ECLIPSE PTP

for K and FX10 computers

BASIC USER MANUAL

V1.5

This manual contains instructions on installing Eclipse PTP on Linux operating systems and using Eclipse PTP for developing and running parallel applications on K and FX10 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.

This manual describes Eclipse PTP usage with Fujitsu supercomputers FX10 and K, but can be useful for other supercomputers as well.

Symbol  shows mouse click point.

理化学研究所
計算科学研究機構
RIKEN Advanced Institute for Computational Science

Peter Bryzgalov, HPC Usability Research Team

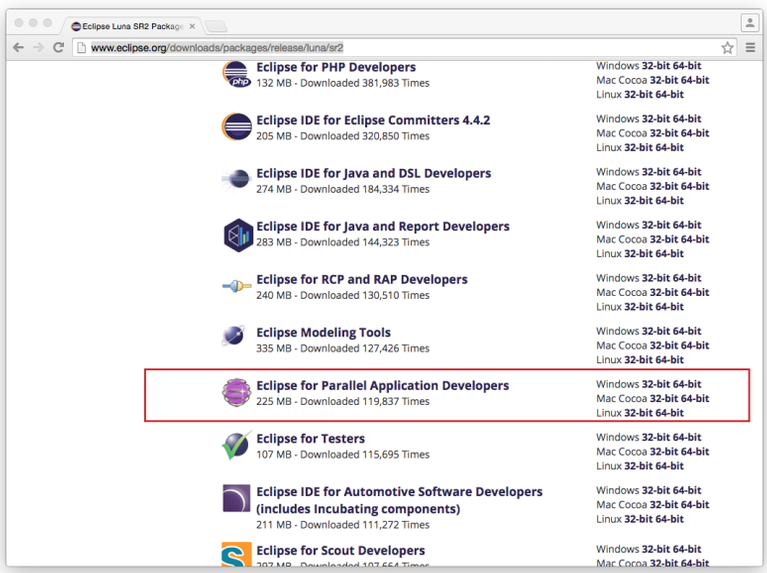
Contents

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:



For Windows we recommend using Mars version, because Luna has problems with submitting profiling jobs on Windows.

Download it from here:
https://www.eclipse.org/downloads/packages/eclipse-parallel-application-developers/mars1

Download archive.

