profiles and traces locally.
Latest TAU distributions can be downloaded from
https://www.cs.uoregon.edu/research/tau/downloads.php

For Linux OS follow these steps. This will install the latest version of TAU (2.25) in $HOME/TAU/tau-2.25 direcotry.

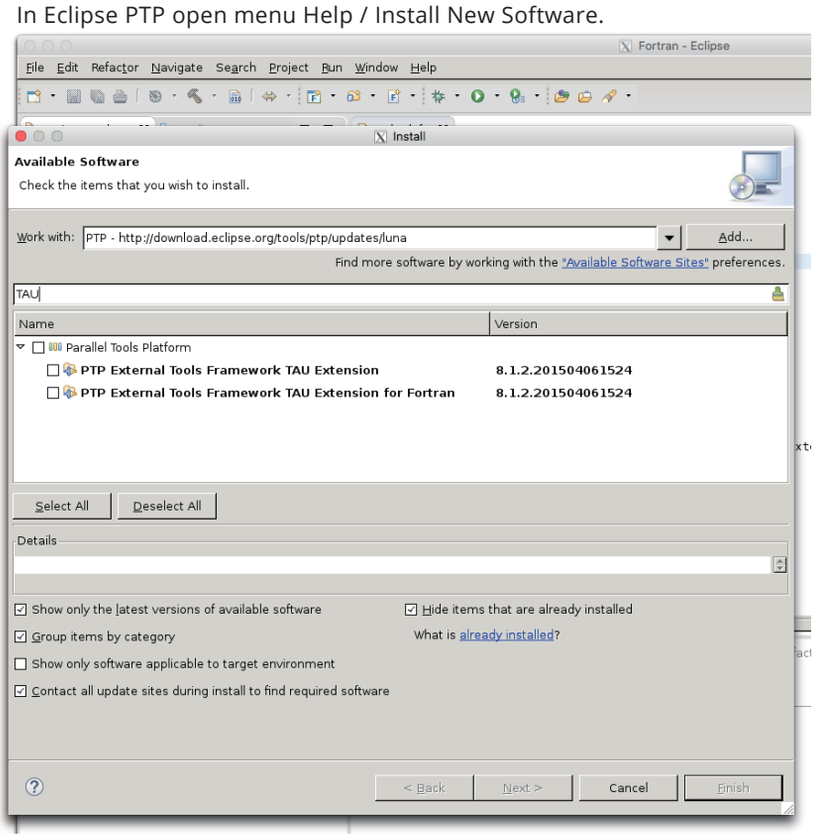
```
mkdir -p $HOME/TAU
cd TAU
wget http://tau.uoregon.edu/tau.tgz
tar -xzvf tau.tgz
cd tau-2.25
./configure
make
make install
```

For OS X install TAU from the package: http://tau.uoregon.edu/tau.dmg

Add the following directory to $PATH variable:
$HOME/TAU/tau-2.25/x86_64/bin

8.1.1. Install Eclipse PTP TAU extensions

In Eclipse PTP open menu Help / Install New Software.



Select TAU Extensions and click FInish button. Restart Eclipse.

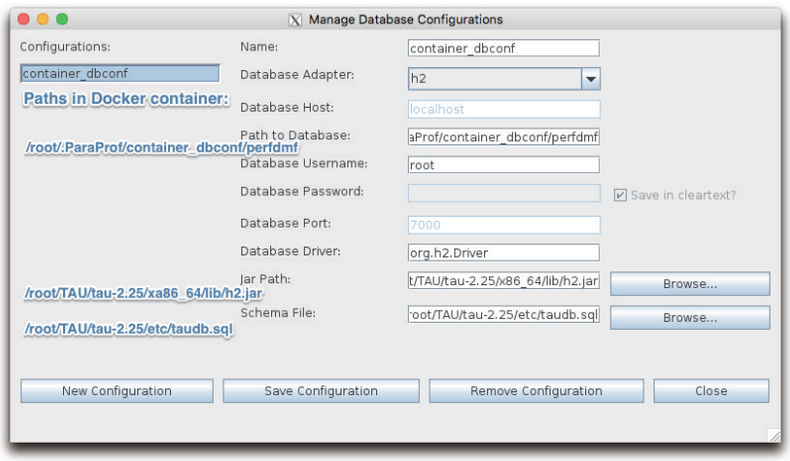
8.1.2. Setup local Database for profiles