On Linux:

```
$ 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

On OS X extract downloaded archive and move extracted folder to `/Applications` directory.

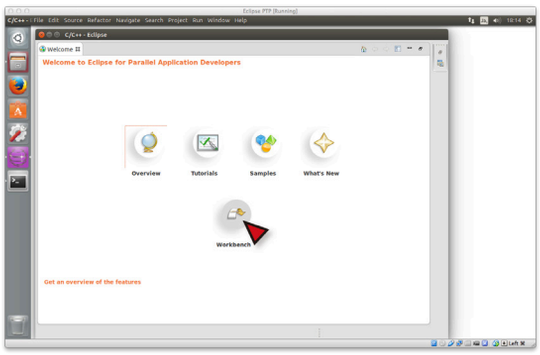
Update Java if necessary.

1.2 Start Eclipse PTP

Note!

Remember your workspace path. All your projects should be stored under this directory.

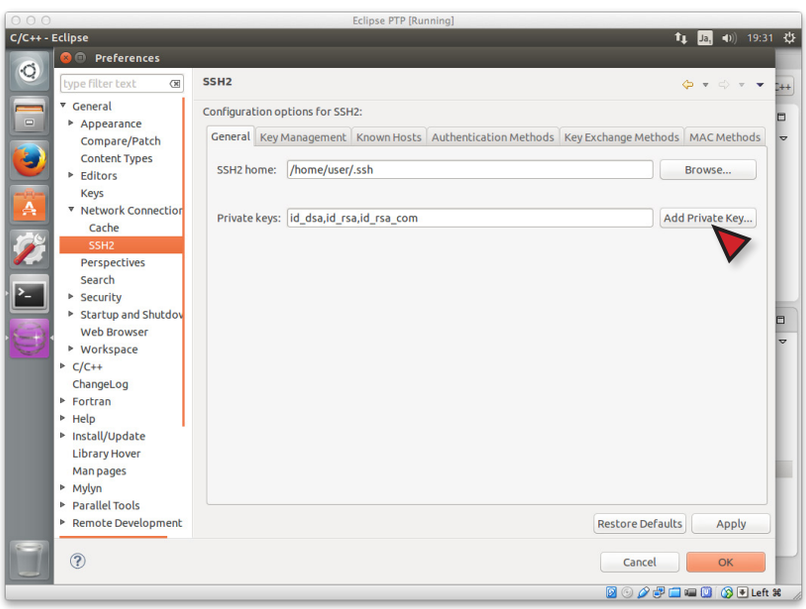
Start Eclipse PTP from extracted directory.
Set Workspace directory.
Eclipse window will show up.



1.3 Add SSH keys

To be able to connect to remote computers with SSH and private keys, you need to tell Eclipse which SSH key files can be used.

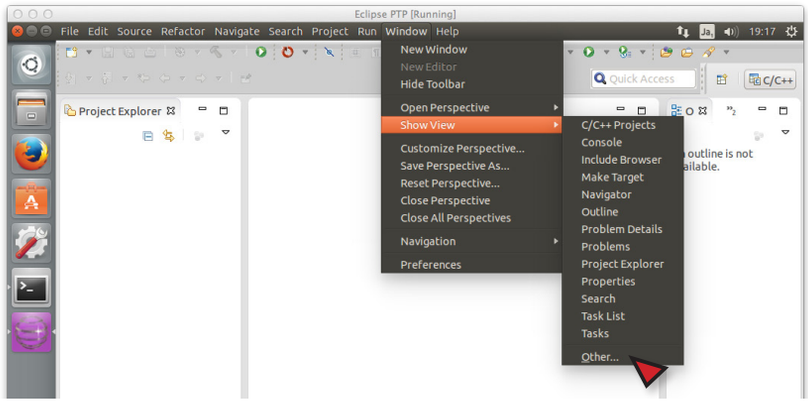
Open Eclipse Preferences window.
Unfold General / Network connection, select SSH2.
Click Add Private Key and select files with SSH keys you are going to use.



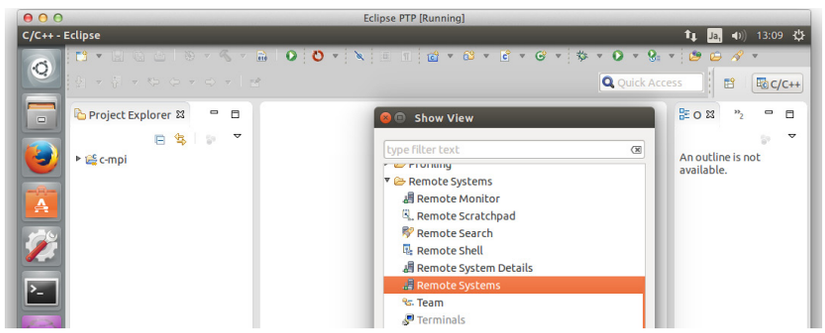
1.4 Add a Remote system connection

To browse remote files, login with SSH and execute remote command in Terminal you need a Remote System connection.

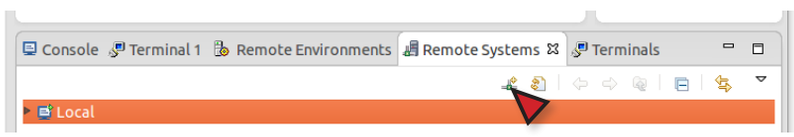
To create new Remote System connection open Remote Systems view: Window / Show View / Other,



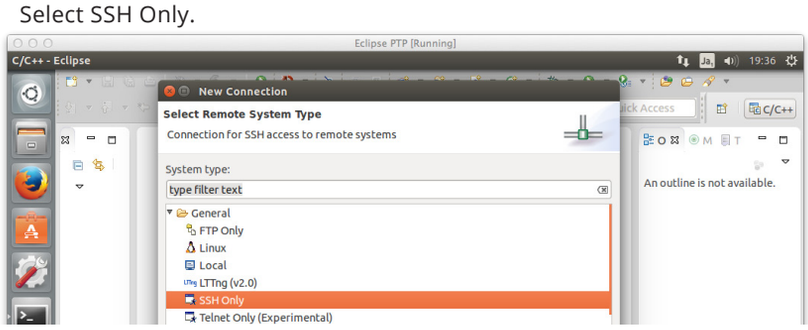
Remote Systems / Remote systems.



In Remote Systems window click on  button to add a new connection to a remote system.

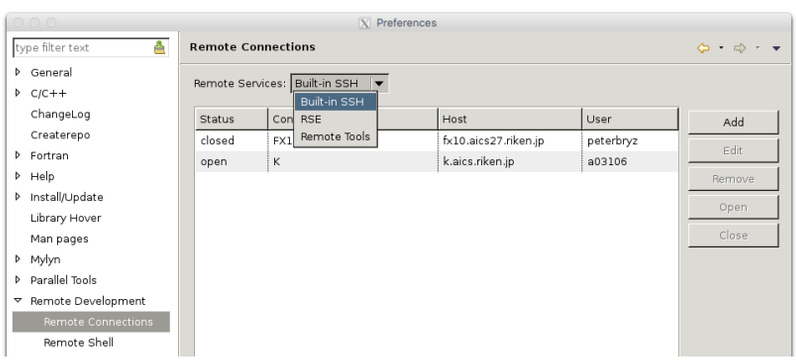


Select SSH Only.



Enter Host name, Connection name and Description. Click “Finish”.

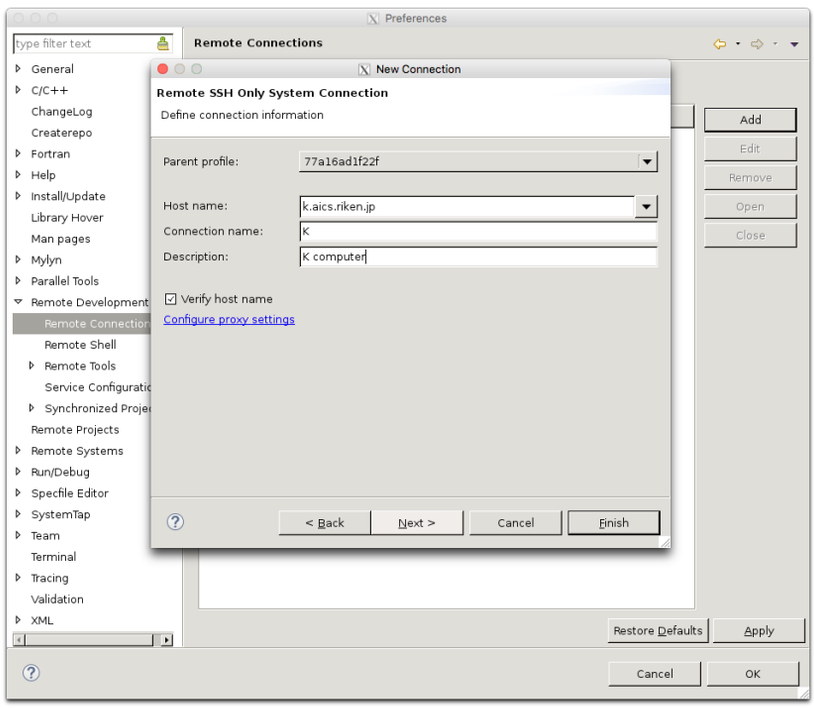
To see available connections open Preferences window, unfold Remote Development and select Remote Connections.



Using “Remote Services” drop-down menu you can select connections type. Remote system connection that you have created should be visible in “RSE” Remote Services.

You can also create new connection here in Preferences window. Select “RSE”, click “Add”, select “SSH only”, click “Next”.

New Connection for Remote System Explorer (RSE)

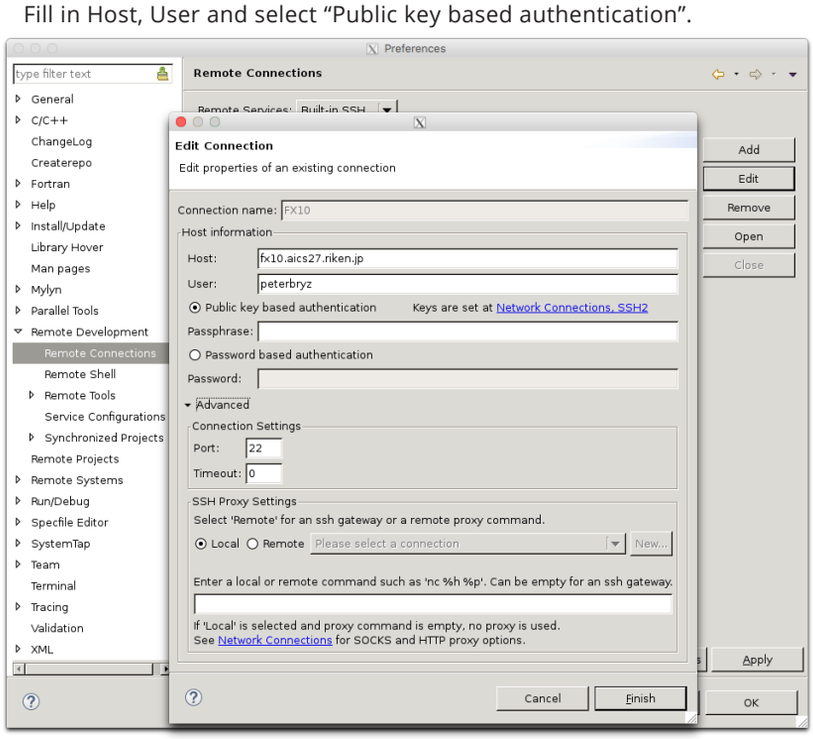


Another type of connection is used for synchronization, building and execution of synchronized and remote projects.

1.5 Add a synchronization connection

For monitoring and projects synchronization “Built-in SSH” type connections are used. To create a new Built-in SSH connection select “Built-in SSH” in the “Remote Services” drop-down menu of Eclipse Preference window and click “Add” button.

Fill in Host, User and select “Public key based authentication”.



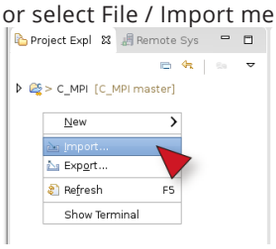
Click “Finish”.

1.6 Install Target System Configurations

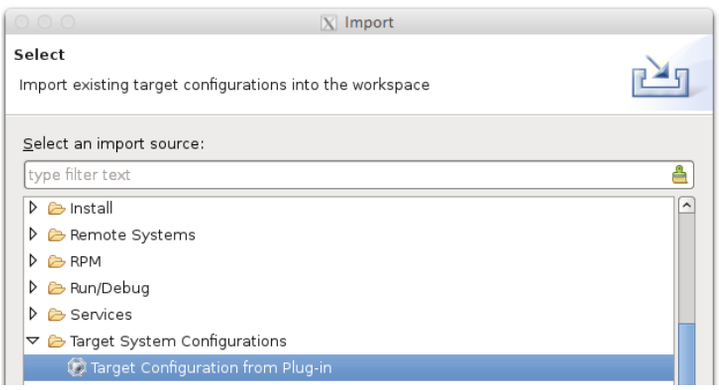
Target System Configuration (TSC) is necessary for Eclipse to work with Job Management System installed on a supercomputer. It is used for submitting jobs to a supercomputer, monitoring, etc.

To use PJM-* (and other custom) TSCs you need to create “targetConfigurations” project in Eclipse PTP in the following way:

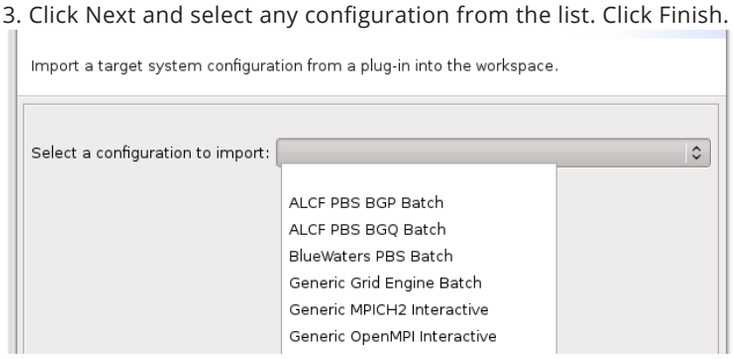
1. Right-click empty space of Project Explorer window and select Import, or select File / Import menu.



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.

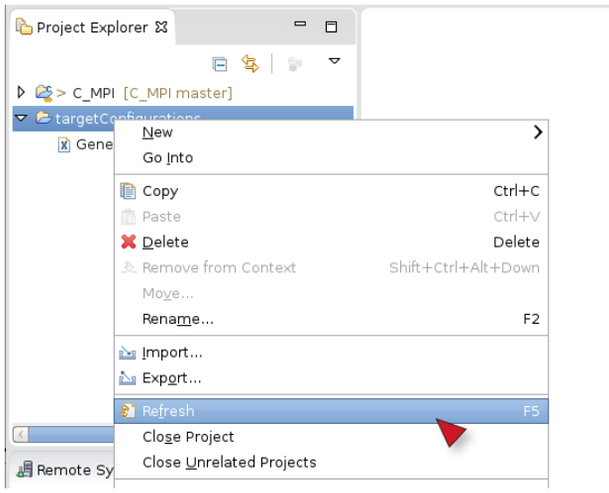
5. Find directory of targetConfigurations project in your Eclipse workspace directory on your disk. Removed imported TSC and download TSC for the K computer “PJM-Generic-MPI” into this directory with the following commands:

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

For FX10 use another TSC – “PJM-Generic-MPI-FX10”:

```
> rm *
> 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 target-Configurations project.

1.7 Install LML driver on remote computer

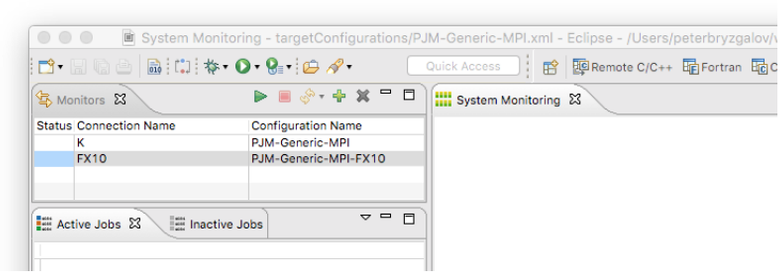
Monitoring in Eclipse PTP requires installation of LML DA driver on remote (super)computer.

Create a remote connection in Eclipse PTP Preferences / Remote Development / Remote Connections. See 1.4 for help on creating connections. It is "Built-in SSH" type connections that are used for monitoring.

In Eclipse PTP System Monitoring perspective create a new monitor in "Monitors" window with button.

To start a monitor use green triangle button above Monitors window.

Eclipse PTP System Monitoring pespective with two Monitor connections.



After you start a monitor for the first time `.eclipsesettings` directory is created under your home directory on the remote system and LML DA driver is installed into it. But because LML DA driver does not support "Parallelnavi" job management system installed on the "K" and FX10 computers, you will see errors. Stop the monitor with button .

Login to the remote computer. For the “K” computer execute the following in your home directory:

```
> cd .eclipsesettings
> git clone --branch K https://github.com/pyotr777/LML4PJM.git
> cd LML4PJM
> ./install.sh
```

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

This will install "Parallelnavi" support for LML DA driver.

Back in Eclipse PTP start again the monitor. After some time you should see a node map and active jobs list.

Note

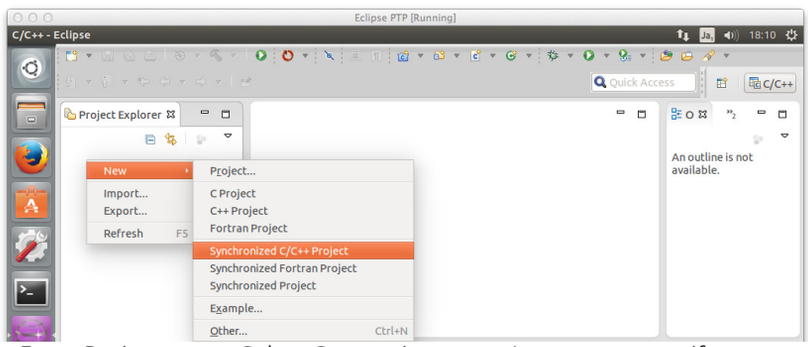
To test LML DA driver run `runlml` script in .eclipsesettings directory.

To debug monitoring with Eclipse PTP create file `.LML_da_options` file in .eclipsesettings directory and add `keeptmp=1` line into it. This will keep temporary directories "tmp_*" in .eclipsesettings directory. You can inspect files in these directories to see LML DA driver logs and to see what workflow and layout configurations have been used.

2. Create synchronized project

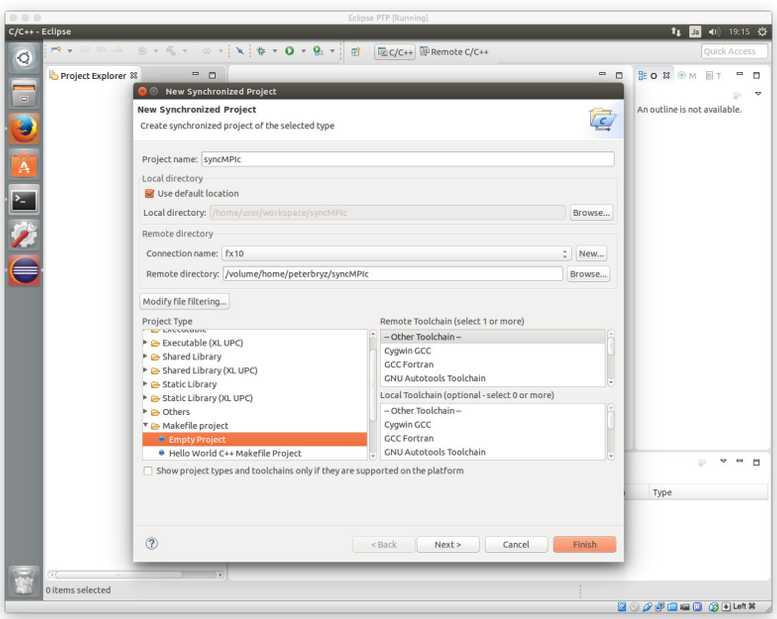
2.1 From code on remote location

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.

New Synchronized Project wizard window



Your new project will appear in Project Explorer.

Note your project icon. Synchronized project icon has a bidirectional arrow sign:

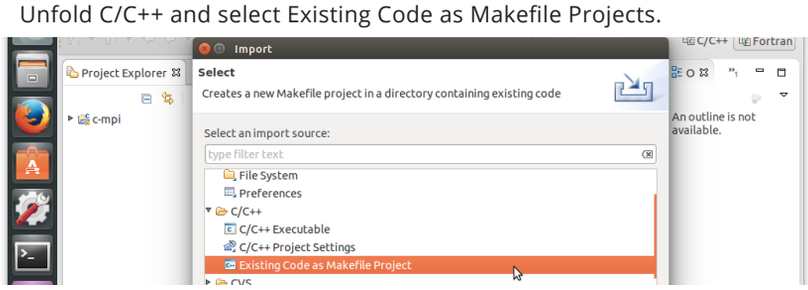
2.2 From local code

2.2.1. Import local code

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

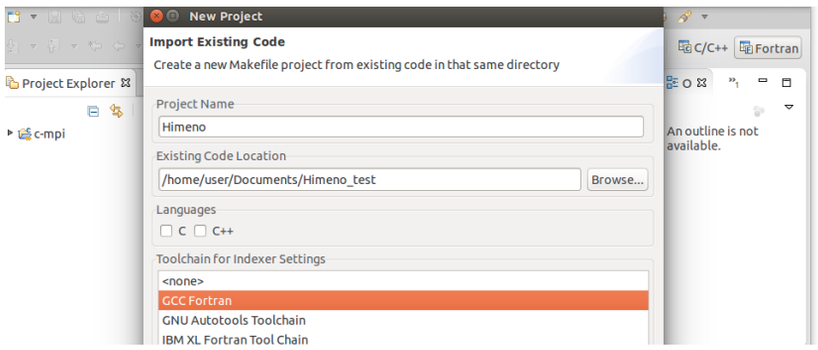
To convert local source code to Eclipse project right-click on empty space in Project Explorer and select Import...

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



Click Next.

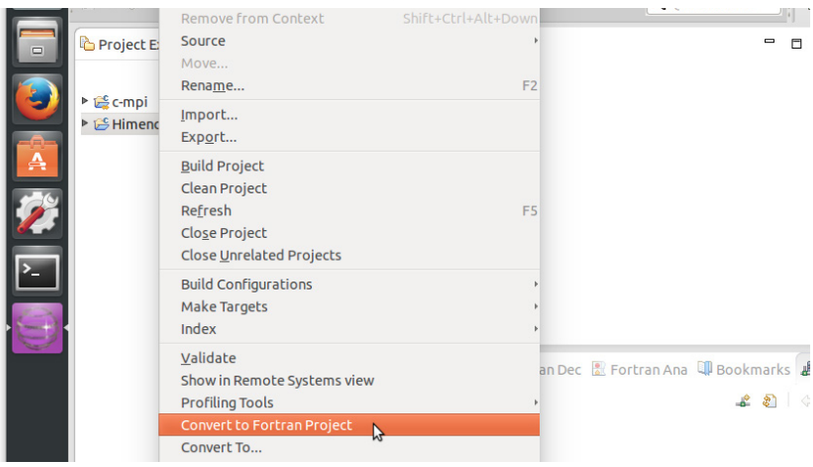
Give your project a name, select directory with your code, and select Toolchain.



Click Finish.

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

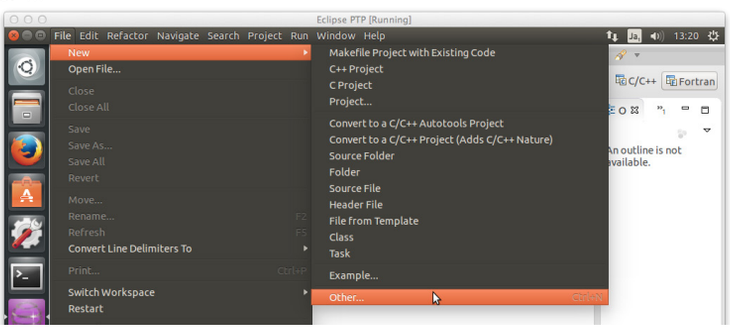
Convert it into a Fortran project. Right-click on the project in Project Explorer and select Convert to Fortran project.



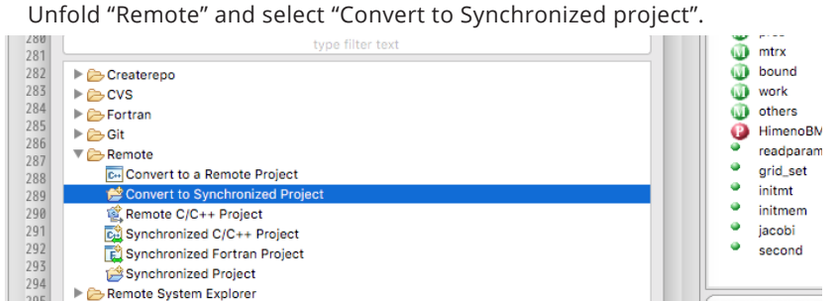
Make sure, the project folder icon changed to  . There is no arrow, because the project is not synchronized (local).