To store TAU profiles in local DB:
Run paraprof (you need to setup $PATH as instructed above):

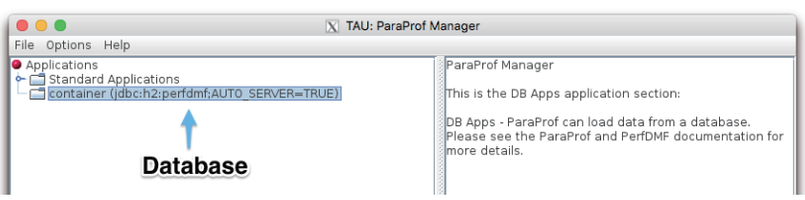
```
> paraprof
```

In Paraprof select menu File / Database configuration.
Here is a sample configuration:

Profiles Database configuration in Paraprof



You can use any Name and Username. For the database port any number of unsed port greater than 1024 will do.
Save configuration.
If you succesefully created DB, you will see it in Paraprof:



Please note that traces cannot be stored in a database.

8.2 TAU instrumentation

TAU (version 2.24.1) is installed on the "K" computer in /opt/aics/TAU/ directory. TAU allows two basic options for performance analysis: *profiling* and *tracing*. Application profiles show the exclusive and inclusive time spent in each function, how many times each function was called, how many profiled functions did each function invoke, and what the mean inclusive time per call was. Application traces show when and where event occured in terms of the process that executed it and the location in the source code.

More information on profiling and tracing with TAU can be found in TAU documentation:

https://www.cs.uoregon.edu/research/tau/docs.php

For profiling and tracing user application must be recompiled with TAU instrumentation. There are two instrumentation methods available on K computer:

- Compiler Based Instrumentation,
- Source Based Instrumentation.

Table 1. Different instrumentation methods

Method	Requires recompiling	Requires PDT	Shows MPI events	Routine-level event	Low level events (loops, phases, etc...)	Throttling to reduce overhead	Ability to exclude file from instrumentation	Ability to exclude other regions of code
Compiler	Yes		Yes	Yes		Yes	Yes	
Source	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

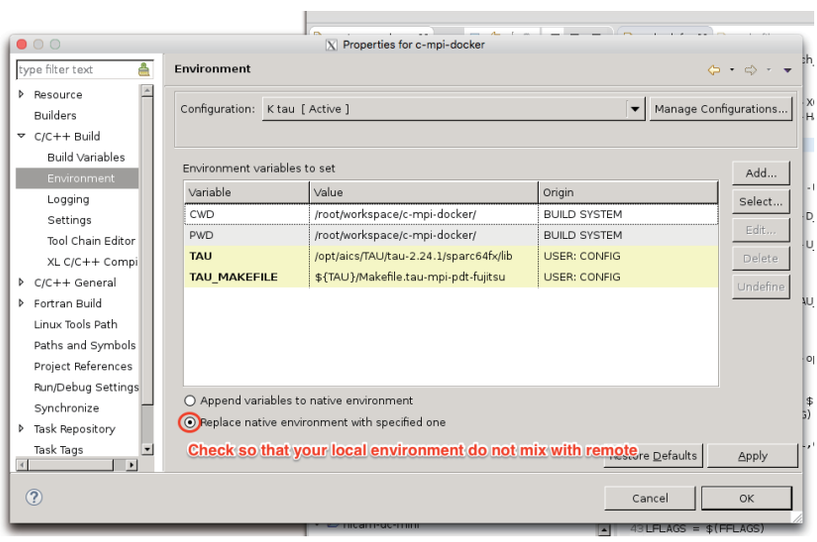
Source instrumentation requires PDT and produces larger profiles and traces, so it is recommanded to start with Compiler instrumentation and use Source instrumentation if more features are needed.