2.2.2. Convert local project to synchronized

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



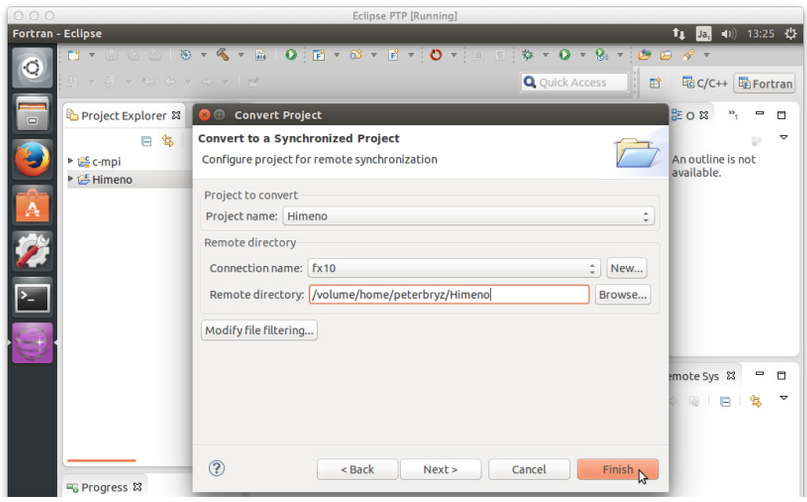
Unfold “Remote” and select “Convert to Synchronized project”.



Click “Next”.

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

Click Finish.



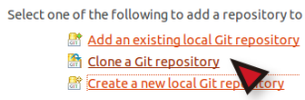
Note, that project icon changed to  .

2.3 From git repository

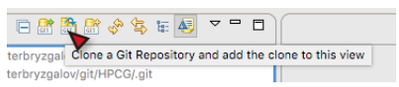
To create a new synchronized project from existing remote git repository:

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

In Git perspective select Clone a Git repository.

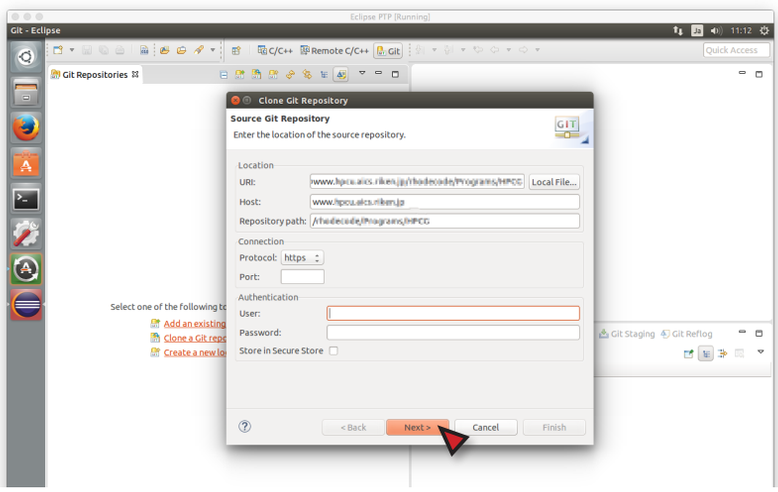


Or use toolbar:

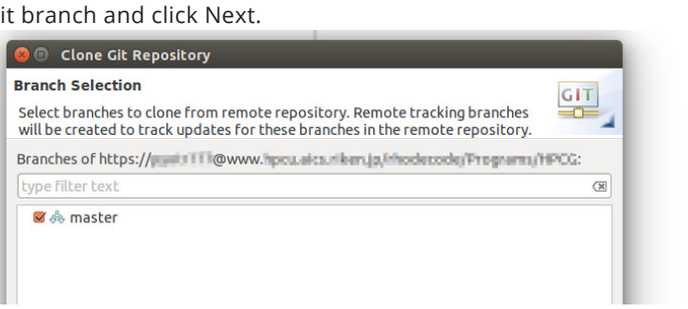


Fill in repository URI and other fields (some fields will be filled automatically after you fill URI). Click Next.

Cloning git repository



Select git branch and click Next.



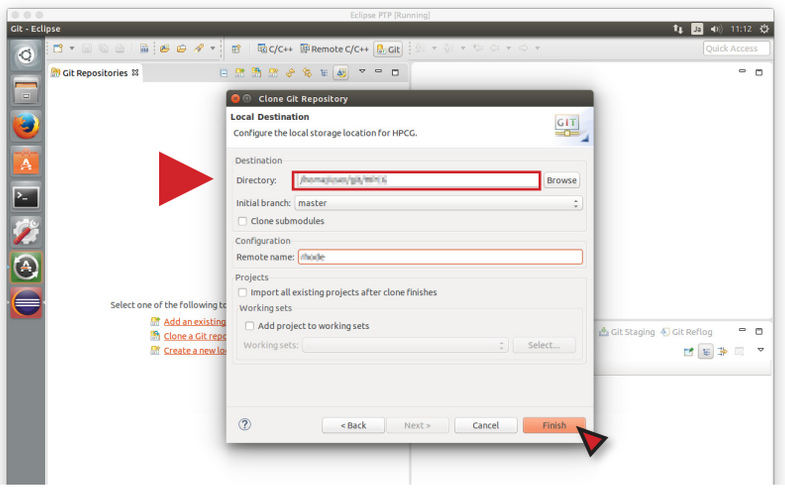
Note!

It is not recommended to make git repositories with Eclipse settings files included 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.

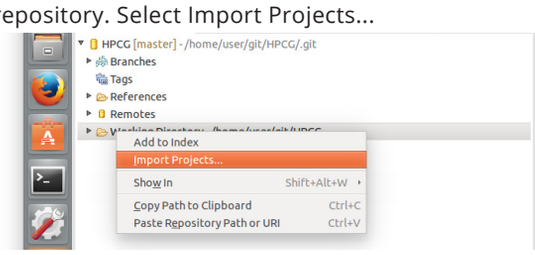
Remember this path, you will need it later



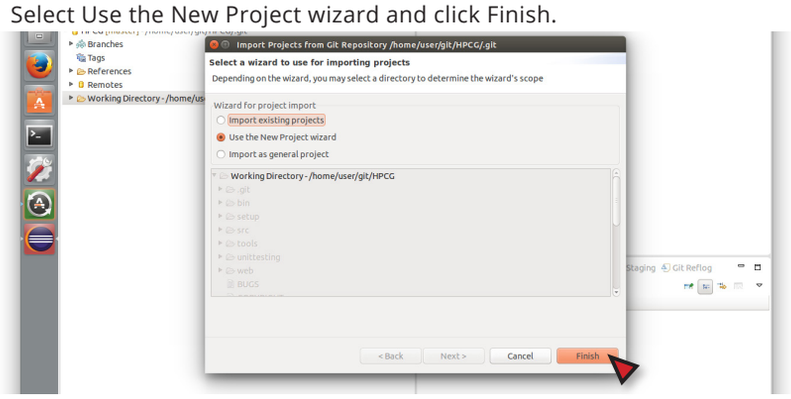
Repository is cloned.

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

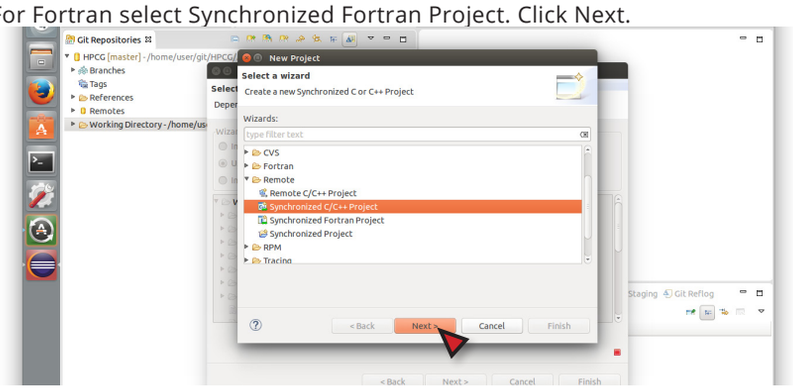
Importing new project from git repository



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.

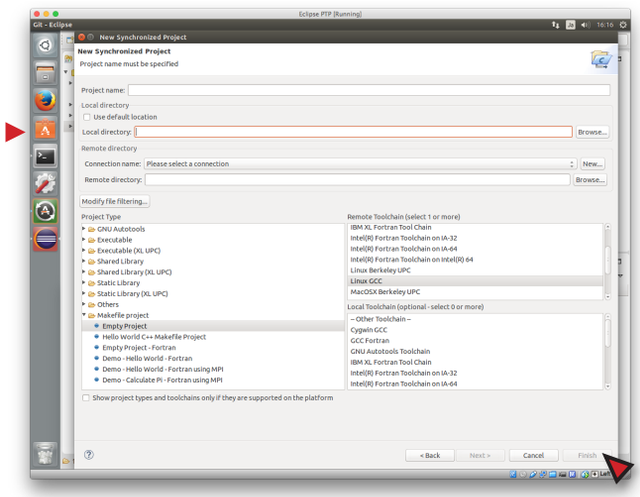


Set the “Project name” to the directory where you cloned the project. For example, if you cloned repository into /home/user/workspace/my_project directory, your project name will be “my_project”.

Select a connection. Set a folder to store the project on the remote machine in “Remote directory”. You can set it to whatever remote path you like.

Select Project Type. Makefile project / Empty Project will be good for most cases. Do not change toolchains. Click Finish.

New synchronized project wizard



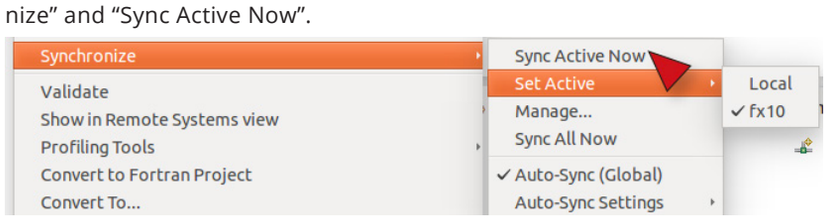
Your new project now should be visible in Project Explorer (C/C++ or Remote C/C++ perspective).

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

3. Synchronization

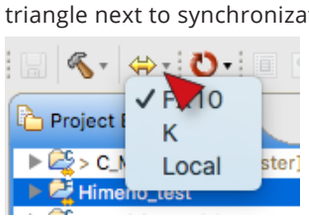
Files in local and remote locations of synchronized projects are automatically synchronized after project is built on remote location, but after project execution on the remote computer files are not synchronized automatically.

To manually 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”.



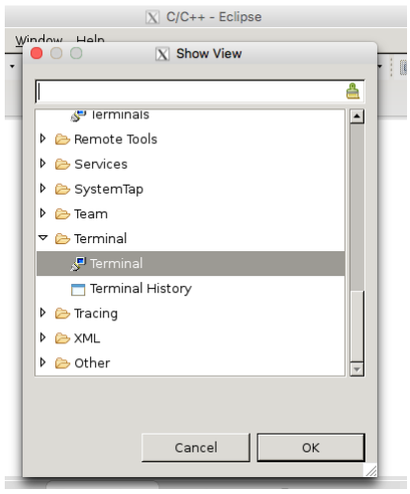
You can also select a project and press synchronization button ⇔ on the toolbar. Current active synchronization connection will be used.

To change connection used for synchronization use menu under “Set Active” item of the right-click menu as shown above or click on a little black triangle next to synchronization button on the toolbar.

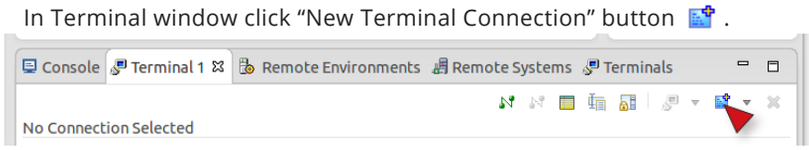


4. Remote terminal

With remote terminal you can login to remote system with SSH.
Open Terminal window from menu Window / Show view / Other ...
Terminal / Terminal.

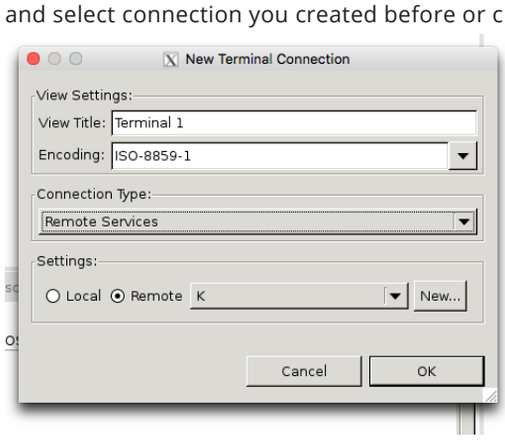


In Terminal window click “New Terminal Connection” button .

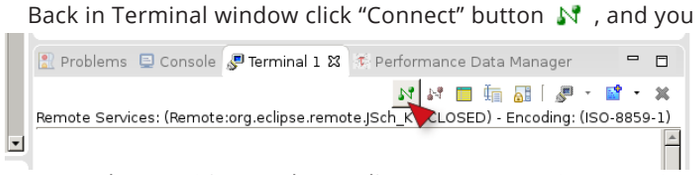


and select connection you created before or create a new connection.

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*.

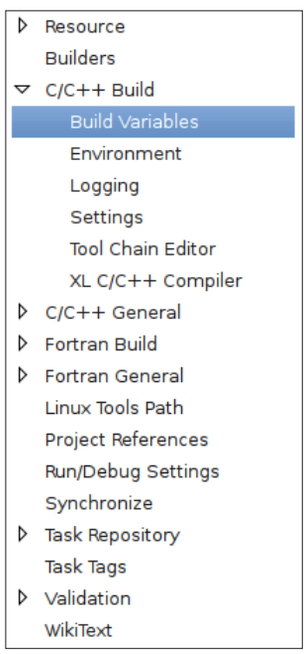
Makefile based project contains its own build command – a makefile or build script. For Managed projects Eclipse manages build process, no makefile required.

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 menu 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.

Project Properties menu



On the left you can see Project Properties menu.

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

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) .

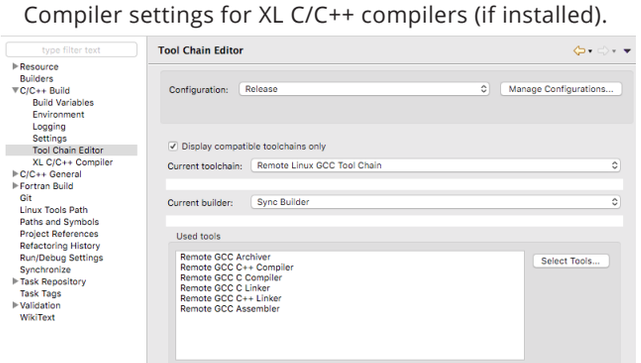
Tool Chain Editor

Compiler selection for local or remote build.

XL C/C++ Compiler

Compiler settings for XL C/C++ compilers (if installed).

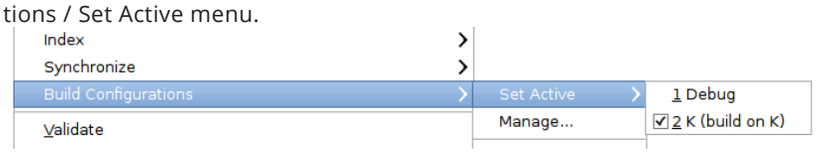
Sample tool chain settings for a synchronised C project.



C/C++ General/Preprocessor Include Paths...

Set include paths here.

Active build configuration will be used when the build button  is clicked. Active configuration can be changed with context menu: right click on project, then select the build configuration from the Build Configurations / Set Active menu.



6. Run a project

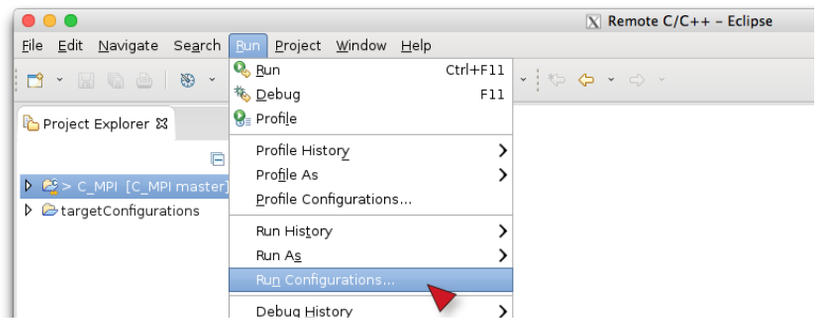
To run a project you need to set up at least one Run configuration. Run Configurations are used to define various parameters of running a Synchronized project as a parallel application on a remote system.

6.1 Run Configurations

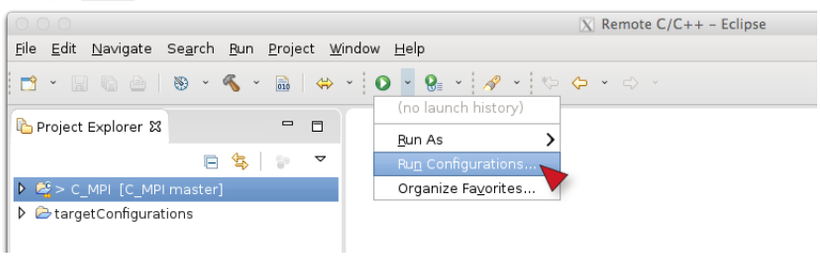
Run configurations uses Target System Configurations (TSC) for setting system-specific parameters. For K, FX10 and other systems with Fujitsu “Parallelnavi” job scheduler you can use PJM-* TSCs created by HPC Usability Research Team of AICS RIKEN. See below for instructions on how to install these TSCs.

6.1.1. Create Run Configuration

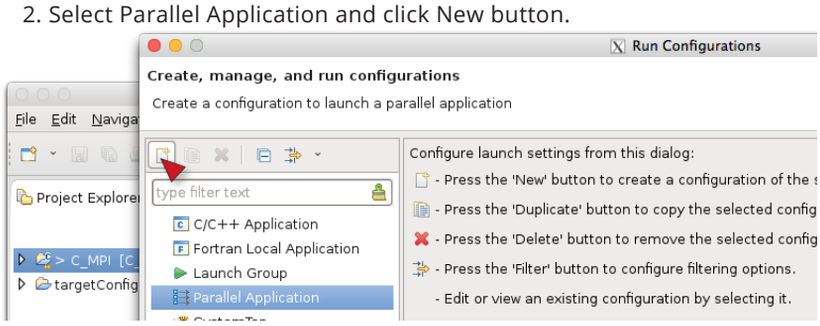
1. Select your project in Project Explorer and select Run / Run Configurations... menu



or use the menu under the downward black arrow next to the Run button on the toolbar.