To use TAU instrumentation you will need to initialise environmet variables on the remote (supercomputer) side. For that pupropse we provide an environment initialisation script file for K computer located at /opt/aics/TAU/env.sh. We recommend you insert this script into your *.bashrc* on the K computer like this:

```
source /opt/aics/TAU/env.sh
```

Also you need to add TAU_MAKEFILE variable to your project C/C++ Build or Fortran Build / Environment:

Sample build environment settings for TAU instrumentation.



On **FX10** you have to install TAU and optionally PDT in your home directory and edit environment initialisation script to point to your TAU installation.

You need to recompile your source code and produce instrumented binary. To do that use the following scripts instead of compilers:

Table 2. TAU instrumentation scripts

TAU script	use for
tau_f90.sh	Fortran
tau_cc.sh	C
tau_cxx.sh	C++

8.2.1. Compiler Based instrumentation

To use Compiler based instrumentation you need to set up yet another environment variable in your Build Environment:

TAU_OPTIONS=-optCompInst

You can also use options with the instrumentation scripts:

```
> tau_cc.sh -tau_makefile=/opt/aics/TAU/tau-2.24.1/sparc64fx/lib/Makefile.tau-mpi-pdt-fujitsu -tau_options=-optCompInst sampleprogram.c
```

8.2.2. Source Based instrumentation

Source based instrumentated binary can be produced in exectly the same manner as compiler based instrumentation. The only difference is that you do not need to add -optCompInst option (or environemnt variable) for TAU instrumentation script.

```
> tau_cc.sh -tau_makefile=/opt/aics/TAU/tau-2.24.1/sparc64fx/lib/Makefile.tau-mpi-pdt-fujitsu sampleprogram.c
```

8.3 Running instrumented binary

Instrumented binary can be run on compute nodes. By default it will produce profile.N.0.0 files (by the number of MPI ranks) in the same directory with your job script. You need to add **staging out instructions** for these files to your job script. See 8.3.1 for sample instructions.

There are some environment variables you can set on compute nodes with your job script to control TAU profiling and tracing on compute nodes.

Table 3. TAU configuration variables

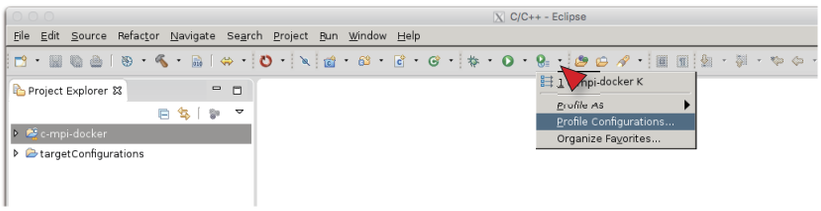
Environment variable	Effect
TAU_PROFILE	Set to 1 to enable profiling, to 0 to disable.
TAU_TRACE	1 to enable tracing, 0 to disable
TAU_PROFILE_FORMAT	When set to snapshot TAU will generate condensed snapshot profiles (they merge together different metrics so there is only one file per node) instead of the default kind. When set to merged, TAU will pre-compute mean and std. dev. at the end of execution.
PROFILEDIR	Set to relative path to directory where to save profiles. The direcotry need to exist.
TRACEDIR	Specifies the directory where trace file are to be stored.

For more variables see:
https://www.cs.uoregon.edu/research/tau/docs/old/apa.html

8.3.1. Profile Configurations

Similar to Run Configurations, Eclipse PTP has Profile Configurations that can generate a job script and submit a job.

To run a profiling or tracing job you need to create a new Profile Configuration in Eclipse PTP. To create a new Profile Configuration, find this button on the toolbar of Eclipse PTP and click on the small black triangle next to the button.



From a pop-up menu select "Profile Configurations...".

You will be asked if you want to connect to K or FX10. Click "Yes". A Profile Configuration window will open.

You can name your configuration as you like. Configurations are supposed to be used with particular project for a particular run configuration, so we recommend to include you project name, target supercomputer and short job description in the name.