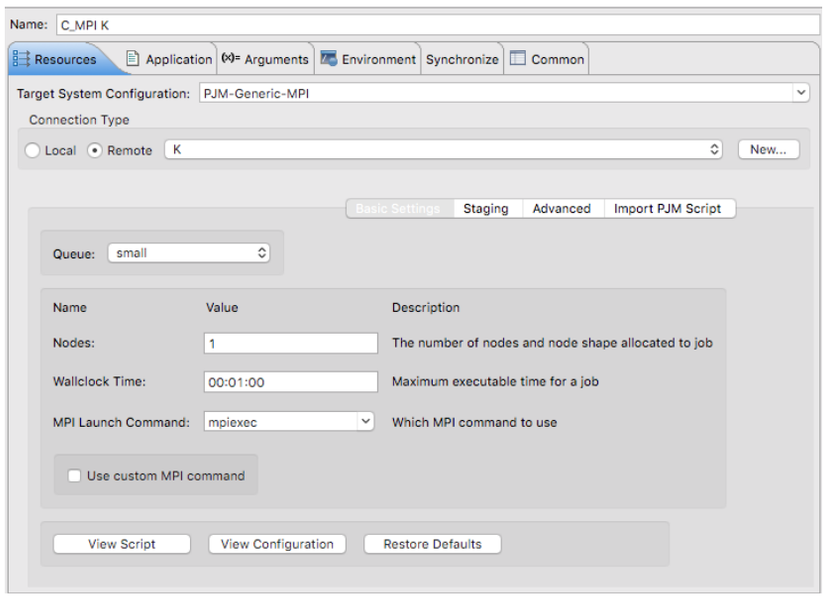
2. Select Parallel Application and click New button.



3. Give your new configuration a name, select Target System Configuration, connection to the remote system (one of synchronization connections you created at 1.5) or create a new connection.

On Basic Settings tab select Queue, number of nodes, estimated job run time (Wallclock time), and MPI command.

Run configuration window with PJM-Generic-MPI TSC.



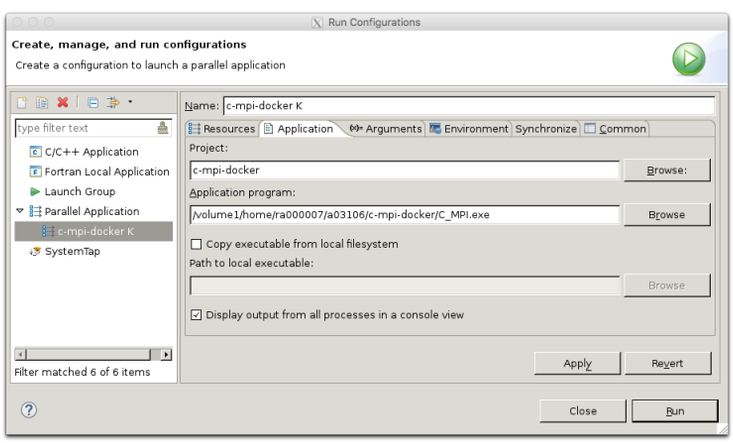
Note

TSC for FX10 computers "PJM-Generic-MPI-FX10" does not have "Staging" tab because staging is not used on FX10.

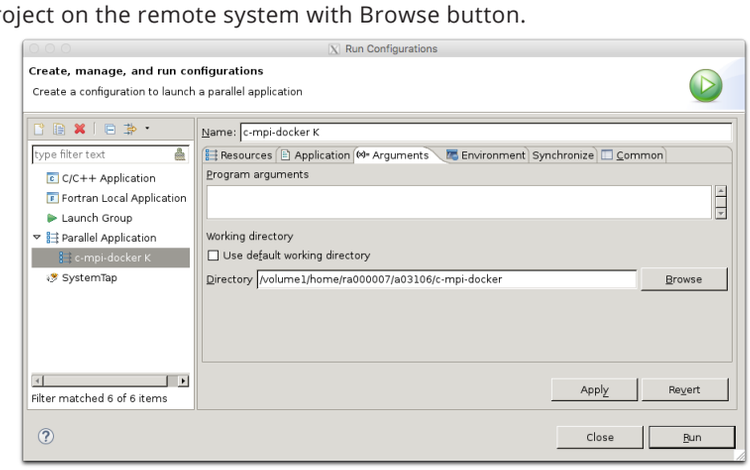
You can add staging commands to your job script using the Staging tab. Use Advanced tab for adding arbitrary commands to your job script. Text you enter in the text fields on this tab will be included in your job script without modifications.

If you already have a job script you want to use you can import it using Import PJM Script tab.

4. On Application tab select your execution file in Application program field.



5. On Arguments tab set Working directory to the absolute path of your project on the remote system with Browse button.



6. Define your program arguments on Arguments tab and environment variables on Environment tab, if necessary.
7. Click Apply button to save Run Configuration.

You can view your job script before submitting it by pressing “View Script” button on Resources tab.

Job scripts by default will not produce statistics reports on the “K” and FX10 computers. To create statistics reports add `#PJM -s` command to your job script using the advanced tab of Run Configurations.

To run program on remote computer press Run button of Run Configuration window or select previously saved Run Configuration from Run menu. Run menu can be accessed by pressing downward black arrow next to the Run button  on the toolbar.

After your job is submitted for execution on a remote system a dialog will appear asking if you want to switch to Monitoring view. Beware, that on the “K” computer rendering nodes map in Monitoring view takes some time (about 2 minutes or more).

6.2 Submitting a job manually in Terminal

Just in case you do not want to use Eclipse PTP Run configuration you can submit your job manually.

After your job script is ready, make sure the project is synchronized.

Login to the remote computer using Terminal window.

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

In Terminal window change directory on the remote machine 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 Eclipse PTP Project Explorer window.

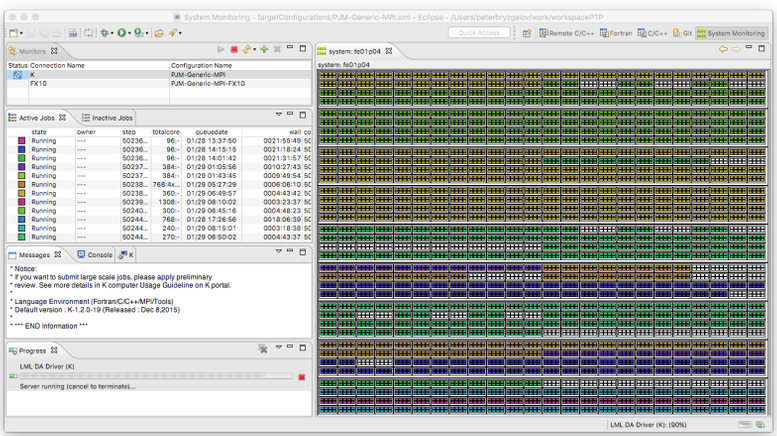
Note! Jobs submitted manually are not recognized by Eclipse. This means, in particular, that your project is not synchronized by Eclipse and you cannot control your job in Eclipse “System Monitoring” perspective (see 7.1).

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 **two minutes** before compute nodes chart will be rendered in Monitoring view.

The compute nodes chart and Active Jobs list will be updated about every 2 minutes for the "K" computer. Pressing "Refresh" button in Monitors window will not make updates faster.

Sample monitoring screen for the "K" computer



7.1 Monitoring

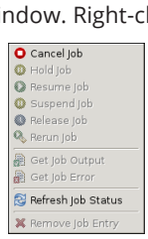
Monitoring scripts run at remote computer at predefined intervals (about 1-2 minutes) collecting information about compute nodes and active jobs. These scripts will continue running even if you switch to another perspective in Eclipse PTP and Monitoring view will be hidden.

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.

Clicking on a running job in Active Jobs window will display nodes used by that job on the compute nodes chart.

7.2 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.

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

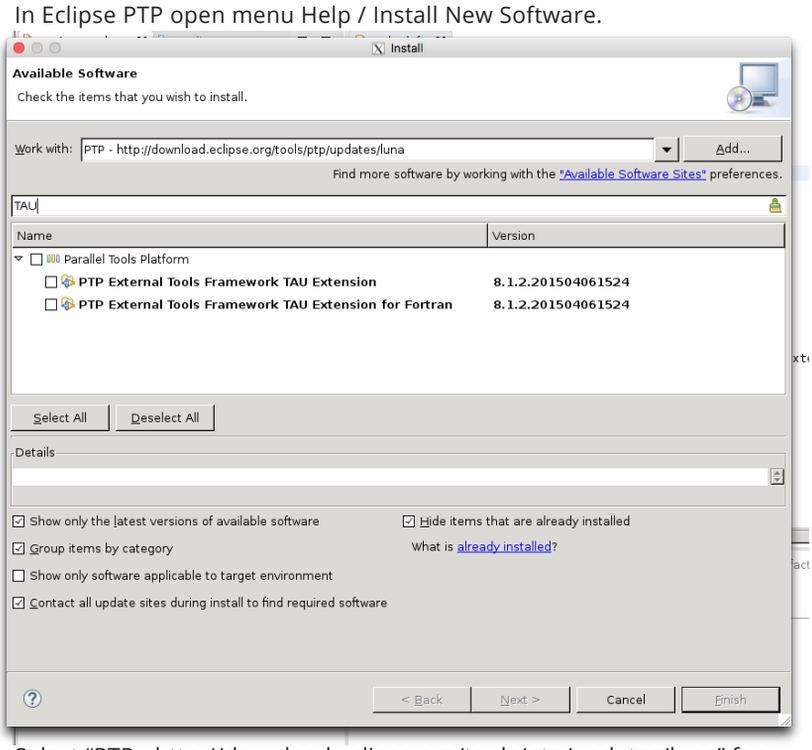
```
mkdir -p $HOME/TAU
cd $HOME/TAU
wget http://tau.uoregon.edu/tau.tgz
tar -xzvf tau.tgz
cd tau-2.25
./configure
make
make install
```

Latest TAU distributions can be downloaded from
https://www.cs.uoregon.edu/research/tau/downloads.php
For OS X install TAU from the dmg package. For Windows install TAU with exe file.

Add the following directory to $PATH variable on Linux and OS X:
$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 “PTP - http://download.eclipse.org/tools/ptp/updates/luna” for Luna and “PTP - http://download.eclipse.org/tools/ptp/updates/mars” for Mars.

Select TAU Extensions and click Finish button. Restart Eclipse.

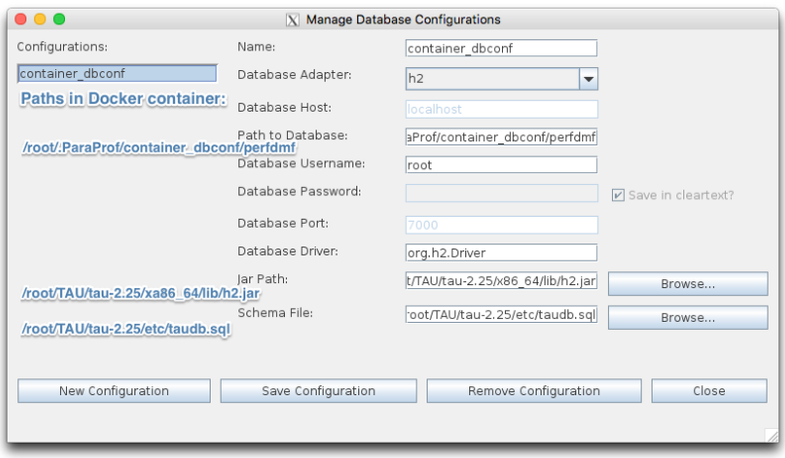
8.1.2. Setup local Database for profiles

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:

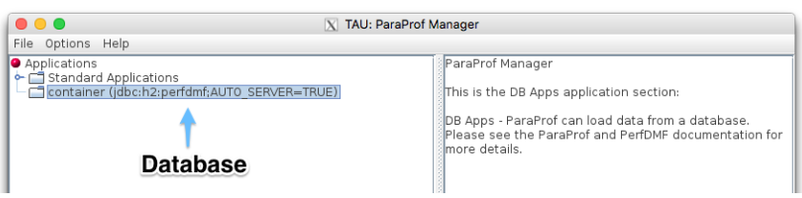
Profiles Database configuration in Paraprof



For the database port any number of unused port greater than 1024 will do.
Save configuration.
Instead of Paraprof you can use DB configuration script:

```
/applications/tau/tau/apple/bin > ./perfdmf_configure
```

If you successfully created DB, you will see it in Paraprof:

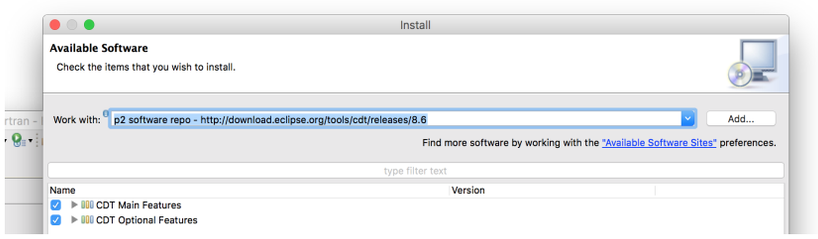


Please note that traces cannot be stored in a database.

Now install plugins into Eclipse PTP as instructed here: https://www.cs.uoregon.edu/research/tau/docs/newguide/ch06.html

```
/applications/tau/tau > /Applications/TAU/tau/tools/src/eclipse/install_plugins.sh "/Applications/eclipse PTP/"
Installing to /Applications/eclipse PTP//dropins
...
Eclipse plugins installed!
```

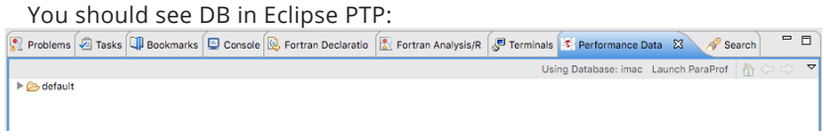
Install CDT plugins if they are not yet installed:



Restart Eclipse with flag -clean:

```
/Applications/eclipse PTP > ./eclipse -clean
```

You should see DB in Eclipse PTP:



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 occurred 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

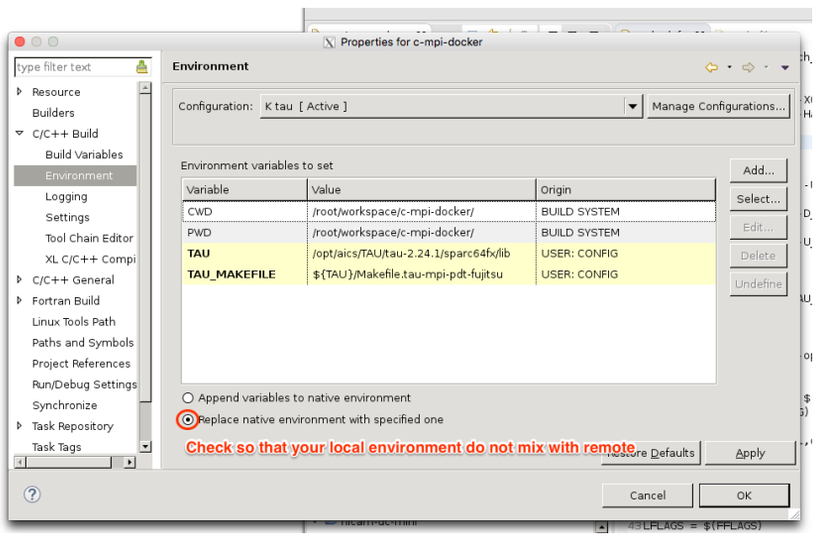
Source instrumentation requires PDT and produces larger profiles and traces, so it is recommanded to start with Compiler instrumentation and use Source instrumentation if more features are needed.

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 instrumented binary can be produced in exactly the same manner as compiler based instrumentation. The only difference is that you do not need to add -optCompInst option (or environment 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

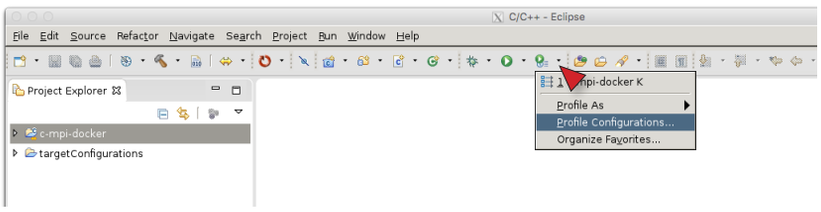
Environment variable	Effect
TAU_PROFILE	Set to 1 to enable profiling, to 0 to disable.
TAU_TRACE	1 to enable tracing, 0 to disable
TAU_PROFILE_FORMAT	When set to snapshot TAU will generate condensed snapshot profiles (they merge together different metrics so there is only one file per node) instead of 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 directory 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/newguide/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.

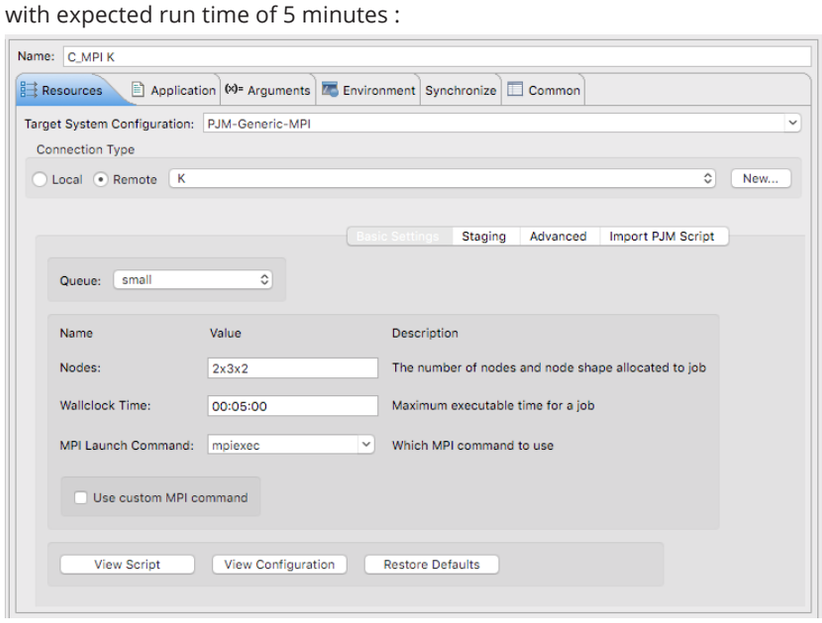
For a Target System Configuration select "PJM-Generic-MPI" for K comuter and "PJM-Generic-MPI-FX10" for FX10.