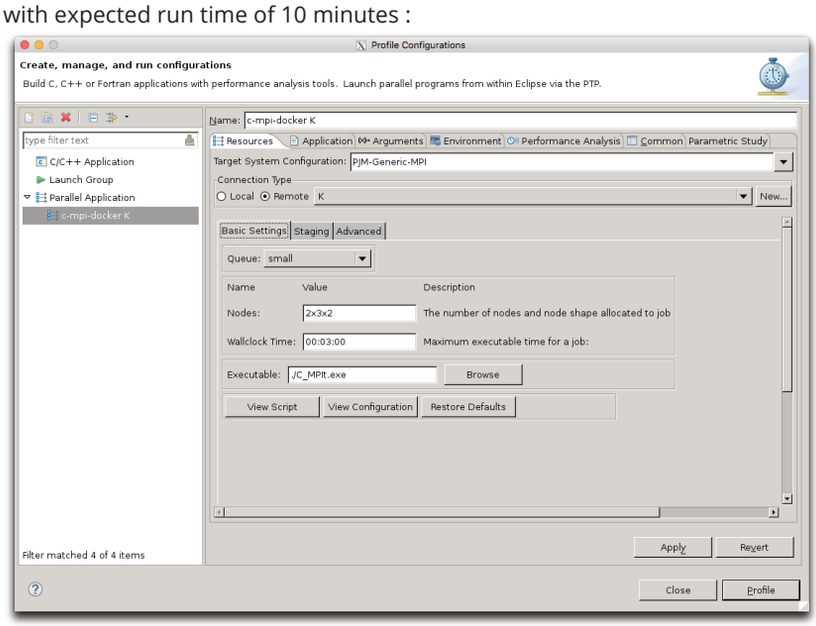
For a Target System Configuration select "PJM-Generic-MPI" for K comuter and "PJM-Generic-MPI-FX10" for FX10.

Select a desired Connection or create a new one.

Fill in basic job parameters in Basic Settings tab.

Here is a Profile Configuration for running a job on 12 nodes of K computer with expected run time of 10 minutes :

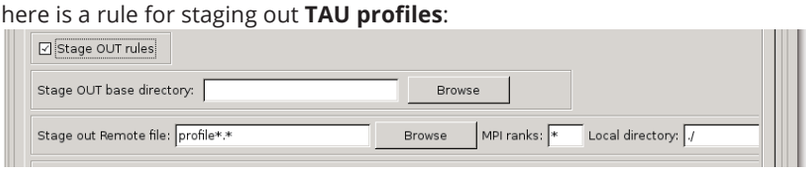
Sample Profile Configuration window



Be sure to set **Executable** to the relative path to a binary file instrumented with TAU.

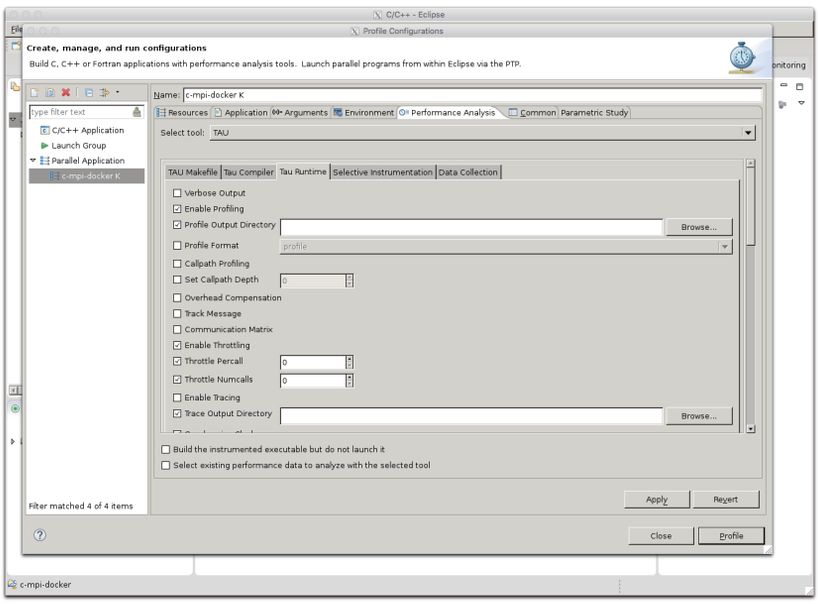
You have to take care about staging out profiles and traces from compute nodes to login node. In Staging tab add rules for staging out. For example here is a rule for staging out **TAU profiles**:

Staging out rule for TAU profiles



TAU traces consist of two sets: evens.N.edf files and tautrace.N.0.0.trc files. N is a number from 0 to the number of MPI ranks -1. You need to stage out all of these files.