Select a desired Connection or create a new one.

Fill in basic job parameters in Basic Settings tab.

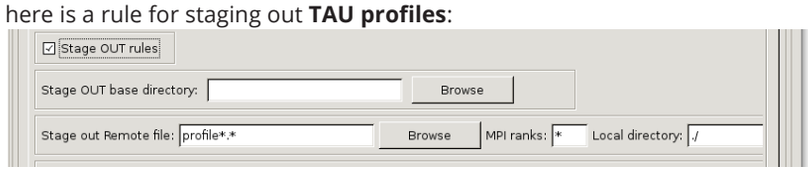
Here is a Profile Configuration for running a job on 12 nodes of K computer with expected run time of 5 minutes :

Sample Profile Configuration window



You have to take care about staging out profiles and traces from compute nodes to a 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

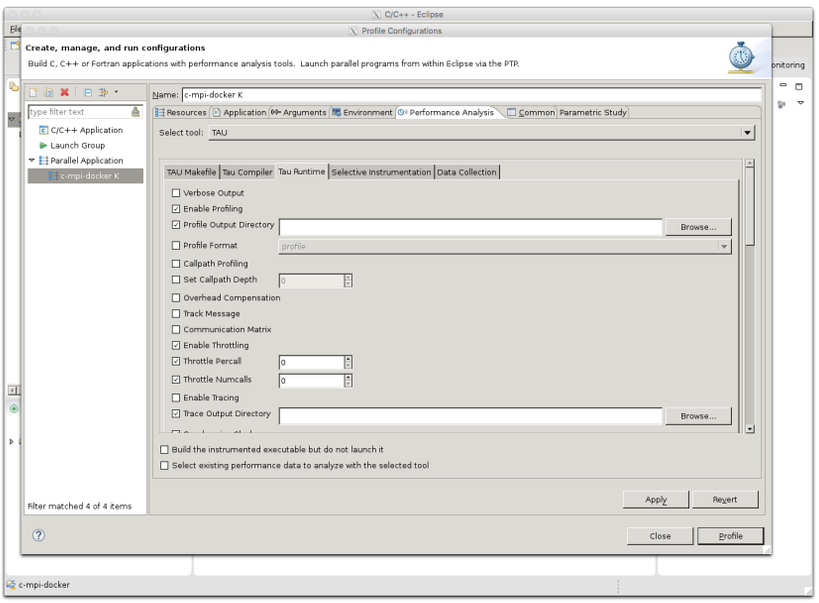


TAU traces consist of two sets: evens.N.edf files and tautrace.N.0.0.trc files. N is a number from 0 to the number of MPI ranks -1. You need to stage out all of these files.

On Application tab select a binary instrumented with TAU.

Configure TAU behaviour on compute nodes

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

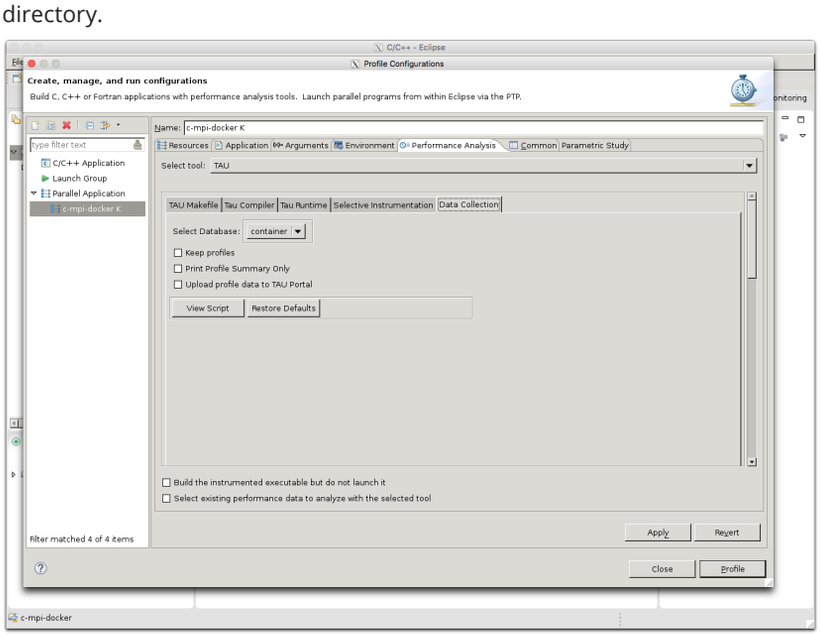


On Runtime tab of Performance Analysis you have to take care of Porfile and Trace Output Directory.

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.

Configure profiles collection



On Data Collection tab select Database to store profiles. See 8.1.2 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.

8.3.3. 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.

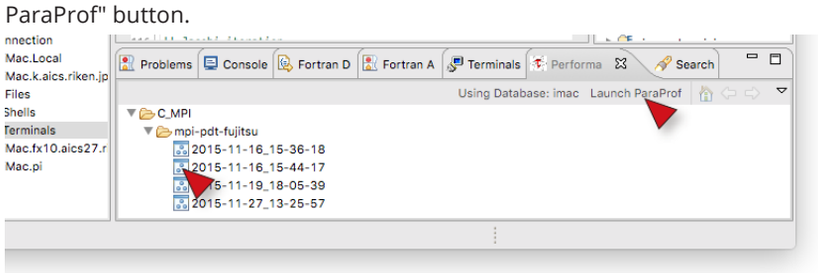
Do not forget to set staging out rules for profiles or traces in your job script.

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

8.4 View TAU profiles

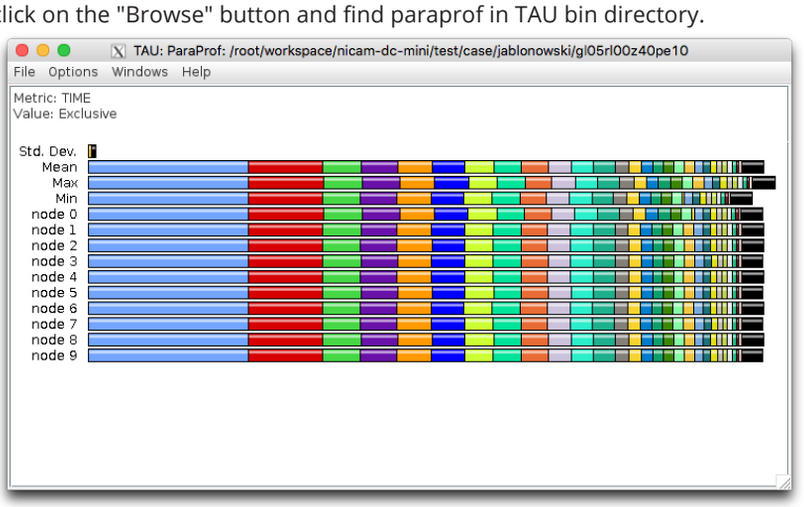
TAU profiles can be viewed in **Paraprof**.

If you have set up a profile DB, you can launch Paraprof from Eclipse PTP "Performance Data Manager" view by clicking on a profile or on the "Launch ParaProf" button.



Otherwise you can unfold Profiles directory of your project in Project Explorer, right-click on any profile and select Open With / Other... menu, then click on the "Browse" button and find paraprof in TAU bin directory.

Paraprof window with exclusive execution times of routines in an application



For Paraprof usage details see TAU documentation web site: https://www.cs.uoregon.edu/research/tau/docs.php

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.

To merge trace files trautrace.*.trc and events.*.edf run

```
> tau_treemerge.pl
```

in the directory with trace files. This command will produce two files: tau.edf and tau.trc.

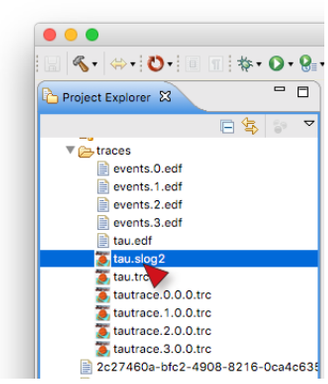
To convert these files to SLOG2 format run in the same directory

```
> traceconv.sh tau
```

traceconv.sh is a custom script installed on the K computer. If it cannot be found use the following command instead:

```
> tau2slog2 tau.trc tau.edf -o tau.slog2
```

To open trace file in Jumpshot right-click on the SLOG2 file in Project Explorer, select Open With / Other... menu, then click on the "Browse" button and find jumpshot in TAU bin directory.



Jumpshot users guide can be found here:

http://www.mcs.anl.gov/research/projects/perfvis/software/viewers/jumpshot-4/usersguide.html

Jumpshot window with timline.
Yellow lines show goups of MPI communication events between ranks.