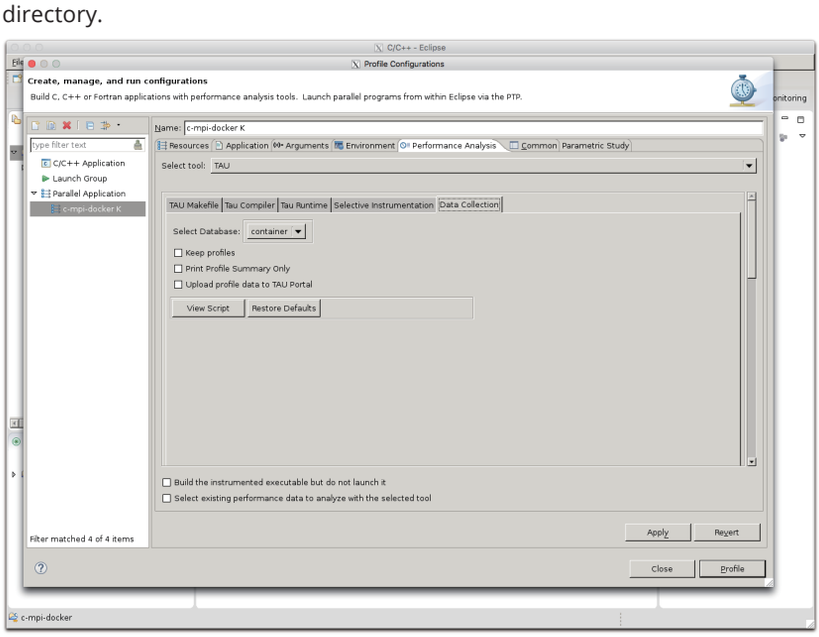
On Performance Analysis tab select TAU from drop-down list on top.



On Runtime tab of Performance Analysis you have to take care of Porfile and Trace Output Directory.

If you set staging out rule for profiles as indicated above, you have to set Profile Output Directory path to be empty. Do not just uncheck it!

If you chose to use a special directory for storing traces or profiles, you need to enter the path to that directory here. The directory need to be created on computing nodes before job starts. You can use Advanced tab of PJM-Generic-MPI on Resorces tab for adding a command for creating the directory.



On Data Collection tab select Database to store profiles. See 6.2.2 for instructions on DB setup.

You can press View Script button any time to see what your job script will look like.

8.3.2. Submit profiling or tracing job

When you are ready to submit the job press Profile button in bottom right corner of Profile Configuration window. Your job will be submitted and the settings will be saved in Profile configuration, so you can easily stubmit same job later.

After you submit a job you will be asked to switch to Monitor view.

After your job is finished profiles should be stored in a DB automatically. Profiles stored in a DB can be opened in Paraprof by clicking on them.

Profiles that are not stored in a DB can be found in Profiles subdirectory of your project. To open profiles not stored in a DB you need to open Paraprof and select profiles from the menu File / Open, or right-click on a profile in Project Explorer view, select Open With and then select Paraprof.

Traces will be in the directory you set with staging out rules. If you set an empty path for Trace Output Directory, traces will be in your project folder.

Traces can be viewed in Jumpshot application.

See below for instructions on viewing traces.

8.5.1. Profiling with custom job script

When you have to use a particular job script you cannot use Eclipse profiling features and have to submit your job manually.

Submit your custom job script for execution as described in 6.1.3, but use an instrumented binary. By default TAU will profile your application. You can control TAU behaviour by setting environment variables on compute nodes as decribed in 8.3 using your job script.

Do not forget to set staging out rules for profiles or traces in your job script.

After a job finished execution you have to manually synchronised your project in Eclipse to see profiles and traces in Project Explorer.

8.4 View TAU pofiles

TAU profiles can be viewed in Paraprof.

8.5 View TAU traces

TAU traces need to be merged and converted to SLOG2 format before you can open them in Jumpshot trace viewer.

TAU tracing job produces two sets of files with .trc and .edf extentions. Number of files in each set is the number of MPI ranks that your application has used.